# Subanta Generator Paper

[How to use the code:](#h.guezwoklub8s)

[For using the code online:](#h.jv8m65g62icm)

[For using the code offline on your computer:](#h.1q58xjojosa0)

[For Sandhi generation:](#h.5unkfjmdzb81)

[For offline usage:](#h.zcv5jkcdls3o)

[For Subanta generation:](#h.kb2vecn235zy)

[For offline usage:](#h.pgh63gdqo8ez)

[Code location:](#h.xtl6jxfnr2ri)

[Actual codes used:](#h.gx95qrezi5rm)

[1. ajax requirements](#h.eoo8j4fmokpq)

[1. पुल्लिङ्ग](#h.ao114nybfuh5)

[2. स्त्रीलिङ्ग](#h.mcx25e25izl0)

[3. नपुंसकलिङ्ग](#h.y3sxhybznm18)

[2. ajax.php](#h.hxvadg3vbx25)

[3. function.php](#h.hhxjxyy56cyl)

[/\* Defining grammatical arrays \*/](#h.gkxwjqfo5yom)

[/\* Function to find pratyAhAra from given two letters \*/](#h.nt4oc4aid3r7)

[/\* function pratyAhAra check is for checking in regular expressions.](#h.saifhlom2z48)

[/\* Function one is for replacing one letter in the whole array of $text with another letter \*/](#h.h7lj1uks6df3)

[/\* Function two is for replacing one letter in the whole array of $text with another letter \*/](#h.3pfvxfe3cs6c)

[/\* Function three is for replacing one letter in the whole array of $text with another letter \*/](#h.s2fbtptays86)

[/\* function flatten is a function to flatten a multidimentional array \*/](#h.b2vrednxbieg)

[/\* function savarna is to find out savarNa of a character from the given pratyAhAra \*/](#h.wpw3ee8ehpqv)

[/\* function display will show the text on screen. \*/](#h.jla0ljdupil4)

[/\* function dvitva will be used to duplicate a letter \*/](#h.avcbjpdrhm6s)

[/\* function flat is used to make an array useful for regular expressions \*/](#h.hxlo3lposlyv)

[/\* function blank will return an array having n blank members. blank(2) === array("",""); \*/](#h.jscjitht811j)

[/\* function checkarray is used to see whether the sequence of $a,$b,$c,$d is found in any of the members of $text \*/](#h.6o0x1kkroo0l)

[/\* function sub is a modified version of checkarray. It searches for the consecutive occurences of $a,$b,$c in the members of array $text.](#h.rtvswvd9r3i5)

[/\* function arr to search the occurence of a pattern in any of the member of an array \*/](#h.d2bn8ijiqf1f)

[/\* function to remove the last n letters from a word in the $text array and replace them with another word \*/](#h.4fbqk1fy6ki)

[/\* function to remove the first n letters from a word in the $text array and replace them with another word \*/](#h.rpyjlh7t6emf)

[/\* vriddhi function \*/](#h.ujfv58r1yw4z)

[/\* guna function \*/](#h.2qxp9kj2mxzh)

[/\* dirgha function \*/](#h.soa6mf2gca94)

[/\* anekAca function \*/](#h.jd7c2ndmpyis)

[/\* ends function \*/](#h.lcmnqgkt8gwq)

[/\* function Ti to remove the Ti \*/](#h.gfk3wv4ilh30)

[/\* function mit \*/](#h.6ggq539cf8xl)

[/\* function samprasarana \*/](#h.yf65grcrhk16)

[/\* function itcheck \*/](#h.al1h7kecdmt3)

[/\* function it (to find an it marker). The function adds the marker to its arrays. \*/](#h.qgba4zewxdf)

[/\* function to find out it markers when they can occur only in the pratayayas. \*/](#h.ere0v0dp56ug)

[/\* Functions which are not used in the code \*/](#h.bvwstq83fkwz)

[4. mystyle.css](#h.raremwyt1gik)

[5. script.js](#h.ewoufwxbbpim)

[6. subanta.html](#h.ta2lteco1pne)

[7. subanta.php](#h.hxvadg3vbx25)

[/\* Including arrays and functions \*/](#h.pjjmq1yf1vay)

[/\* hides error reports. \*/](#h.c4m6uj6pemgf)

[/\* set execution time to an hour \*/](#h.fgbfliayk0fq)

[/\* Reading from the HTML input. \*/](#h.po7atgdheh8d)

[/\* Defining the variables used in the code and their default values .](#h.kwel3j77c7si)

[/\* a for loop for entering all sup pratyayas one by one. Sambuddhi is at the last after sup. \*/](#h.2awy3l889kx7)

[/\* Code for converting from IAST to SLP1 \*/](#h.og3ry61cln5m)

[/\* Code for converting from devanagari - SLP1 \*/](#h.fkd2mqrsvk6w)

[/\* displaying the data back to the user \*/](#h.qtj5l8wxiclr)

[/\* for sambodhana vibhaktis display \*/](#h.ni5r2n3ex30m)

[/\* preprocessing for the sup pratyayas. \*/](#h.kejxswlmrj7)

[/\* main coding part starts from here. Based on Siddhantakaumudi text. \*/](#h.y2c4iiyi5b9y)

[/\* Defining an array $text. \*/](#h.bb9s46r13fc3)

[/\* displaying general information about the sup vibhaktis \*/](#h.irukgh60awb8)

[/\* arthavadadhAturapratyayaH prAtipadikam (1.2.45), kRttaddhitasamAsAzca (1.2.46), pratyayaH (3.1.1), parazca (3.1.2), GyAppradipadikAt (4.1.1), svaujasamauTCaSTAbhyAmbhisGebhyAmbhyasGasibhyAmbhyasGasosAmGyossup (4.1.2), vibhaktizca (1.4.104), supaH (1.4.103) \*/](#h.3mee8k37e003)

[/\* dvyekayordvivacanaikavacane (1.4.22) \*/](#h.nctbjachlfz8)

[/\* bahuSu bahuvacanam (1.4.21) \*/](#h.g38937oq2oiq)

[/\* sarUpANAmekazeSa ekavibhaktau (1.2.64) \*/](#h.j161r74h7d4z)

[/\* ekavacanaM sambuddhiH (2.3.49) \*/](#h.ddotmsuakdve)

[/\* nityabahuvacanAnta special messages \*/](#h.u2ccvh9imgl6)

[/\* nityadvivacanAnta special messages \*/](#h.whj5uz7ppfzv)

[/\* tyadAdi sambodhana special messages \*/](#h.jho6bid91hdo)

[/\* defining sarvanama status \*/](#h.5id15oz6x862)

[/\* vibhASAprakaraNe tIyasya GitsUpasaMkhyAnam (vA 242) \*/](#h.5e852kl74gr)

[/\* sarvAdIni sarvanAmAni (1.1.27) \*/](#h.hmj7dtrrxg8x)

[/\* defininig eranekAca \*/](#h.qffmlqg6hzuf)

[/\* defining oH supi variable \*/](#h.qlr2wn7mjx5j)

[/\* GI definition \*/](#h.puy6iorxy1e)

[/\* makArAnta pulliGga dhAtu definition \*/](#h.s6veymc883o3)

[/\* idam, idakam anvAdeza definition \*/](#h.vzil0jeaslbg)

[/\* nakArAnta pulliGga ShanvAat definition \*/](#h.j8zx5ou6ut6q)

[/\* jakArAnta pulliGga kvin pratyaya definition \*/](#h.7qqsalh8upn3)

[/\* tyadAdi gauNatva definition \*/](#h.vws455rvswnr)

[/\* asmad / yuSmad -> nau / naH etc definition \*/](#h.348uai8bt3no)

[/\* ancu definitions \*/](#h.9dic9ieyyztl)

[/\* defining goanc \*/](#h.c25d99gyy7qg)

[/\* defining Nyanta \*/](#h.f908w32ndjl3)

[/\* defining dhatus \*/](#h.xxmybea2je8v)

[/\* defining kvip \*/](#h.k7n9huf11rrv)

[/\* strIliGga definitions \*/](#h.5i89mcu6ot1h)

[/\* defining Abantatva \*/](#h.usuwdjk25v7p)

[/\* nityastrIliGga definition \*/](#h.cq8zv4wvbw37)

[/\* praDI definition \*/](#h.d5kwje2t2usc)

[/\* sulU definition \*/](#h.5ozmo8q8dldp)

[/\* suDI definition \*/](#h.3lai7rjz6uty)

[/\* UkArAnta strIliGga - defining \*/](#h.pngwxgm49u7h)

[/\* punarBU definition \*/](#h.sjwuoi6gwjaj)

[/\* bhASitapuMska definition \*/](#h.u97wtt4hldy3)

[/\* tRtIyAdiSu bhASitapuMskaM puMvadgAlavasya (7.1.74) \*/](#h.4miiqusu6tcg)

[/\* defining kRt pratyayas \*/](#h.9dnk7e19u74b)

[/\* kRdatiG (3.1.93) \*/](#h.ex404l4me10x)

[/\* Defining pada and bham \*/](#h.yz1qd7pygedy)

[/\* suDanapuMsakasya (1.1.43) \*/](#h.2l5n84175n6a)

[/\* bahuguNavatuDati saGkhyA (1.1.28) \*/](#h.rcpj4ngy4ltg)

[/\* Dati ca (1.1.25) \*/](#h.q53q0gal8upy)

[/\* SNAntA Sat (1.1.24) \*/](#h.881v075xo4df)

[/\* defining "SaT" \*/](#h.6pn36botxi4p)

[/\* same in all lingas - special messages \*/](#h.2rvd5rd5euel)

[/\* checking for presence of aJcu verb. \*/](#h.u4sq0ubz6exr)

[/\* zatRvat finder \*/](#h.ijy3zetbs1nh)

[/\* atvanta finder \*/](#h.a120o3tpljtn)

[/\* na SaTsvasrAdibhyaH (4.1.10) \*/](#h.f683la6lr3sc)

[/\* striyAM ca (7.1.96) \*/](#h.yullcuz900ib)

[/\* Rnnebhyo Gip (4.1.5) \*/](#h.7nrutfd3bm1r)

[/\* dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118) \*/](#h.w8daolpkoxi5)

[/\* yU stryAkhyau nadI (1.4.3) and prathamaliGgagrahaNaJca (vA 1036) \*/](#h.wisw01dsqfng)

[/\* abhyasta definition \*/](#h.qhwwgbdr1rq)

[/\* jakSityAdayaH SaT (6.1.6) \*/](#h.y7vunperw1ym)

[/\* ubhe abhyastam (6.1.5) \*/](#h.21k53c8mqj4f)

[/\* tyadAdiSu dRSo'nAlocane kaJca (3.2.60) \*/](#h.jn117dl7obfb)

[/\* kvin pratyaya from asRj \*/](#h.bziouwccwvin)

[/\* no kvin pratyaya from viSvasRj \*/](#h.dokpbxcr9acb)

[/\* spRSo'nudake kvin (3.2.58) \*/](#h.9640brtkrf63)

[/\* Defining $vasu \*/](#h.mwrq6g2n65rg)

[/\* Defining ugit \*/](#h.fvmxctiy7fe8)

[/\* dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118) \*/](#h.wehy5zwvcd6t)

[/\* adDDatarAdibhyaH paJcabhyaH (7.1.25) and ekatarAtpratiSedho vaktavyaH (vA 4287) \*/](#h.8p5ttgq5lz2w)

[/\* maghavA bahulam (6.4.128) \*/](#h.htnymsq5e5h8)

[/\* arvaNastrasAvanaJa (6.4.127) \*/](#h.azbdytljh991)

[/\* patch for dRz \*/](#h.etiwhwvuehsh)

[/\* RtvigdadhRksragdiguSNigaJcuyujikruJcAM ca (3.2.59) \*/](#h.mzt96iyjtm17)

[/\* rAyo hali (7.2.85) \*/](#h.p06kv37sav6g)

[/\* hrasvo napuMsake prAtipadikasya (1.2.47) \*/](#h.b92dlyl3eozi)

[/\* ato'm (7.1.24) \*/](#h.hf9db9mh1ab1)

[/\* defining whether the first word is asmad / yuzmad \*/](#h.cqa6kdumxj6n)

[/\* Whole replacements for asmad / yuSmad \*/](#h.euwq6otfffv8)

[/\* yuSmadasmadoH SaSThIcaturthIdvitIyAsthayorvAMnAvau (8.1.20) \*/](#h.9sw2r4nokgl8)

[/\* bahuvacanasya vasnasau (8.1.21) \*/](#h.h09g98qukpnr)

[/\* temayAvekavacanasya (8.1.22) \*/](#h.i6gt0qu4i2ep)

[/\* tvAmau dvitIyAyAH (8.1.23) \*/](#h.hq4mzgak7tq5)

[/\* pratyaya Adeza for yuSmad / asmad \*/](#h.wwyuqx2dpyft)

[/\* yuSmadasmadbhyAM Gaso'z (7.1.27) \*/](#h.mz879i803xgo)

[/\* Geprathamayoram (7.1.28) \*/](#h.8rqozv3qsygn)

[/\* zaso na (7.1.29) \*/](#h.cy10heoth0pf)

[/\* bhyaso bhyam (7.1.30) \*/](#h.smxqmsrdclvd)

[/\* paJcamyA at (7.1.32) \*/](#h.2rwuhaks4yhv)

[/\* ekavacanasya ca (7.1.32) \*/](#h.o9b5ewjkieqb)

[/\* prakRti Adezas for asmad / yuSmad \*/](#h.vguynwz7e67b)

[/\* yo'ci (7.2.89) \*/](#h.hxeprt5k7vhs)

[/\* yuSmadasmadoranAdeze (7.2.86) \*/](#h.673yug45o5md)

[/\* dvitIyAyAJca (7.2.87) \*/](#h.rrnaq5u5l9dk)

[/\* zeSe lopaH (7.2.90) \*/](#h.ro4r61hku2lp)

[/\* maparyantasya Adezas for asmad / yuSmad \*/](#h.lugu8bbmlnmw)

[/\* yUyavayau jasi (7.2.93) \*/](#h.uzaf8or6mbpm)

[/\* tvAhau sau (7.2.94) \*/](#h.si590y59823)

[/\* tubhyamahyau Gayi (7.2.95) \*/](#h.89zdmxagtb0t)

[/\* tavamamau Gasi (7.2.96) \*/](#h.wx9qsobpi20g)

[/\* tvamAvekavacane (7.2.97) \*/](#h.x6hefey39jc7)

[/\* yuvAvau dvivacane (7.2.92) \*/](#h.29ra6xuj3mgj)

[/\* ato guNe patch \*/](#h.rbn7ppbqnyr7)

[/\* prathamAyAzca dvivacane bhASAyAm (7.2.88) \*/](#h.no0vsag1xamp)

[/\* aGgakArye kRte punarnAGgakAryam (pa 93) \*/](#h.x4fp6dr2v5v0)

[/\* idamo maH (7.2.108) \*/](#h.tapxlhy3w3rn)

[/\* ido'y puMsi (7.2.111) \*/](#h.k33eajqzadpv)

[/\* yaH sau (7.2.110) \*/](#h.thtwdxcp0x24)

[/\* anvAdeze napuMsake enadvaktavyaH (vA 1569) \*/](#h.rxr42tg2bsqf)

[/\* svamornapuMsakAt (7.1.23) \*/](#h.n2xi0pc6t1ta)

[/\* kimaH kaH (7.2.103) \*/](#h.q2ix7tqjqb6h)

[/\* Rnnebhyo GIp (4.1.5) \*/](#h.woi9tp5vpkf7)

[/\* aSTana A vibhaktau (7.2.84) \*/](#h.ybw9xqy5gzii)

[/\* aSTAbhya auz (7.1.21) \*/](#h.hjjgdywt1bk1)

[/\* pratyayasya lukzlulupaH (1.1.61) \*/](#h.aeshhwbwfrpl)

[/\* SaDbhyo luk (7.1.22) \*/](#h.m05vycaurflv)

[/\* pratyayalope pratyayalakSaNam (1.1.62) and na lumatAGgasya (1.1.63) \*/](#h.pwlwcrbmptra)

[/\* jazzasoH ziH (7.1.20) \*/](#h.ald3l3mn8f3)

[/\* zi sarvanAmasthAnam (1.1.42) \*/](#h.t3i87x1s2hqc)

[/\* declaring sarvanamasthana1 variable \*/](#h.7ygq1i8zqne7)

[/\* am sambuddhau (7.1.99) \*/](#h.zerpt4yv1u5u)

[/\* puMso'suG (7.1.89) \*/](#h.yx7g051pvel8)

[/\* yujerasamAse (7.1.71) \*/](#h.nsg93bol8mu)

[/\* A sarvanAmnaH (6.3.91) \*/](#h.ut316kwi71if)

[/\* defining tri variable for tisRcatasR \*/](#h.qdry6d4bslnu)

[/\* tricaturoH striyAM tisRcatasR (7.2.99) \*/](#h.q3tqk1acevhc)

[/\* caturanaDuhorAmudAttaH (7.1.98) \*/](#h.p2emdhip9hi9)

[/\* sAvanaDuhaH (7.1.82) \*/](#h.u5oaoptd6xs1)

[/\* jarAyA jarasanyatarasyAm (7.2.101) \*/](#h.zg1rmgh6cxjx)

[/\* paddannomAshRnnizasanyUSandoSanyakaJChakannudannAsaJChasprabhRtiSu (6.1.63) \*/](#h.i4ywyoxsqnhy)

[/\* mAMsapRtanAsAnUnAM mAMspRtsnavo vAcyAH SasAdau vA (vA 3416) \*/](#h.wzrc3kh8z9gc)

[/\* asthidadhisakthyakSNAmanaGudAttaH (7.1.75) \*/](#h.lq8mkvfep3ry)

[/\* saGkhyAvisAyapUrvasyAhnasyAhannanyatarasyAM Gau (6.3.110) \*/](#h.tw9gbrnr5eq8)

[/\* sakhyurasambuddhau (7.1.92) \*/](#h.xc8cuhx6y0q1)

[/\* anaG sau (7.1.93) and Gicca (1.1.53) \*/](#h.848tmbhvr3u2)

[/\* alo'ntyAtpUrva upadhA (1.1.65) \*/](#h.62ak93jr3v26)

[/\* diva aut (7.1.84) \*/](#h.z3qsglgzxbwm)

[/\* diva ut (6.1.131) \*/](#h.1f41fckra6br)

[/\* zeSo ghyasakhi (1.4.7) \*/](#h.9jczdm5i9286)

[/\* na mu ne (8.2.3) \*/](#h.1luw4pdevh7z)

[/\* yaci bham (1.4.14) and A kaDArAdekA saJjJA (1.4.1) \*/](#h.tnge74t09ev3)

[/\* svAdiSvasarvanAmasthAne (1.4.17) \*/](#h.fsr7s8tqbyva)

[/\* SaTcaturbhyazca (7.1.55) \*/](#h.bnvlo39ywuyl)

[/\* patch for aSTana A vibhaktau \*/](#h.za2re9jjla6e)

[/\* nopadhAyAH (6.4.7) \*/](#h.2wif0jlx3tca)

[/\* pAdaH pat (6.4.130) \*/](#h.160qn5foefaz)

[/\* TeH (6.4.143) \*/](#h.dzoowh45vi0a)

[/\* bhasya TerlopaH (7.1.88) \*/](#h.3jzbyhooj4z4)

[/\* pathimathyRbhukSAmAt (7.1.85) \*/](#h.abxl5vegcvht)

[/\* ito'tsarvanAmasthAne (7.1.85) \*/](#h.61iub7sra0k4)

[/\* tho'nthaH (7.1.87) \*/](#h.to796sevdhzc)

[/\* sambuddhau napuMsakAnAM nalopo vA vAcyaH (vA) \*/](#h.q5u8yxw07wg)

[/\* sau ca (6.4.13) \*/](#h.zaekidfq6f3x)

[/\* inhanpUSAryamNAM zau (6.4.12) \*/](#h.ygtdmu2mnagy)

[/\* sarvanAmasthAne cAsambuddhau (6.4.8) \*/](#h.wd45xlgyp3lz)

[/\* tRjvatkroSTuH (7.1.95), vibhASA tRtIyAdiSvaci (7.1.97) and numaciratRjvadbhAvebhyo nuT pUrvavipratiSedhena (vA 4374) \*/](#h.f70j6kd1hs5z)

[/\* AGo nA'striyAm (7.3.120) \*/](#h.7iszuyyswlcc)

[/\* defining some sets \*/](#h.93lviqf9ziuv)

[/\* iko'ci vibhaktau (7.1.73) \*/](#h.krzviuwe2zue)

[/\* vA'mzasoH (6.4.80) \*/](#h.ye9o3wolrpb4)

[/\* lomno'patyeSu bahuSvakAro vaktavyaH (vA 2560) \*/](#h.hywcao6bjjrs)

[/\* aci ra RtaH (7.2.100) \*/](#h.54dbelvpv4b1)

[/\* RduzanaspurudaMso&nehasAM ca (7.1.94) \*/](#h.e1cflxolru5f)

[/\* Rto GitsarvanAmasthAnayoH (7.3.110) \*/](#h.9p3ir0g9y0dn)

[/\* aptRntRcsvasRnaptRneSTRtvaSTRkSattRhotRpotRprazAstRRNAm (6.4.11) \*/](#h.wfj6l37grunf)

[/\* apo bhi (7.4.48) \*/](#h.fddzfcxo60ax)

[/\* Ato dhAtoH (6.4.140) \*/](#h.cyofpb6iiqsh)

[/\* na saMyogAdvamantAt (6.4.137) \*/](#h.tbnzcnvz3ow0)

[/\* zvayuvamaghonAmataddhite (6.4.133) \*/](#h.dsdo9tgqgvfx)

[/\* vAha UTh (6.4.132) \*/](#h.7obsddawk3x9)

[/\* vasoH samprasAraNam (6.4.131) \*/](#h.6rel267pxn0v)

[/\* samprasAraNAcca (6.1.108) \*/](#h.qcgnc91dna1c)

[/\* na samprasAraNe samprasAraNam (6.2.37) \*/](#h.lpq0cqlgq4s1)

[/\* napuMsakAcca (7.1.19) \*/](#h.e7vqw6pzf60e)

[/\* auGaH zyAM pratiSedho vAcyaH (vA) \*/](#h.6y8gd3hcrspg)

[/\* paddannomAshRnnizasanyUSandoSanyakaJChakannudannAsaJChasprabhRtiSu (6.1.63) special case kakuddoSaNI \*/](#h.qtkhrlioekqm)

[/\* bhasya (6.4.129) and allopo'naH (6.4.134) and vibhASA GizyoH (6.4.236) and na saMyogAdvamantAt (6.4.137) \*/](#h.9sorhs4n5w5g)

[/\* ho hanterJNinneSu (7.3.54) \*/](#h.1qq9xxlkot6h)

[/\* jasi ca (7.3.109) \*/](#h.s7e0g9964gor)

[/\* trestrayaH (7.1.53) \*/](#h.y9lskm35rnk6)

[/\* viSvagdevayozca TeradryaJcatAvapratyaye (6.3.92) \*/](#h.7qyqwiwjwbxc)

[/\* patch for iko yaNaci \*/](#h.q0bjqs9ft2s4)

[/\* adasa au sulopazca (7.2.107) \*/](#h.fomhqf5sw148)

[/\* tyadAdInAmaH (7.2.102) \*/](#h.52v1a049n9ms)

[/\* ajAdyataSTAp (4.1.4) \*/](#h.6zyz5u2jien)

[/\* akaH savarNe dIrghaH (6.1.101) \*/](#h.avb9gf7bk002)

[/\* adasa au sulopazca (7.2.107) \*/](#h.bg6gz9mi5f6p)

[/\* tadoH saH sAvanantyayoH (7.2.106) \*/](#h.921bk47iwhdz)

[/\* sarvanAmnaH smai (7.1.14) \*/](#h.aawfrfrk31ec)

[/\* GasiGyoH smAtsminau (7.1.15) \*/](#h.w87xqdx09xfv)

[/\* pUrvAdibhyo navabhyo vA (7.1.16) \*/](#h.bceqnq4oemqz)

[/\* TAGasiGasAminAtsyAH (7.1.12) \*/](#h.jr824gqnzkw8)

[/\* Ami sarvanAmnaH suT (7.1.15) \*/](#h.ju7v20s3onau)

[/\* sAma Akam (7.1.33) \*/](#h.hl84pf81ouhq)

[/\* dvitIyATaussvenaH (2.4.34) \*/](#h.729mu3j6mmyp)

[/\* idamo'nvAdeze'zanudAttastRtIyAdau (2.4.32) \*/](#h.u95y6xaua0ki)

[/\* nAnarthake'lo'ntyavidhiranabhyAsavikAre (vA 490) \*/](#h.8unoifrmps8q)

[/\* Adyantavadekasmin \*/](#h.jptqrlz3lzl5)

[/\* goto Nit (7.1.90) and oto Niditi vAcyam (vA 5035) \*/](#h.pwiux5czs2lr)

[/\* aco JNiti (7.2.115) \*/](#h.2jyl7lp1x1q0)

[/\* RRta iddhAtoH (7.1.100) \*/](#h.ase0zjscrhxc)

[/\* ambArthanadyorhrasvaH (7.3.103) \*/](#h.8hx4lg3f3q6h)

[/\* idudbhAym (7.3.117) \*/](#h.4905i5lgwjk3)

[/\* GerAmnadyAmnIbhyaH (7.3.116) \*/](#h.avtqp4vyuujg)

[/\* aut (7.3.118) \*/](#h.gqzodjn75msb)

[/\* ANnadyAH (7.3.112) \*/](#h.dx6edsfuhs8n)

[/\* acca gheH (7.3.119) \*/](#h.xsn2cpz1ylh9)

[/\* gherGiti (7.3.111) \*/](#h.4ylonr7ksjd1)

[/\* auGaH ApaH (7.1.18) \*/](#h.614r3346c4h1)

[/\* sambuddhau ca (7.3.106) \*/](#h.vutnkieiqr5x)

[/\* sarvanAmnaH syADDhrasvazca (7.3.114) \*/](#h.214309y2oqgv)

[/\* hali lopaH (7.2.113) \*/](#h.ljxskzdlset9)

[/\* anApyakaH (7.2.112) \*/](#h.ae9ud9prot04)

[/\* AGi cApaH (7.3.105) \*/](#h.20zf0adxl3hm)

[/\* vibhASA diksamAse bahuvrIhau (1.1.28) \*/](#h.khohfeaboqeb)

[/\* yADApaH (7.3.113) \*/](#h.2ln657mm392d)

[/\* nedamadasorakoH (7.1.11) \*/](#h.kw7xixsb6m5)

[/\* jasaH zI (7.1.17) \*/](#h.telun1vk4kz9)

[/\* pUrvaparAvaradakSiNottarAparAdharANi vyavasthAyAmasaJjJAyAm (1.1.34) \*/](#h.jem8vwywn9hf)

[/\* svamajJAtidhanAkhyAyAm (1.1.35) \*/](#h.f3jwjvm0wed1)

[/\* antaraM bahiryogopasaMvyAnayoH (1.1.36) \*/](#h.vjd9rgut7oxy)

[/\* vibhASA jasi \*/](#h.b6zpqmwectth)

[/\* na bahuvrIhau \*/](#h.62tkjfyoi93)

[/\* tRtIyAsamAse \*/](#h.vxdfc4a6jg62)

[/\* dvandve ca \*/](#h.ek18uiikheou)

[/\* prathamacaramatayAlpArdhakatipayanemAzca (1.1.33) \*/](#h.vfi1t7msk4mu)

[/\* dvandve ca (1.1.31) \*/](#h.xywf41yb2rio)

[/\* upadeze'janunAsika it (1.3.2)\*/ // Temporary patch. Not coded perfectly.](#h.klway0g73vuq)

[/\* patch for saMyogAntalopa of maghavan \*/](#h.xbb2d5qxb0qp)

[/\* AdirGiTuDavaH (1.3.5) \*/](#h.b5xblnxnufan)

[/\* cuTU (1.3.7) \*/](#h.f21fpxhtpwgx)

[/\* SaH pratyayasya (1.3.6) \*/](#h.m5illhufakke)

[/\* GeryaH (7.1.13) \*/](#h.r0ne65m4noqc)

[/\* lazakvataddhite (1.3.8) \*/](#h.nb7xvic5p5so)

[/\* na vibhaktau tusmAH (1.3.4) \*/](#h.2tp9hz60bxgj)

[/\* halantyam (1.3.3) and tasya lopaH \*/](#h.hl8zwe80vgnv)

[/\* it additions in case of kvin and kvip \*/](#h.gnlzm3q2um7x)

[/\* nAJceH pUjAyAm (6.4.30) \*/](#h.qi92kiyassv3)

[/\* kruJca \*/](#h.6xr7qxsqzhab)

[/\* defining dhatus with 'i' as it \*/](#h.rx6e7ut89lvt)

[/\* aniditAM hala upadhAyAH kGiti (6.4.24) \*/](#h.yw1ta4qtwr3v)

[/\* samaH sami (6.3.93) \*/](#h.imvd0y4nag8s)

[/\* sahasya sadhriH (6.3.95) \*/](#h.vx8stigzu7vs)

[/\* tirasastiryalope (6.3.94) \*/](#h.e8vzqcwvtrnf)

[/\* atvasantasya cAdhAtoH (6.4.14) \*/](#h.tzk6h5wnj28f)

[/\* ugidacAM sarvanAmasthAne'dhAtoH (7.1.70) \*/](#h.le8zo1l7omfv)

[/\* uda It (6.4.139) \*/](#h.yk2p9fuyrtre)

[/\* acaH (6.4.138) \*/](#h.61jkeb4y3m2n)

[/\* cau (6.3.138) \*/](#h.n1qb4j60zgpe)

[/\* shatR \*/](#h.qg602nk0bimk)

[/\* nAbhyAsAcChatuH (7.1.78) \*/](#h.ogj7zsx2s89t)

[/\* vA napuMsakasya (7.1.79) \*/](#h.6kvxjhl2tc7b)

[/\* zapzyanornityam (7.1.81) \*/](#h.pcazkxuor5e2)

[/\* AcChInadyornum (7.1.80) \*/](#h.720ur51rlji3)

[/\* special message for ikAra of kvip, kvin etc. \*/](#h.ehjlnnoc316r)

[/\* verapRktasya (3.1.67) \*/](#h.eof84zdkm97z)

[/\* GasiGasozca (6.1.110) \*/](#h.cs5otp6r4y4t)

[/\* hrasvanadyApo nuT (7.1.54) \*/](#h.1gc2frn0glc2)

[/\* bahUrji exception to napuMsakasya jhalacaH (7.1.72) \*/](#h.xm0z6d2d8nok)

[/\* beBid, cecCid exception to napuMsakasya jhalacaH (7.1.72) \*/](#h.dnt95asxtjid)

[/\* svap patch \*/](#h.etragc537iul)

[/\* napuMsakasya jhalacaH (7.1.72) \*/](#h.yh6mgdtxfbxb)

[/\* sarvanAmasthAne cAsambuddhau (6.4.8) \*/](#h.x58g50qgqj3g)

[/\* dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118) \*/](#h.cjaz0aty2n9)

[/\* varSAbhvazca (6.8.84) \*/](#h.stvaq5qz8jyc)

[/\* na bhUsudhiyoH (6.4.85) \*/](#h.7zc3q1ddbr8z)

[/\* kvau luptaM na sthAnivat (vA 431) \*/](#h.nbd4zu7acgdr)

[/\* aci znudhAtubhruvAM yvoriyaGuvaGau (6.4.77) \*/](#h.t8tscxgy5has)

[/\* gatikAraketarapUrvapadasya yaN neSyate (vA 5034) \*/](#h.ch7uyiy45wag)

[/\* eranekAco'saMyogapUrvasya (6.4.82) \*/](#h.j4cr0mb9gf9j)

[/\* oH supi (6.4.83) \*/](#h.4p5idp5zin1o)

[/\* ami pUrvaH (6.1.107) \*/](#h.7dmu5ocbckih)

[/\* sAntamahataH saMyogasya (6.4.10) \*/](#h.2q5q7n6v9n9x)

[/\* Rta ut (6.1.111) \*/](#h.br7i8bco5aa1)

[/\* auto'mzasoH (6.1.93) \*/](#h.dqgsqq133ksq)

[/\* ato bhisa ais (7.1.9) \*/](#h.e14up34iyjv9)

[/\* yasyeti ca (6.4.148) \*/](#h.ruw0nspvaj2j)

[/\* do-while loop for sapAdasaptAdhyAyI \*/](#h.j58t39gorny9)

[/\* osi ca (7.3.104) \*/](#h.6o3y94eksyxh)

[/\* na tisRcatasR (6.4.4) \*/](#h.aoxfriinfxlw)

[/\* explanation of prarINAm \*/](#h.kwr16vq9jyq6)

[/\* nAmi (6.4.3) and nR ca (6.4.6) \*/](#h.rhyaw4y7wffg)

[/\* bahuvacane jhalyet (7.3.103) \*/](#h.pft1jcjw38qn)

[/\* supi ca (7.3.102) \*/](#h.hd5ybajmrhuh)

[/\* pragRhya section \*/](#h.mlwhoh5gym5e)

[/\* plutapragRhyA aci nityam (6.1.125) \*/](#h.pzjqes2k39da)

[/\* iko'savarNe zAkalyasya hrasvazca (6.1.127) \*/ // Right now coded for only dIrgha. Clarify wheter the hrasva preceding also included?](#h.wga9tslki3jj)

[/\* upasargAdRti dhAtau (6.1.11) and vA supyApizaleH (6.1.12) \*/](#h.sz4gr8a8q2wm)

[/\* RtyakaH (6.1.128) \*/](#h.bx9quus8a2nb)

[/\* vAkyasya TeH pluta udAttaH (8.2.82) \*/](#h.v7p99jyzhjb)

[/\* pratyabhivAde'zUdre (8.2.83) \*/](#h.hlxnnghakg2g)

[/\* dUrAddhUte ca (8.2.84) \*/](#h.jy3q5gij9u3v)

[/\* haihe prayoge haihayoH (8.2.85) \*/](#h.pawmi3ajs0xe)

[/\* guronanRto'nantyasyApyekaikasya prAcAm (8.2.86) \*/](#h.uvm9iwl3koqo)

[/\* aplutavadupasthite (6.1.129) \*/](#h.uv4ps5i9gwy4)

[/\* I3 cAkravarmaNasya (6.1.130) \*/](#h.38zieurx9fte)

[/\* IdUdeddvivacanaM pragRhyam (1.1.11) \*/](#h.qzlkzwf1x1rb)

[/\* adaso mAt (1.1.12) \*/](#h.apxvmg9003zo)

[/\* ze (1.1.13) \*/](#h.fmapn9dgb7bp)

[/\* nipAta ekAjanAG (1.1.14) \*/](#h.citksyoqry37)

[/\* ot (1.1.15) \*/](#h.vodnsp4rx4p0)

[/\* sambuddhau zAkalyasyetAvanArSe (1.1.16) \*/](#h.twnbiwg0mq29)

[/\* UYaH (1.1.17) \*/](#h.1vzftyl8uh4c)

[/\* U! (1.1.18) \*/](#h.up50fd3o6rby)

[/\* maya uYo vo vA (8.3.33) \*/](#h.s52embp7bqlh)

[/\* IdUtau ca saptamyarthe (1.1.19) \*/](#h.jjypscpf16l0)

[/\* zakandhvAdiSu pararUpaM vAcyam (vA 3632) \*/](#h.8ohqan7y5nma)

[/\* omAGozca (6.1.95) \*/](#h.rmycmmk9qtko)

[/\* ATazca (6.1.90) \*/](#h.kw6cjbhlu6k1)

[/\* striyAH (6.4.79) \*/](#h.oagdpv6qr0rn)

[/\* nAdici (6.1.104) \*/](#h.l6blw7mjzbrk)

[/\* dIrghAjjasi ca (6.1.105) \*/](#h.yia6pn8vf7b)

[/\* prathamayoH pUrvasavarNaH (6.1.102) \*/](#h.vapxy3ixcs4x)

[/\* tasmAcChaso naH puMsi (6.1.103) \*/](#h.v2jl6ktuolk9)

[/\* Rti savarNe R vA (vA 3640) and lRti savarNe lR vA (vA 3641) \*/](#h.9i5qasiazw2y)

[/\* akaH savarNe dIrghaH (6.1.101) \*/](#h.aqsspthz2m8t)

[/\* ato guNe (6.1.17) \*/](#h.9u7odldiilnl)

[/\* hrasvasya guNaH (7.3.108) \*/](#h.jsin74aa2lwh)

[/\* halGyAbbhyo dIrghAtsutisyapRktaM hal (6.1.68) and apRkta ekAlpratyayaH (1.2.41) \*/](#h.bfx8yfm84ypj)

[/\* patch for varShABU sambodhana \*/](#h.rdoohhyh7rx)

[/\* eGhrasvAtsambuddheH (6.1.69) and ekavacanaM sambuddhiH (2.3.49) \*/ // removed the last letter, not as in sutra. Look out for issues if any crops up.](#h.w80txknc4leh)

[/\* dazca (7.2.109) \*/](#h.398baqpraxrw)

[/\* iko yaNaci (6.1.77) \*/](#h.8vnh3w6abv5x)

[/\* sarvatra vibhASA goH (6.1.122) \*/](#h.rxcjtjtdwgff)

[/\* avaG sphoTAyanasya (6.1.123) \*/](#h.96ou51nlkhwn)

[/\* indre ca (6.1.124) \*/](#h.hda0kmyebpb7)

[/\* eGaH padAntAdati (6.1.109) \*/](#h.pslr5cmp84eh)

[/\* eco'yavAyAvaH (7.1.78) \*/](#h.qgeuozkz841x)

[/\* vAnto yi pratyaye (6.1.71), goryutau CandasyupasaMkhyAnam (vA 3543), adhvaparimANe ca (vA 3544) dhAtostannimittasyaiva (6.1.80) \*/](#h.x8qmydc4snkd)

[/\* kSayyajayyau zakyArthe (6.1.81)\*/](#h.fdpi2kmbz3u4)

[/\* krayyastadarthe (6.1.82) \*/](#h.72r0ggj1s7rs)

[/\* khyatyAtparasya (6.1.112) \*/](#h.wj5z8w59fj9b)

[/\* Exceptions to sasajuSo ruH \*/](#h.ni6t74vb35pc)

[/\* etattadoH sulopo'konaJsamAse hali (6.1.132) \*/](#h.ypa8dge496ls)

[/\* so'ci lope cetpAdapUraNam (6.1.134) \*/](#h.57fcqsdo72gt)

[/\* aniditAM patch for sraMs and dhvaMs \*/](#h.q4695r7wwmt9)

[/\* vasusraMsudhvaMsvanaDuhAM daH (8.2.72) \*/](#h.xt5deq5pwsy2)

[/\* saMyogAnta patch for asmad / yuSmad \*/](#h.ft2hgg1r63t8)

[/\* patch for asiddhatva of SakAra of pratyaya as in pipaThiS \*/](#h.jnq8w4li3fxd)

[/\* patch for asiddhatva of SakAra of doS \*/](#h.x0u5laykuvsu)

[/\* sasajuSo ruH (8.2.66) \*/](#h.axaghvi7lqmc)

[/\* ahan(8.2.68) and ro'supi (8.2.69) and rUparAtrirathantareSu vAcyam (vA 4847) \*/](#h.pq6tvxatxl1)

[/\* samaH suTi (8.3.5) \*/ // have used @ as mark of anunAsika u of ru.](#h.z6hpsvrp9900)

[/\* khyAYAdeze na (vA 1591) \*/](#h.mewulie11bjl)

[/\* pumaH khayyampare (8.3.6) \*/](#h.xslf0zp0htm0)

[/\* nazChavyaprazAn (8.3.7) \*/](#h.kwwmoerbmm7e)

[/\* nRUnpe (8.3.10) \*/](#h.pjvgkfacsae3)

[/\* svatavAn pAyau (8.3.11) \*/](#h.kqbfmht2fu7o)

[/\* kAnAmreDite (8.3.12) \*/](#h.qng1rff1paq1)

[/\* ato roraplutAdaplute (6.1.113) \*/](#h.flp12tautt2p)

[/\* hazi ca (6.1.114) \*/](#h.m9m5frb0ppg0)

[/\* ekaH pUrvaparayoH (6.1.84) \*/](#h.mi4186or0946)

[/\* akSAdUhinyAmupasaMkhyAnam (vA 3604) \*/](#h.f81w45ydwjya)

[/\* svAdireriNoH (vA 3606) \*/](#h.6fb94nyrsqzi)

[/\*prAdUhoDhoDyeSaiSyeSu (vA 3605) \*/](#h.tsgaa1ji3gen)

[/\* Rte ca tRtIyAsamAse (vA 3607) \*/](#h.98e6o6j1rpu4)

[/\* pravatsatarakambalavasanadazArNAnAmRNe (vA 3608-9) \*/](#h.61q0rg3pwyne)

[/\* etyedhatyuThsu (6.1.89) \*/](#h.djv1zn7pj6pl)

[/\* AdguNaH (6.1.87) \*/](#h.ygm9e376diy9)

[/\* uraNraparaH (1.1.51) \*/](#h.bvbub3jb3z88)

[/\* eGi pararUpam (6.1.94) \*/](#h.x231di15818b)

[/\* eve cAniyoge (vA 3631) \*/](#h.4qpzfhktzd2r)

[/\* vA supyapizaleH (6.1.92) \*/](#h.qj58bwsu9c45)

[/\* aco'ntyAdi Ti (1.1.64) \*/](#h.kgpdgqtotwvq)

[/\* otvoShThayoH samAse vA (vA 3634) \*/](#h.39a7ue2sub2w)

[/\* nAmreDitasyAntasya tu vA (6.1.99), tasya paramAmreDitam (8.1.2) \*/](#h.lb2w741sqwn5)

[/\* avyaktAnukaraNasyAta itau (6.1.98) \*/](#h.b0ko2r3w3chb)

[/\* vRddhireci (6.1.88) \*/](#h.owzy5soa7joy)

[/\* udaH sthAstambhvoH pUrvasya (8.1.61) \*/](#h.y5dzzk41svl1)

[/\* saMhitAyAm (6.1.72) \*/](#h.w1azjjcjpqq7)

[/\* Che ca (6.1.73) \*/](#h.mddpskgyg382)

[/\* AGmAGozca (6.1.74) \*/](#h.ktotbuc7dcv6)

[/\* dIrghAt (6.1.75) and padAntAdvA (6.1.76) \*/](#h.4mcamr2g7sz5)

[/\* yasmAtpratyayavidhistadAdi pratyaye'Ggam (2.4.13) \*/](#h.4uleave3s6qr)

[/\* creating a + elision for two ++ simultaneously. \*/](#h.n51t0gu3u70y)

[/\* tripAdI functions \*/](#h.9orndlsijgz5)

[/\* na NisambuddhyoH (8.2.8) \*/](#h.xmshh49kig1i)

[/\* patches for maghavA bahulam \*/](#h.cavoq9faib0r)

[/\* NAvuttarapade pratiSedho vaktavyaH (vA 4785) \*/](#h.m2mcnh0v5dy)

[/\* sambuddhau napuMsakAnAM nalopo vA vAcyaH (vA 4786) \*/](#h.flog1cqad7s2)

[/\* na lopaH prAtipadikAntasya (8.2.7) \*/](#h.uw19ryjb6fhg)

[/\* nalopaH supsvarasaJjJAtugvidhiSu kRti (8.2.2) \*/](#h.jvsufdk2a50)

[/\* patch for udan Adeza of udaka \*/](#h.tnuumijtxge1)

[/\* nazervA (8.2.63) \*/](#h.1jssjrk36oem)

[/\* kvinpratyayasya kuH (8.2.62) defining \*/](#h.lvb51xklilo4)

[/\* vrazcabhrasjamRjayajarAjabhrAjacChazAM ca (8.2.35) \*/](#h.p4l2mp6voaz3)

[/\* nimittApAye naimittikasyApyapAyaH (paribhASA) \*/](#h.lbx5bcn1vtf5)

[/\* vivikz patch for overcoming skoH saMyogAdyorante ca \*/](#h.37zwefpocmqv)

[/\* rakS, takS patch to bar application of skoH saMyogAdyorante ca \*/](#h.5zfkdsdjc31c)

[/\* pipak, vivak, didhak patch to bar application of skoH saMyogAdyorante ca \*/](#h.42oak0t3xw56)

[/\* skoH saMyogAdyorante ca (8.2.29) \*/](#h.qtbax0i2ga7n)

[/\* patch for cikIrz \*/](#h.b808wsihvhkx)

[/\* rAtsasya (8.2.24) \*/](#h.beu4o64easa6)

[/\* saMyogAntasya lopaH (8.2.23) \*/](#h.hwzfftjteolb)

[/\* coH kuH (8.2.30) \*/](#h.2439kc6iz49b)

[/\* vA druhamuhaSNuhaSNihAm (8.2.34) \*/](#h.teihgxqe56aq)

[/\* dAderdhAtorghaH (8.2.33) \*/](#h.46btpl7s0uog)

[/\* naho dhaH (8.2.35) \*/](#h.o7sv7xqo0vh6)

[/\* AhasthaH (8.2.36) \*/](#h.46xvcsicne6a)

[/\* ho DhaH (8.2.32) \*/](#h.9mx1f9kdx1ys)

[/\* ekAco bazo bhaS jhaSantasya sdhvoH (8.2.37) \*/ // Not good code. Think hard.](#h.lbvbqqdrk7jy)

[/\* jhalAM jazo'nte (8.2.39) \*/](#h.ceu970a69r2u)

[/\* kvinpratyaya patch for asRj \*/](#h.hwzqvdfnopsa)

[/\* kvinpratyayasya kuH (8.2.62) \*/](#h.5p26lxb8k0oa)

[/\* vizvasya vasurAToH (6.3.128) \*/](#h.2mngez4cetxh)

[/\* bhobhagoaghoapUrvasya yo'zi (8.3.17) : \*/](#h.vrj6dao78gcd)

[/\* SaDhoH kassi (8.2.41) \*/](#h.45el1pnrvnvl)

[/\* mo no dhAtoH (8.2.64) \*/](#h.10p2mtyg36n2)

[/\* rvorupadhAyA dIrgha ikaH (8.2.76) \*/](#h.db8qkewc4tdd)

[/\* hali ca (8.2.77) \*/](#h.rcgkdx8qf9s8)

[/\* eta Idbahuvacane (8.2.81) \*/](#h.v65bbartnhoj)

[/\* adaso'serdAdu do maH (8.2.80) \*/](#h.fpfyz3pu3uvy)

[/\* vyorlaghuprayatnataraH zAkaTAyanasya (8.3.18) \*/](#h.j70wctnrcfre)

[/\* roH supi (8.3.16) \*/](#h.2hu6hzr2r76f)

[/\* kharavasAnayorvisarjanIyaH (8.3.15) \*/](#h.xdihgh18tg5i)

[/\* Dho Dhe lopaH (8.3.13) \*/](#h.l7dpnwhny8s7)

[/\* ro ri (8.3.14) \*/](#h.5jq3lzh12gga)

[/\* Dhralope pUrvasya dIrgho'NaH (6.3.111) \*/](#h.54hnnxqvb4v8)

[/\* lopaH zAkalyasya (8.3.19) and vyorlaghuprayatnataraH zAkaTAyanasya (8.3.18) \*/](#h.spk6ri1awl7r)

[/\* hali sarveSAm (8.3.22) \*/](#h.guw3to37ie9r)

[/\* oto gArgyasya (8.3.20) \*/](#h.wklrb2a8610e)

[/\* uJi ca pade (8.3.21) \*/](#h.cda5esdjfozq)

[/\* mo rAji samaH kvau (8.3.25) \*/](#h.1eqk46x4me7g)

[/\* mo'nusvAraH (8.3.23) \*/](#h.rmjigb4yiw3b)

[/\* nazcApadAntasya jhali (8.3.24) \*/](#h.6r9y8wg7psqf)

[/\* he mapare vA (8.3.26) and yavalapare yavalA veti vaktavyam (vA 4902) \*/](#h.a1yojv32xs4h)

[/\* napare naH (8.3.27) \*/](#h.f3pzlqy68np9)

[/\* GNoH kukTukzari (8.3.28) \*/](#h.48r060odfpth)

[/\* DaH si dhuT (8.3.29) \*/](#h.lqg98jz6gq3h)

[/\* nazca (8.3.30) \*/](#h.433xrhx6455w)

[/\* zi tuk (8.3.31) \*/](#h.t8waef509ze)

[/\* Gamo hrasvAdaci GamuNnityam (8.3.32) \*/](#h.766tkyu9ayq)

[/\* saheH sADaH saH (8.3.56) \*/](#h.tdhh3o5tgxe0)

[/\* sampuGkAnAM so vaktavyaH (vA 4892) \*/](#h.1ory6267lcb)

[/\* samo vA lopameke (bhASya) \*/](#h.uc11yxyq0tu2)

[/\* dvistrizcaturiti kRtvo'rthe (8.3.43) \*/](#h.5hwhrpps111x)

[/\* muhusaH pratiSedhaH (vA 4911) \*/](#h.hph5jcowx69g)

[/\* kaskAdiSu ca (8.3.48) \*/](#h.goyxtx45of13)

[/\* isusoH sAmarthye (8.3.44) and nityaM samAse'nuttarapadasthasya (8.3.45) \*/](#h.gkse0q4d01ze)

[/\* idudupadhasya cApratyayasya (8.3.41) \*/](#h.7pvi2me9yxzf)

[/\* ekAdezazAstranimittikasya na Satvam | kaskAdiSu bhrAtuSputrazabdasya pAThAt (vA 4915) \*/](#h.y4v4bygpsnhq)

[/\* iNaH SaH (8.3.39) \*/](#h.925fblm5mu3s)

[/\* namaspurasorgatyoH (8.3.40) \*/](#h.r9zbr8fzldlt)

[/\* tiraso'nyatarsyAm (8.3.42) \*/](#h.sy3ghkkf491l)

[/\* ataH kRkamikaMsakumbhapAtrakuzAkarNISvanavyayasya (8.3.46) \*/](#h.f76koeslj6wy)

[/\* adhazzirasI pade \*/](#h.de8qxk4s3dmj)

[/\* so'padAdau (8.3.38), pAzakalpakakAmyeSviti vAcyam (vA 5033), anavyayasyeti vAcyam (vA 4902) and kAmye roreveti vAcyam (vA 4902) \*/](#h.6umufykgqqf)

[/\* zarpare visarjanIyaH (8.3.35) \*/](#h.301gr8hhdehe)

[/\* kupvoH &k&pau ca (8.3.37) \*/](#h.tqsocq6ovefx)

[/\* visarjanIyasya saH (8.3.34) \*/](#h.hrvmhj5gk1oy)

[/\* vA zari (8.3.36) \*/](#h.7y9q4y3uyiw1)

[/\* kharpare zari vA visargalopo vaktavyaH (vA 4906) \*/](#h.qzbcewn4lk3e)

[/\* apadAntasya mUrdhanyaH (8.3.55), iNkoH (8.3.57) and AdezapratyayayoH (8.3.59) \*/](#h.q7czbdb6v1io)

[/\* numvisarjanIyazarvyavAye'pi (8.3.58) \*/](#h.ydqj2yta73ya)

[/\* hanteratpUrvasya (8.4.22) \*/](#h.bexlh8taian0)

[/\* raSAbhyAM no NaH samAnapade (8.4.1) \*/](#h.y2no58t8hz3r)

[/\* ekAjuttarapade NaH (8.4.12) \*/](#h.4bvy7r2r3rvt)

[/\* Patch to remove the + sign \*/](#h.m05csg7ch6jp)

[/\* aTkupvAGnumvyavAye'pi (8.4.2) and na padAntasya (8.4.37) \*/](#h.hr0awxaiha5s)

[/\* RvarNAnnasya NatvaM vAcyam (vA 4969) \*/](#h.ygmj0m7qunl1)

[/\* na padAntasya 8.4.37) \*/](#h.qg9g524630z9)

[/\* stoH zcunA zcuH (8.4.40) \*/](#h.7h60ewxk0u39)

[/\* stoH zcunA zcuH (8.4.40) and zAt (8.4.44) \*/](#h.njf1mgkvk18w)

[/\* anAmnavatinagarINAmiti vAcyam (vA 5016) \*/](#h.re1fm0h8ecga)

[/\* stoH STunA STuH (8.4.41) and na padAntATToranAm (8.4.41) and toH Si (8.4.43) \*/](#h.6e1106audcaj)

[/\* yaro'nunAsike'nunAsiko vA (8.4.45) \*/](#h.dfktivh1hj7z)

[/\* nAdinyAkroze putrasya (8.4.48) \*/](#h.wilcayss3vey)

[/\* vA hatajagdhayoH (vA 5022) \*/](#h.wu1d2oyve85c)

[/\* cayo dvitIyAH zari pauSkarasAderiti vAcyam (vA 5023) \*/](#h.4pj355lrui1i)

[/\*anaci ca (8.4.47)\*/](#h.o996hu88igdm)

[/\* By anaci ca (according to mahAbhASya example of vAkk) \*/](#h.p9hnfqzdmu79)

[/\* zaraH khayaH (vA 5019) \*/](#h.eo8quv4mcdiv)

[/\* aco rahAbhyAM dve (8.4.46) \*/](#h.annm7y30ysqy)

[/\* triprabhRtiSu zAkaTAyanasya (8.4.50)\*/](#h.u0rw8xvyirw4)

[/\* sarvatra zAkalyasya (8.4.51) \*/](#h.5izv1sy9f96m)

[/\* dIrghAdAcAryANAm (8-4-52) \*/](#h.3jjctfbsbo1k)

[/\* jhalAM jaz jhaSi (8.4.53) \*/](#h.elee7dargheh)

[/\* yaNo mayo dve vAcye (vA 5018) yaN in paJcamI and may in SaSThI)\*/](#h.d3qytx1p57zm)

[/\* yaNo mayo dve vAcye (vA 5018) may in paJcamI and yaN in SaSThI)\*/](#h.grbodwvkwf21)

[/\* vA'vasAne (8.4.54) \*/](#h.n5280kdu9g2)

[/\* khari ca (8.4.55) \*/](#h.8ca2cthyoovx)

[/\* aNo'pragRhyasyAnunAsikaH (8.4.57) \*/](#h.rj714fem5nhs)

[/\* anusvArasya yayi parasavarNaH (8.4.58) and vA padAntasya (8.4.59) \*/](#h.cc2dlz8zkh5n)

[/\* torli (8.4.60) \*/](#h.h390h25qogwm)

[/\* jhayo ho'nyatarasyAm (8.4.62) \*/](#h.1phb5y5ewf7d)

[/\* zazCho'Ti (8.4.63) and ChatvamamIti vAcyam (vA 5025) \*/](#h.xn153mi17hn4)

[/\* halo yamAM yami lopaH (8.4.64) \*/](#h.7rbr16czcbx3)

[/\* jharo jhari savarNe (8.4.65) \*/](#h.tgjf95xrrnm)

[/\* Final Display \*/](#h.5deykgn67r1x)

[/\* setting the $pada back to pratyaya for next use \*/](#h.2johpcnfbhcn)

# 

# How to use the code:

## For using the code online:

Use<http://lanover.com/lan/sanskrit/sandhi.html> for sandhi generation. use<http://lanover.com/lan/sanskrit/subanta.html> for subanta generation.

## For using the code offline on your computer:

1. Download the latest version of XAMPP from<https://www.apachefriends.org/index.html>

2. Install XAMPP.

3. Locate XAMPP directory. Usually it is c://xampp

4. go to xampp/htdocs directory. 5. Create a directory named sanskrit.

6. Copy paste the files mentioned in the next section in sanskrit directory. (This is one time job). 6. Go to xampp folder and click on xampp-control.

7. Click start for Apache, MySQL, Filezilla. (You will have to do steps 6 and 7 every time you start your computer. This functions as server for your html and php files.)

## For Sandhi generation:

This is an attempt to create an open source sandhi joiner.

The text used is siddhAntakaumudI.

Install sandhi.html, sandhi.php, dev-slp.php, slp-dev.php, function.php and the mystyle.css in your localhost and run the sandhi.html.

Input your words - it will fetch the output from the PHP files and display it there.

### For offline usage:

Please run localhost/sanskrit/sandhi.html from your browser.

## For Subanta generation:

Install dev-slp.php, slp-dev.php, function.php, mystyle.css, subanta.html, subanta.php, ajax.php and script.js in your system and then input a word. Give the feedback if the machine asks for any. Then machine will give you derivation of all 21 vibhakti sets of that particular word.

### For offline usage:

Please run localhost/sanskrit/subanta.html from your browser.

# Code location:

All above are available under GNU licence on <https://github.com/drdhaval2785/sanskrit>

# Notes on different files used:

## 1. ajax requirements, ajax.php, script.js

These files are discussed together because they deal with the user feedback part of our code. The reference numbers used will be the same as ajax requirememt.docx file.

User feedback can be of two types.

The first type is where the feedback is taken, because the declention depends on the intention of user. This is non negotiable type of feedback in the sense that even in future we won’t be able to do away with them. I won’t be writing anything extra for these feedback. Just will place a ‘no’ besides it.

The second type is where we don’t know (as of now) how to decide some parameter. e.g. right now we are asking the user whether the word is Abanta or not. In future, when the machine learns how to differentiate a word having Abanta, we will no longer need this feedback. In that sense, this type is negotiable. As the machine progresses, these feedback points can be removed. I will try to note down the second type of feedback with a ‘yes and how we can remove them in future.

### 1. पुल्लिङ्ग

1.1.1.1. सर्वादयः सञ्ज्ञा के तौर पे प्रयुक्त हुए हैं ? - no

1.1.1.2. सर्वादयः उपसर्जनीभूत हैं ? - no.

1.1.1.3. सर्वादयः तृतीया तत्पुरुष समास में प्रयुक्त हुए हैं ? - yes. When samAsa machine gets matured.

1.1.1.4. सर्वादयः द्वन्द्व समास में प्रयुक्त हुए हैं ? - yes. samAsa machine.

1.1.1.5. सर्वादयः बहुव्रीहि समास में प्रयुक्त हुए हैं ? - yes. samAsa machine.

1.1.1.5.1. दिक्समास है - no

1.1.1.5.2. दिक्समास नहीं है - no

1.1.1.6. उपर से किसी में नहीं प्रयुक्त हुए हैं - not applicable.

1.1.1.6.1. The word ends with any of the following. $purva = array("pUrva","para","avara","dakziRa","uttara","apara","aDara");

1.1.1.6.1.1. सञ्ज्ञा है या व्यवस्था के अर्थ में नहीं है - no

1.1.1.6.1.2. सञ्ज्ञा नहीं है और व्यवस्था के अर्थ में प्रयुक्त हुए हैं । - no

1.1.1.6.2. The word ends with $sva = array("sva",);

1.1.1.6.2.1. ज्ञाति या धन के अर्थ में प्रयुक्त हुआ है - no

1.1.1.6.2.2. ज्ञाति या धन के अर्थ में प्रयुक्त नहीं हुआ है - no

1.1.1.6.3. The word ends with $antara = array(“antara”);

1.1.1.6.3.1. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त हुआ है - no

1.1.1.6.3.1.1. उसके बाद का शब्द पुरि है ? - yes. When we start analysing sentences instead of words.

1.1.1.6.3.1.2. नहीं - same as above

1.1.1.6.3.2. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त नहीं हुआ है - no

1.1.1.6.4. The word ends with $datara = array(“atara”, “atama”); and !== “anyatama”

1.1.1.6.4.1. डतर / डतम प्रत्यय से शब्द की सिद्धि हुई है - yes. When pratyaya machine is mature.

1.1.1.6.4.2. नहीं - same as above.

1.1.1.6.5. The word ends with $sama = array(“sama”);

1.1.1.6.5.1. सम शब्द सर्व के पर्याय में प्रयुक्त हुआ है - no

1.1.1.6.5.2. सम शब्द तुल्य के पर्याय में प्रयुक्त हुआ है - no

1.2. The word entered ends with “A” (आ).

1.2.1. क्या यह आकारान्त धातु है ? - yes. When dhAtu, kvibanta, kvinnanta, nAmadhAtu etc every type of dhAtu is known

1.2.2. नहीं - not applicable

1.3. The word entered ends with “i” (इ)

1.3.1. The word entered ends with “saKi” (सखि)

1.3.1.1. प्राधान्य – जैसे कि सुसखा - no. There is a very faint chance that samAsa machine may handle this properly. Maybe survey of existing corpus will have to be done. That may help to get examples, but can never be exhaustive.

1.3.1.2. उपसर्जनीभूत – जैसे कि परमसखा - no.

1.3.1.3. लाक्षणिक – जैसे कि अतिसखिः (सखीमतिक्रान्तः) - yes. These seem to be very typical grammatical words. They / their pattern can be enumerated by some grammarian.

1.3.2. The word entered ends with ‘dvi’ (द्वि)

1.3.2.1. सञ्ज्ञा है या उपसर्जनीभूत है - no

1.3.2.2. सञ्ज्ञा नहीं है और उपसर्जनीभूत नहीं है - no

1.4. The word entered ends with ‘I’ (ई) ->This is a bit complicated. Maybe we should think of something like check boxes.

1.4.1. अनदीसञ्ज्ञकः अधातुः – वातप्रमी (इको यणचि, अमि पूर्वरूपम्‌, शसि पूर्वसवर्णदीर्घः, ङौ सवर्णदीर्घः) - yes. dhAtu - When dhAtu, kvibanta, kvinnanta, nAmadhAtu etc every type of dhAtu is known.

nadI - when nityastrIliGga and strIpratyaya prakaraNa is known.

GI - when strIpratyaya machine is over.

But still the difficulty would be, both vAtapramI words will look the same. So, difficult to segregate without human intervention.

1.4.2. नदीसञ्ज्ञकः अधातुः (सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, इको यणचि, अमि पूर्वरूपम्, शसि पूर्वसवर्णदीर्घः, ङौ सवर्णदीर्घः)

1.4.2.1. ङ्‍यन्त – बहुश्रेयसी (सुलोपः) - yes.

1.4.2.2. अङ्‍यन्त – अतिलक्ष्मीः (न सुलोपः) - yes. feminine words with an IkAra without ‘GI’ pratyayas are very less. Mostly we can enumerate from uNAdisUtras.

1.4.3. अनदीसञ्ज्ञकः धातुः (असंयोगपूर्वकः इवर्णः, अनेकाच्‌ अङ्गम्‌) – प्रधी, वातप्रमी (क्विबन्त)

(अजादिषु प्रत्ययेषु एरनेकाचो…इति यणादेशः) -

1.4.3.1. नीधात्वन्त – उन्नीः, ग्रामणीः (ङेरामादेशः) - yes. Will have to combine sandhi with nI dhAtu.

1.4.3.2. सखा – (सखायमिच्छति) – (अनङ्‌णित्त्वे, ङसिङसोरुत्त्वम्‌) - no

1.4.3.3. सखीः, सुखीः, सुतीः, लूनीः, क्षामीः, प्रस्तीमीः (ख्यत्यात्परस्य इत्यनेन ङसिङसोरुत्त्वम्‌) - yes. I think these cases can be enumerated.

1.4.4. नदीसञ्ज्ञकः धातुः (असंयोगपूर्वकः इवर्णः, अनेकाच्‍ अङ्गम्‌)

(सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, अजादिषु प्रत्ययेषु एरनेकाचो…इति यणादेशः)

1.4.4.1. ङ्‍यन्त – कुमारी (ङ्‍यन्तत्वात्‌ सुलोपः) - yes.

1.4.4.2. अङ्‍यन्त – प्रकृष्टा धीर्यस्य सः प्रधीः – (अङ्‍यन्तत्वान्न सुलोपः) - yes

1.4.5. अनदीसञ्ज्ञकः धातुः (संयोगपूर्वकः इवर्णः / एकाच्‌ अङ्गम्‌ / गतिकारकेतरपूर्वकः) या सुधीशब्दः

(अजादिषु प्रत्ययेषु ‘अचिश्नुधातुभ्रुवां.. इतीयङादेशः) - yes. when machine learns gati and kAraka properly.

1.4.5.1. नीः (ङेरामादेशः) - yes

1.4.5.2. अङ्‍यन्त - यवक्रीः, सुश्रीः (अङ्‍यन्तत्वान्न सुलोपः), सुधी:, शुष्कीः, पक्वीः इत्यादि - yes.

1.4.6. नदीसञ्ज्ञकः धातुः (संयोगपूर्वकः इवर्णः / एकाच्‌ अङ्गम्‌ / गतिकारकेतरपूर्वकः ) – शुद्धधीः, परमधीः, दुर्धीः, वृश्चिकभीः

(सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, अजादिषु प्रत्ययेषु इयङादेशः)

1.5. The word ends with U (ऊ).

1.5.1. अनदीसञ्ज्ञकः अधातुः – हूहू (इको यणचि, अमि पूर्वरूपम्‌, शसि पूर्वसवर्णदीर्घः) - yes. enumerate UkArAnta adhAtu prAtipadikas. I think they are quite rare.

1.5.2. नदीसञ्ज्ञकः अधातुः – अतिचमूः (सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, इको यणचि, अमि पूर्वरूपम्, शसि पूर्वसवर्णदीर्घः, ङौ सवर्णदीर्घः) - yes. when nadI is known

1.5.3. धातुः (असंयोगपूर्वकः उवर्णः, अनेकाच्‌ अङ्गम्‌ , गतिकारकपूर्वकः) – खलपूः, सुलूः, उल्लूः(अजादिषु प्रत्ययेषु ओः सुपि इति यणादेशः) - yes. when gati, kAraka, dhAtu are known

1.5.4. धातुः (संयोगपूर्वकः उवर्णः / एकाच्‌ अङ्गम्‌ / गतिकारकेतरपूर्वकः) – कटप्रूः, लूः, परमलूः( अजादिषु प्रत्ययेषु ‘अचिश्नुधातुभ्रुवां.. इत्युवङादेशः) - yes. when gati, kAraka, dhAtu are known.

1.6. The word ends with ‘catur’ (चतुर्‌)

1.6.1. $second=== “Am” and $so!== “catur”

1.6.1.1. उपसर्जनीभूत - no

1.6.1.2. प्राधान्य - no

1.7. The word is “idam” / “idakam”

1.7.1. अन्वादेश है - yes. when we start analysing sentences instead of words.

1.7.2. अन्वादेश नहीं - yes.

1.8. The word ends with ‘n’

The word ends with array(“paYcan”, “saptan”, “zwan”, “navan”, ‘daSan”); and is not equal to them.

1.8.1. समास में नान्त शब्द उपसर्जनीभूत है ? - no

1.8.2. उपसर्जनीभूत नहीं है । - no

1.9. The word ends with ‘j’

1.9.1. क्विन्‌ प्रत्ययान्त है ? - no. kvip and kvin pratyayAnta seem the same to me now. See later.

1.9.1.1. ends with ‘yuj’

1.9.1.1.1. युजि - no

1.9.1.1.2. युजिर्‌ - no

1.9.2. क्विन्‍ प्रत्ययान्त नहीं है । - no.

1.9.3. ends with array(“BrAj”)

1.9.3.1. भ्राज्‌ धातु फणादि है ? - no

1.9.3.2. नहीं - no

1.10. The word ends with $tyadadi = array("dvi","tyad","tad","etad","idam","adas","eka","idakam");

1.10.1. त्यदादि शब्द सञ्ज्ञा या उपसर्जनीभूत के रूप में प्रयुक्त हुए हैं ? - no

1.10.2. त्यदादि शब्द सञ्ज्ञा या उपसर्जनीभूत नहीं है । - no

1.10.2.1. The word belongs to array(“etad”, “idam”);

1.10.2.1.1. अन्वादेश है ? - yes. When we treat sentences.

1.10.2.1.2. अन्वादेश नहीं है । - yes

1.11. The word ends with array(“smad”,’yuzmad”) and !in\_array(“asmad”, “yuzmad”)

1.11.1. अस्मद्‌ / युष्मद्‌ एकत्ववाची है - no

1.11.2. अस्मद्‌ / युष्मद्‌ द्वित्ववाची है - no

1.11.3. अस्मद्‌ / युष्मद्‌ बहुत्ववाची है - no

1.12. The word is in\_array(‘asmad’, ‘yuzmad’) &&in\_array($so,array(‘am’, ‘Ow’, ‘Sas’, ‘Ne’, ‘ByAm’, ‘Byas’, ‘Nas’, ‘os’, ‘Am’)

1.12.1. अस्मद्‌ / युष्मद्‌ का प्रयोग पद के बाद में नहीं हुआ है । (निषेध) - yes. When we analyse sentences instead of words.

1.12.2. अस्मद्‌ / युष्मद्‌ का प्रयोग पाद के प्रारम्भ में हुआ है । (निषेध) - yes. When we analyse sentences instead of words.

1.12.3. अस्मद्‌ / युष्मद्‌ का च, वा, हा, अह या एव से साक्षात्‌ योग है । (निषेध) yes. When we analyse sentences instead of words.

1.12.4. अस्मद्‌ / युष्मद्‌ का अचाक्षुष ज्ञानार्थ धातु से योग है । (निषेध) - yes. Enlist jJAnArtha dhAtus. Possible when we analyse sentences.

1.12.5. उपरमें से कोई नहीं - not applicable.

1.12.5.1. अन्वादेश है - yes.

1.12.5.1.1. विद्यमानपूर्व प्रथमान्त शब्द के परे अस्मद्‌ / युष्मद्‌ हैं

1.12.5.1.1.1. आमन्त्रित से परे हैं - yes. When we explain what is Amantrita.

1.12.5.1.1.1.1. विशेष्य से परे समानाधिकरण विशेषण है - yes. when we analyse sentences.

1.12.5.1.1.1.1.1. बहुवचन है (विभाषा) - yes. with help of tiGantas, we can analyse vacana.

1.12.5.1.1.1.1.2. बहुवचन नहीं है (नित्य) - yes.

1.12.5.1.1.1.2. विशेष्य से परे समानाधिकरण विशेषण नहीं है (निषेध)

1.12.5.1.1.2. आमन्त्रित से परे नहीं है (विभाषा)

1.12.5.1.2. अस्मद्‌/ युष्मद्‌ विद्यमानपूर्व प्रथमान्त शब्द के परे नहीं हैं (नित्य)

1.12.5.2. अन्वादेश नहीं है (विभाषा)

1.13. The word ends with array(“aYc”, “AYc”, “anc”, “Anc) as the first word.

1.13.1. क्विन्‌ प्रत्ययान्त है - yes. Check rules which specify kvin after aYcu verb.

1.13.1.1. पूजाके अर्थ में प्रयुक्त है ? - no

1.13.1.2. पूजा के अर्थ में प्रयुक्त नहीं है । - no

1.13.2. क्विन् प्रत्ययान्त नहीं है - no

1.14. The first word ends with array(“Bavat”).

1.14.1. भाते र्डवतु ? - no

1.14.2. भूधातोः शतृप्रत्ययः ? - no

1.15. The first word ends with array(“takz”, “rakz”)

1.15.1. ण्यन्त - yes. When machine understands the difference between Nyanta and non Nyanta.

1.15.2. अण्यन्त - yes.

1.16. The first word ends with ‘m’

1.16.1. धातु - yes. when dhAtus are learnt

1.16.2. अधातु - no

1.17. The word ends with ‘at’

1.17.1. अत्वन्त है - yes. when machine learns pratyayas

1.17.2. अत्वन्त नहीं है - yes

1.17.2.1. अभ्यस्त है - yes. when machine learns the process to create abhyastas.

1.17.2.2. अभ्यस्त नहीं है - yes

1.18. The word ends with ‘zaz’ and not equal to ‘zaz’

1.18.1. प्राधान्य - no

1.18.2. उपसर्जनीभूत - no

### 2. स्त्रीलिङ्ग

2.1. The word ends with ‘A’ (आ)

2.1.1. शब्द आबन्त नहीं है – जैसे कि गोपा ? - yes. After strIprakaraNa is ready

2.1.2. शब्द आबन्त है – जैसे कि रमा - yes

2.1.2.1. Ends with $sarvanamastri = array("sarvA","viSvA","uBA","uBayA","atarA","atamA","anyA","anyatarA","itarA","tvA","nemA","simA","pUrvA","parA","avarA","dakziRA","uttarA","aparA","aDarA","svA","antarA","ekA","dvA",);

2.1.2.1.1. सर्वादयः सञ्ज्ञा के तौर पे प्रयुक्त हुए हैं ? - no

2.1.2.1.2. सर्वादयः उपसर्जनीभूत हैं ? - no

2.1.2.1.3. सर्वादयः तृतीया तत्पुरुष समास में प्रयुक्त हुए हैं ? - yes

2.1.2.1.4. सर्वादयः द्वन्द्व समास में प्रयुक्त हुए हैं ? - yes

2.1.2.1.5. सर्वादयः बहुव्रीहि समास में प्रयुक्त हुए हैं ? - yes

2.1.2.1.5.1. दिक्समास है - no

2.1.2.1.5.2. दिक्समास नहीं है - no

2.1.2.1.6. उपर से किसी में नहीं प्रयुक्त हुए हैं - not applicable

2.1.2.1.6.1. The word ends with any of the following. $purva = array("pUrva","para","avara","dakziRa","uttara","apara","aDara");

2.1.2.1.6.1.1. सञ्ज्ञा है या व्यवस्था के अर्थ में नहीं है - no

2.1.2.1.6.1.2. सञ्ज्ञा नहीं है और व्यवस्था के अर्थ में प्रयुक्त हुए हैं । - no

2.1.2.1.6.2. The word ends with $sva = array("sva",);

2.1.2.1.6.2.1. ज्ञाति या धन के अर्थ में प्रयुक्त हुआ है - no

2.1.2.1.6.2.2. ज्ञाति या धन के अर्थ में प्रयुक्त नहीं हुआ है - no

2.1.2.1.6.3. The word ends with $antara = array(“antara”);

2.1.2.1.6.3.1. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त हुआ है - no

2.1.2.1.6.3.1.1. उसके बाद का शब्द पुरि है ? - yes

2.1.2.1.6.3.1.2. नहीं - not applicable

2.1.2.1.6.3.2. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त नहीं हुआ है - no

2.1.2.1.6.4. The word ends with $datara = array(“atara”, “atama”); and !== “anyatama”

2.1.2.1.6.4.1. डतर / डतम प्रत्यय से शब्द की सिद्धि हुई है - yes.

2.1.2.1.6.4.2. नहीं - yes

2.1.2.1.6.5. The word ends with $sama = array(“sama”);

2.1.2.1.6.5.1. सम शब्द सर्व के पर्याय में प्रयुक्त हुआ है - no

2.1.2.1.6.5.2. सम शब्द तुल्य के पर्याय में प्रयुक्त हुआ है - no

2.2. The word ends with ‘i’ (इ)

2.2.1. Ends with ‘tri’ and is not ‘tri’

2.2.1.1. त्रि शब्द स्त्रीलिङ्ग में है - no

2.2.1.2. त्रि शब्द स्त्रीलिङ्ग में नहीं है - no. depends on user’s wish

2.3. The word ends with ‘I’ (ई)

2.3.1. ङ्‍यन्त, नदीसञ्ज्ञक (गौरी) - yes

2.3.2. अङ्‍यन्त, नदीसञ्ज्ञक (लक्ष्मी) - yes

2.3.3. ङ्‍यन्त, इयङ्स्थानिक (स्त्री) - yes. Better coding for iyaG, uvaG etc are needed. Current coding is more like a patch. That coding will need dhAtu, kvip, kvin etc. So pending right now. right now working with user feedback.

2.3.4. अङ्‍यन्त, इयङ्स्थानिक (श्री) - yes

2.3.5. प्रधीशब्द

2.3.5.1. प्रध्यायतीति प्रधीः - no

2.3.5.1.1. वृत्तिकारादीनां मते लक्ष्मीवद्रूपम्‌ - not applicable

2.3.5.1.2. कैयटमते तु पुंवद्रूपम्‌ - not applicable

2.3.5.2. प्रकृष्टा धीः (लक्ष्मीवद्‌, अमि शसि च प्रध्यम्‌, प्रध्यः इति विशेषः) - no

2.3.6. सुधी

2.3.6.1. सुष्ठु धीः यस्याः सा, सुष्ठु ध्यायतीति - no

2.3.6.1.1. वृत्तिकारमते श्रीवत्‌ - not applicable

2.3.6.1.2. मतान्तरे पुंवत्‌ - not applicable

2.3.6.2. सुष्ठु धीः (श्रीवत्‌) - no

2.3.7. ग्रामणीः (पुंवत्‌) - not applicable

2.4. Ends with ‘U’ (ऊ)

2.4.1. नदीसञ्ज्ञक (वधूः) – गौरीवत्‌ - yes

2.4.2. इयङ्स्थानिक (भ्रूः) – श्रीवत्‌ - yes

2.4.3. न नदीसञ्ज्ञक (खलपूः) – पुंवत्‌ - yes

2.4.4. पुनर्भूः, दृन्भूः, पुनर्भूः – दृन्करपुनः... से यण्‌ तथा नदीकार्य - not applicable

2.4.4.1. $so===”Am”

2.4.4.1.1. समास है

2.4.4.1.1.1. समास का उत्तरपद एकाच्‌ है और पूर्वपद में रकार या षकार है - yes. After samAsa machine is mature.

2.4.4.1.1.2. समास का उत्तरपद एकाच्‌ नहीं है या पूर्वपद में रकार या षकार नहीं है - yes

2.4.4.1.2. समास नहीं - yes

2.4.5. वर्षाभूः - not applicable

2.4.6. स्वयम्भूः - नित्यस्त्रीलिङ्गत्वाभावात्‌ न नदीसञ्ज्ञा । अङ्‍यन्तत्वान्न सुलोपः । - not applicable

2.5. The word is idam

2.5.1. अन्वादेश है - yes. When we start handling sentences.

2.5.2. अन्वादेश नहीं है - yes

2.6. The word is diS

2.6.1. क्विन्‌ प्रत्ययान्त है - no

2.6.2. क्विप्‌ प्रत्ययान्त है - no

### 3. नपुंसकलिङ्ग

3.1. ends with $ik=array(“i”, “I”, “u”, “U”, “f”, “F”, “x”, “X”);

3.1.1. भाषितपुंस्क है ? - yes. Will have to verify from dictionary databases for gender, whether it is bhASitapuMska or not. But samAsas will create problem then. Partly doable.

3.1.2. भाषितपुंस्क नहीं है । - yes

3.2. The word is equal to “idam”,”etad”

3.2.1. अन्वादेश - yes

3.2.2. अन्वादेश नहीं है - yes

3.3. The word ends with “at”,”At”

3.3.1. अभ्यास के परे शतृप्रत्ययान्त शब्द है - yes

3.3.2. नहीं - yes

3.3.2.1. अवर्णान्त अङ्ग के परे शतृ प्रत्यय का अवयव है

3.3.2.1.1. शप्‌ / श्यन्‌ प्रत्यय के अवर्ण के परे शतृ प्रत्यय का अवयव है - yes. When zap/zyan is taught

3.3.2.1.2. नहीं - yes

3.3.2.2. नहीं - yes

# Actual codes used:

## 1. ajax requirements

This is the documentation where we take user feedback. They are constraints where the derivation is based on user’s intention or some modalities which the machine is unable to handle at this particular point of time. e.g. We take feedback whether it is nadI saJjJA or not. We take this feedback right now, but as the machine progresses and we can decide what are nityastrIliGgas and what are not, we can do away with this feedback. Machine may decide on its own. Posting the user feedback places for ready reference here.

### 1. पुल्लिङ्ग

1.1. The word entered ends with “a” (अ).

1.1.1. The word ends with any of the following. $sarvanama = array("sarva","viSva","uBa","uBaya","pUrva","para","avara","dakziRa","uttara","apara","aDara","atara","atama","sva",“antara”,"anya","anyatara","itara","tvat","tva","nema","sama”,“sima","sva","antara","tyad","tad","yad","etad","idam","adas","eka","dvi","yuzmad","asmad","Bavat","kim");

1.1.1.1. सर्वादयः सञ्ज्ञा के तौर पे प्रयुक्त हुए हैं ?

1.1.1.2. सर्वादयः उपसर्जनीभूत हैं ?

1.1.1.3. सर्वादयः तृतीया तत्पुरुष समास में प्रयुक्त हुए हैं ?

1.1.1.4. सर्वादयः द्वन्द्व समास में प्रयुक्त हुए हैं ?

1.1.1.5. सर्वादयः बहुव्रीहि समास में प्रयुक्त हुए हैं ?

1.1.1.5.1. दिक्समास है

1.1.1.5.2. दिक्समास नहीं है

1.1.1.6. उपर से किसी में नहीं प्रयुक्त हुए हैं

1.1.1.6.1. The word ends with any of the following. $purva = array("pUrva","para","avara","dakziRa","uttara","apara","aDara");

1.1.1.6.1.1. सञ्ज्ञा है या व्यवस्था के अर्थ में नहीं है

1.1.1.6.1.2. सञ्ज्ञा नहीं है और व्यवस्था के अर्थ में प्रयुक्त हुए हैं ।

1.1.1.6.2. The word ends with $sva = array("sva",);

1.1.1.6.2.1. ज्ञाति या धन के अर्थ में प्रयुक्त हुआ है

1.1.1.6.2.2. ज्ञाति या धन के अर्थ में प्रयुक्त नहीं हुआ है

1.1.1.6.3. The word ends with $antara = array(“antara”);

1.1.1.6.3.1. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त हुआ है

1.1.1.6.3.1.1. उसके बाद का शब्द पुरि है ?

1.1.1.6.3.1.2. नहीं

1.1.1.6.3.2. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त नहीं हुआ है

1.1.1.6.4. The word ends with $datara = array(“atara”, “atama”); and !== “anyatama”

1.1.1.6.4.1. डतर / डतम प्रत्यय से शब्द की सिद्धि हुई है

1.1.1.6.4.2. नहीं

1.1.1.6.5. The word ends with $sama = array(“sama”);

1.1.1.6.5.1. सम शब्द सर्व के पर्याय में प्रयुक्त हुआ है

1.1.1.6.5.2. सम शब्द तुल्य के पर्याय में प्रयुक्त हुआ है

1.2. The word entered ends with “A” (आ).

1.2.1. क्या यह आकारान्त धातु है ?

1.2.2. नहीं

1.3. The word entered ends with “i” (इ)

1.3.1. The word entered ends with “saKi” (सखि)

1.3.1.1. प्राधान्य – जैसे कि सुसखा

1.3.1.2. उपसर्जनीभूत – जैसे कि परमसखा

1.3.1.3. लाक्षणिक – जैसे कि अतिसखिः (सखीमतिक्रान्तः)

1.3.2. The word entered ends with ‘dvi’ (द्वि)

1.3.2.1. सञ्ज्ञा है या उपसर्जनीभूत है

1.3.2.2. सञ्ज्ञा नहीं है और उपसर्जनीभूत नहीं है

1.4. The word entered ends with ‘I’ (ई) ->This is a bit complicated. Maybe we should think of something like check boxes.

1.4.1. अनदीसञ्ज्ञकः अधातुः – वातप्रमी (इको यणचि, अमि पूर्वरूपम्‌, शसि पूर्वसवर्णदीर्घः, ङौ सवर्णदीर्घः)

1.4.2. नदीसञ्ज्ञकः अधातुः (सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, इको यणचि, अमि पूर्वरूपम्, शसि पूर्वसवर्णदीर्घः, ङौ सवर्णदीर्घः)

1.4.2.1. ङ्‍यन्त – बहुश्रेयसी (सुलोपः)

1.4.2.2. अङ्‍यन्त – अतिलक्ष्मीः (न सुलोपः)

1.4.3. अनदीसञ्ज्ञकः धातुः (असंयोगपूर्वकः इवर्णः, अनेकाच्‌ अङ्गम्‌) – प्रधी, वातप्रमी (क्विबन्त)

(अजादिषु प्रत्ययेषु एरनेकाचो…इति यणादेशः)

1.4.3.1. नीधात्वन्त – उन्नीः, ग्रामणीः (ङेरामादेशः)

1.4.3.2. सखा – (सखायमिच्छति) – (अनङ्‌णित्त्वे, ङसिङसोरुत्त्वम्‌)

1.4.3.3. सखीः, सुखीः, सुतीः, लूनीः, क्षामीः, प्रस्तीमीः (ख्यत्यात्परस्य इत्यनेन ङसिङसोरुत्त्वम्‌)

1.4.4. नदीसञ्ज्ञकः धातुः (असंयोगपूर्वकः इवर्णः, अनेकाच्‍ अङ्गम्‌)

(सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, अजादिषु प्रत्ययेषु एरनेकाचो…इति यणादेशः)

1.4.4.1. ङ्‍यन्त – कुमारी (ङ्‍यन्तत्वात्‌ सुलोपः)

1.4.4.2. अङ्‍यन्त – प्रकृष्टा धीर्यस्य सः प्रधीः – (अङ्‍यन्तत्वान्न सुलोपः)

1.4.5. अनदीसञ्ज्ञकः धातुः (संयोगपूर्वकः इवर्णः / एकाच्‌ अङ्गम्‌ / गतिकारकेतरपूर्वकः) या सुधीशब्दः

(अजादिषु प्रत्ययेषु ‘अचिश्नुधातुभ्रुवां.. इतीयङादेशः)

1.4.5.1. नीः (ङेरामादेशः)

1.4.5.2. अङ्‍यन्त - यवक्रीः, सुश्रीः (अङ्‍यन्तत्वान्न सुलोपः), सुधी:, शुष्कीः, पक्वीः इत्यादि

1.4.6. नदीसञ्ज्ञकः धातुः (संयोगपूर्वकः इवर्णः / एकाच्‌ अङ्गम्‌ / गतिकारकेतरपूर्वकः ) – शुद्धधीः, परमधीः, दुर्धीः, वृश्चिकभीः

(सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, अजादिषु प्रत्ययेषु इयङादेशः)

1.5. The word ends with U (ऊ).

1.5.1. अनदीसञ्ज्ञकः अधातुः – हूहू (इको यणचि, अमि पूर्वरूपम्‌, शसि पूर्वसवर्णदीर्घः)

1.5.2. नदीसञ्ज्ञकः अधातुः – अतिचमूः (सम्बुद्धिह्रस्वः, ङित्सु आडागमः, आमि नुडागमः, ङेरामादेशः, इको यणचि, अमि पूर्वरूपम्, शसि पूर्वसवर्णदीर्घः, ङौ सवर्णदीर्घः)

1.5.3. धातुः (असंयोगपूर्वकः उवर्णः, अनेकाच्‌ अङ्गम्‌ , गतिकारकपूर्वकः) – खलपूः, सुलूः, उल्लूः(अजादिषु प्रत्ययेषु ओः सुपि इति यणादेशः)

1.5.4. धातुः (संयोगपूर्वकः उवर्णः / एकाच्‌ अङ्गम्‌ / गतिकारकेतरपूर्वकः) – कटप्रूः, लूः, परमलूः( अजादिषु प्रत्ययेषु ‘अचिश्नुधातुभ्रुवां.. इत्युवङादेशः)

1.6. The word ends with ‘catur’ (चतुर्‌)

1.6.1. $second=== “Am” and $so!== “catur”

1.6.1.1. उपसर्जनीभूत

1.6.1.2. प्राधान्य

1.7. The word is “idam” / “idakam”

1.7.1. अन्वादेश है

1.7.2. अन्वादेश नहीं

1.8. The word ends with ‘n’

The word ends with array(“paYcan”, “saptan”, “zwan”, “navan”, ‘daSan”); and is not equal to them.

1.8.1. समास में नान्त शब्द उपसर्जनीभूत है ?

1.8.2. उपसर्जनीभूत नहीं है ।

1.9. The word ends with ‘j’

1.9.1. क्विन्‌ प्रत्ययान्त है ?

1.9.1.1. ends with ‘yuj’

1.9.1.1.1. युजि

1.9.1.1.2. युजिर्‌

1.9.2. क्विन्‍ प्रत्ययान्त नहीं है ।

1.9.3. ends with array(“BrAj”)

1.9.3.1. भ्राज्‌ धातु फणादि है ?

1.9.3.2. नहीं

1.10. The word ends with $tyadadi = array("dvi","tyad","tad","etad","idam","adas","eka","idakam");

1.10.1. त्यदादि शब्द सञ्ज्ञा या उपसर्जनीभूत के रूप में प्रयुक्त हुए हैं ?

1.10.2. त्यदादि शब्द सञ्ज्ञा या उपसर्जनीभूत नहीं है ।

1.10.2.1. The word belongs to array(“etad”, “idam”);

1.10.2.1.1. अन्वादेश है ?

1.10.2.1.2. अन्वादेश नहीं है ।

1.11. The word ends with array(“smad”,’yuzmad”) and !in\_array(“asmad”, “yuzmad”)

1.11.1. अस्मद्‌ / युष्मद्‌ एकत्ववाची है

1.11.2. अस्मद्‌ / युष्मद्‌ द्वित्ववाची है

1.11.3. अस्मद्‌ / युष्मद्‌ बहुत्ववाची है

1.12. The word is in\_array(‘asmad’, ‘yuzmad’) &&in\_array($so,array(‘am’, ‘Ow’, ‘Sas’, ‘Ne’, ‘ByAm’, ‘Byas’, ‘Nas’, ‘os’, ‘Am’)

1.12.1. अस्मद्‌ / युष्मद्‌ का प्रयोग पद के बाद में नहीं हुआ है । (निषेध)

1.12.2. अस्मद्‌ / युष्मद्‌ का प्रयोग पाद के प्रारम्भ में हुआ है । (निषेध)

1.12.3. अस्मद्‌ / युष्मद्‌ का च, वा, हा, अह या एव से साक्षात्‌ योग है । (निषेध)

1.12.4. अस्मद्‌ / युष्मद्‌ का अचाक्षुष ज्ञानार्थ धातु से योग है । (निषेध)

1.12.5. उपरमें से कोई नहीं

1.12.5.1. अन्वादेश है

1.12.5.1.1. विद्यमानपूर्व प्रथमान्त शब्द के परे अस्मद्‌ / युष्मद्‌ हैं

1.12.5.1.1.1. आमन्त्रित से परे हैं

1.12.5.1.1.1.1. विशेष्य से परे समानाधिकरण विशेषण है

1.12.5.1.1.1.1.1. बहुवचन है (विभाषा)

1.12.5.1.1.1.1.2. बहुवचन नहीं है (नित्य)

1.12.5.1.1.1.2. विशेष्य से परे समानाधिकरण विशेषण नहीं है (निषेध)

1.12.5.1.1.2. आमन्त्रित से परे नहीं है (विभाषा)

1.12.5.1.2. अस्मद्‌/ युष्मद्‌ विद्यमानपूर्व प्रथमान्त शब्द के परे नहीं हैं (नित्य)

1.12.5.2. अन्वादेश नहीं है (विभाषा)

1.13. The word ends with array(“aYc”, “AYc”, “anc”, “Anc) as the first word.

1.13.1. क्विन्‌ प्रत्ययान्त है

1.13.1.1. पूजाके अर्थ में प्रयुक्त है ?

1.13.1.2. पूजा के अर्थ में प्रयुक्त नहीं है ।

1.13.2. क्विन्‍ प्रत्ययान्त नहीं है

1.14. The first word ends with array(“Bavat”).

1.14.1. भाते र्डवतु ?

1.14.2. भूधातोः शतृप्रत्ययः ?

1.15. The first word ends with array(“takz”, “rakz”)

1.15.1. ण्यन्त

1.15.2. अण्यन्त

1.16. The first word ends with ‘m’

1.16.1. धातु

1.16.2. अधातु

1.17. The word ends with ‘at’

1.17.1. अत्वन्त है

1.17.2. अत्वन्त नहीं है

1.17.2.1. अभ्यस्त है

1.17.2.2. अभ्यस्त नहीं है

1.18. The word ends with ‘zaz’ and not equal to ‘zaz’

1.18.1. प्राधान्य

1.18.2. उपसर्जनीभूत

### 2. स्त्रीलिङ्ग

2.1. The word ends with ‘A’ (आ)

2.1.1. शब्द आबन्त नहीं है – जैसे कि गोपा ?

2.1.2. शब्द आबन्त है – जैसे कि रमा

2.1.2.1. Ends with $sarvanamastri = array("sarvA","viSvA","uBA","uBayA","atarA","atamA","anyA","anyatarA","itarA","tvA","nemA","simA","pUrvA","parA","avarA","dakziRA","uttarA","aparA","aDarA","svA","antarA","ekA","dvA",);

2.1.2.1.1. सर्वादयः सञ्ज्ञा के तौर पे प्रयुक्त हुए हैं ?

2.1.2.1.2. सर्वादयः उपसर्जनीभूत हैं ?

2.1.2.1.3. सर्वादयः तृतीया तत्पुरुष समास में प्रयुक्त हुए हैं ?

2.1.2.1.4. सर्वादयः द्वन्द्व समास में प्रयुक्त हुए हैं ?

2.1.2.1.5. सर्वादयः बहुव्रीहि समास में प्रयुक्त हुए हैं ?

2.1.2.1.5.1. दिक्समास है

2.1.2.1.5.2. दिक्समास नहीं है

2.1.2.1.6. उपर से किसी में नहीं प्रयुक्त हुए हैं

2.1.2.1.6.1. The word ends with any of the following. $purva = array("pUrva","para","avara","dakziRa","uttara","apara","aDara");

2.1.2.1.6.1.1. सञ्ज्ञा है या व्यवस्था के अर्थ में नहीं है

2.1.2.1.6.1.2. सञ्ज्ञा नहीं है और व्यवस्था के अर्थ में प्रयुक्त हुए हैं ।

2.1.2.1.6.2. The word ends with $sva = array("sva",);

2.1.2.1.6.2.1. ज्ञाति या धन के अर्थ में प्रयुक्त हुआ है

2.1.2.1.6.2.2. ज्ञाति या धन के अर्थ में प्रयुक्त नहीं हुआ है

2.1.2.1.6.3. The word ends with $antara = array(“antara”);

2.1.2.1.6.3.1. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त हुआ है

2.1.2.1.6.3.1.1. उसके बाद का शब्द पुरि है ?

2.1.2.1.6.3.1.2. नहीं

2.1.2.1.6.3.2. बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त नहीं हुआ है

2.1.2.1.6.4. The word ends with $datara = array(“atara”, “atama”); and !== “anyatama”

2.1.2.1.6.4.1. डतर / डतम प्रत्यय से शब्द की सिद्धि हुई है

2.1.2.1.6.4.2. नहीं

2.1.2.1.6.5. The word ends with $sama = array(“sama”);

2.1.2.1.6.5.1. सम शब्द सर्व के पर्याय में प्रयुक्त हुआ है

2.1.2.1.6.5.2. सम शब्द तुल्य के पर्याय में प्रयुक्त हुआ है

2.1.2.1.6.6. / / / / / /

2.1.2.1.6.7. . व

2.2. The word ends with ‘i’ (इ)

2.2.1. Ends with ‘tri’ and is not ‘tri’

2.2.1.1. त्रि शब्द स्त्रीलिङ्ग में है

2.2.1.2. त्रि शब्द स्त्रीलिङ्ग में नहीं है

2.3. The word ends with ‘I’ (ई)

2.3.1. ङ्‍यन्त, नदीसञ्ज्ञक (गौरी)

2.3.2. अङ्‍यन्त, नदीसञ्ज्ञक (लक्ष्मी)

2.3.3. ङ्‍यन्त, इयङ्स्थानिक (स्त्री)

2.3.4. अङ्‍यन्त, इयङ्स्थानिक (श्री)

2.3.5. प्रधीशब्द

2.3.5.1. प्रध्यायतीति प्रधीः

2.3.5.1.1. वृत्तिकारादीनां मते लक्ष्मीवद्रूपम्‌

2.3.5.1.2. कैयटमते तु पुंवद्रूपम्‌

2.3.5.2. प्रकृष्टा धीः (लक्ष्मीवद्‌, अमि शसि च प्रध्यम्‌, प्रध्यः इति विशेषः)

2.3.6. सुधी

2.3.6.1. सुष्ठु धीः यस्याः सा, सुष्ठु ध्यायतीति

2.3.6.1.1. वृत्तिकारमते श्रीवत्‌

2.3.6.1.2. मतान्तरे पुंवत्‌

2.3.6.2. सुष्ठुधीः (श्रीवत्‌)

2.3.7. ग्रामणीः (पुंवत्‌)

2.4. Ends with ‘U’ (ऊ)

2.4.1. नदीसञ्ज्ञक (वधूः) – गौरीवत्‌

2.4.2. इयङ्स्थानिक (भ्रूः) – श्रीवत्‌

2.4.3. ननदीसञ्ज्ञक (खलपूः) – पुंवत्‌

2.4.4. पुनर्भूः, दृन्भूः, पुनर्भूः – दृन्करपुनः... से यण्‌ तथा नदीकार्य

2.4.4.1. $so===”Am”

2.4.4.1.1. समास है

2.4.4.1.1.1. समास का उत्तरपद एकाच्‌ है और पूर्वपद में रकार या षकार है

2.4.4.1.1.2. समास का उत्तरपद एकाच्‌ नहीं है या पूर्वपद में रकार या षकार नहीं है

2.4.4.1.2. समास नहीं

2.4.5. वर्षाभूः

2.4.6. स्वयम्भूः - नित्यस्त्रीलिङ्गत्वाभावात्‌ न नदीसञ्ज्ञा । अङ्‍यन्तत्वान्न सुलोपः ।

2.5. The word is idam

2.5.1. अन्वादेश है

2.5.2. अन्वादेश नहीं है

2.6. The word is diS

2.6.1. क्विन्‌ प्रत्ययान्त है

2.6.2. क्विप्‌ प्रत्ययान्त है

### 3. नपुंसकलिङ्ग

3.1. ends with $ik=array(“i”, “I”, “u”, “U”, “f”, “F”, “x”, “X”);

3.1.1. भाषितपुंस्क है ?

3.1.2. भाषितपुंस्क नहीं है ।

3.2. The word is equal to “idam”,”etad”

3.2.1. अन्वादेश

3.2.2. अन्वादेश नहीं है

3.3. The word ends with “at”,”At”

3.3.1. अभ्यास के परे शतृप्रत्ययान्त शब्द है

3.3.2. नहीं

3.3.2.1. अवर्णान्त अङ्ग के परे शतृ प्रत्यय का अवयव है

3.3.2.1.1. शप्‌ / श्यन्‌ प्रत्यय के अवर्ण के परे शतृ प्रत्यय का अवयव है

3.3.2.1.2. नहीं

3.3.2.2. नहीं

## 2. ajax.php

When a user enters a word which meets the criteria mentioned in ajax requirements, the script known as script.js gets activated. This script passes the data to ajax.php for further processing. ajax.php sends the processed output back to the browser (typically a radio button for selection of options). The user ticks the appropriate radio button and thereafter while submitting this information is sent to subanta.php for processing according to SK. So, in a nutshell, script.js and ajax.php are codes which ask for user feedback when specific words mentioned in ajax requirement.docx are entered by user.

**The actual code is as under.**

<?php

// including function.php for arrays and functions.

include "function.php";

// including slp-dev.php for SLP1->devanAgarI conversion for display on front end.

include "slp-dev.php";

// including dev-slp.php for devanAgarI->SLP1 conversion to process data entered in Devanagari.

include "dev-slp.php";

//post element

$word = $\_POST['first'];

$tran = $\_POST['tran'];

// IAST and devanagari handling

// Code for converting from IAST to SLP

$iast = array("a","ā","i","ī","u","ū","ṛ","ṝ","ḷ","ḹ","e","ai","o","au","ṃ","ḥ","kh","ch","ṭh","th","ph","gh","jh","ḍh","dh","bh","ṅ","ñ","ṇ","k","c","ṭ","t","p","g","j","ḍ","d","b","n","m","y","r","l","v","s","h","ś","ṣ",); // IAST letters

$slp = array("a","A","i","I","u","U","f","F","x","X","e","E", "o","O", "M","H","K", "C", "W", "T", "P","G", "J", "Q", "D","B", "N","Y","R","k","c","w","t","p","g","j","q","d","b","n","m","y","r","l","v","s","h","S","z",); // SLP1 letters

if (preg\_match('/[āĀīĪūŪṛṚṝṜḷḶḹḸṃṂḥḤṭṬḍḌṅṄñÑṇṆśŚṣṢV]/',$word)) // If there is any IAST specific characters, we convert IAST->SLP1.

{

$word = str\_replace($iast,$slp,$word);

}

if ($tran === "IAST") // If user selects IAST transliteration, we convert IAST->SLP1.

{

$word = str\_replace($iast,$slp,$word);

}

// Devanagari handling. This is innocuous. Therefore even running without the selection in dropdown menu.

$word = json\_encode($word);

$word = str\_replace("\u200d","",$word); // removing whitespace.

$word = str\_replace("\u200c","",$word); // removing whitespace.

$word = json\_decode($word);

$word = convert1($word); // convert devangari to SLP1.

// usable variables

$arrWord[] = $word; // creating an array $arrWord with first member as $word.

$pancan=array("paYcan","saptan","zwan","navan","daSan"); // array of paJcan words.

$last = substr($word, -1); // returns "last letter"

$last\_4 = substr($word, -4); // returns "last 4 letters"

$last\_5 = substr($word, -5); // returns "last 5 letters"

$html = ''; // output is $html. If nothing applies, default it ''.

/\* A note on the cond variables. \*/

// The values of cond variables is mentioned in ajax requirements.doc. e.g. value="1" name="cond1\_1\_1" would correspond to 1.1.1.1 of ajax requirements. similarly, value="1" name="cond1\_1\_1\_5" would correspond to 1.1.1.5.1.

// Please check ajax requirements.doc to know what this ajax output stands for.

// ajax requirements.doc is created for user feedback, where according to SK, it is not possible for machine to decide what is the intention of the user.

/\* main coding part \*/

if($last === 'a' && $\_POST['first']!=="anyatama")

{

$sarvanama1 = array("sarva","viSva","uBa","uBaya","atara","atama","anya","anyatara","itara","tvat","tva","nema","sima","tyad","tad","yad","etad","idam","adas","eka","dvi","yuzmad","asmad","Bavat","kim","idakam","etara","pUrva","para","avara","dakziRa","uttara","apara","aDara","sva","antara","sama"); // sarvanAma list.

/\* checking whether $word ends with a member of $sarvanAma1 array \*/

// for function ends, see function.php.

if(ends($arrWord,$sarvanama1,1)){

if($\_POST['step'] === '1'){

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1" > सर्वादयः सञ्ज्ञा के तौर पे प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1" > सर्वादयः उपसर्जनीभूत हैं ? ';

$html .= '<input type="radio" value="3" name="cond1\_1\_1" > सर्वादयः तृतीया तत्पुरुष समास में प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="4" name="cond1\_1\_1" > सर्वादयः द्वन्द्व समास में प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="5" name="cond1\_1\_1" > सर्वादयः बहुव्रीहि समास में प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="6" name="cond1\_1\_1" > उपर से किसी में नहीं प्रयुक्त हुए हैं ';

$html .= '</div>';

}

if($\_POST['step'] === '1\_1\_1\_5' && ends($arrWord,array("pUrva","paScima","uttara","dakziNa"),1) ){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1\_5" > दिक्समास है ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1\_5" > दिक्समास नहीं है ';

$html .= '</div>';

}

if($\_POST['step'] === '1\_1\_1\_6'){

$purva = array("pUrva","para","avara","dakziRa","uttara","apara","aDara");

$sva = array("sva");

$antara = array("antara");

$datara = array("atara", "atama");

$sama = array("sama");

$anyatama = array("anyatama");

if(ends($arrWord,$purva,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1\_6\_1" > सञ्ज्ञा है या व्यवस्था के अर्थ में नहीं है ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1\_6\_1" > सञ्ज्ञा नहीं है और व्यवस्था के अर्थ में प्रयुक्त हुए हैं । ';

$html .= '</div>';

}elseif(ends($arrWord,$sva,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1\_6\_2" > ज्ञाति या धन के अर्थ में प्रयुक्त हुआ है ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1\_6\_2" > ज्ञाति या धन के अर्थ में प्रयुक्त नहीं हुआ है ';

$html .= '</div>';

}elseif(ends($arrWord,$antara,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1\_6\_3" > बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त हुआ है ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1\_6\_3" > बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त नहीं हुआ है ';

$html .= '</div>';

}elseif(ends($arrWord,$datara,1) && !ends($arrWord,$anyatama,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1\_6\_4" > डतर / डतम प्रत्यय से शब्द की सिद्धि हुई है ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1\_6\_4" > नहीं ';

$html .= '</div>';

}elseif(ends($arrWord,$sama,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1\_6\_5" > सम शब्द सर्व के पर्याय में प्रयुक्त हुआ है ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1\_6\_5" > सम शब्द तुल्य के पर्याय में प्रयुक्त हुआ है ';

$html .= '</div>';

}

}

if($\_POST['step'] === '1\_1\_1\_6\_3\_1'){

$html .= '<div id="step33">';

$html .= '<input type="radio" value="1" name="cond1\_1\_1\_6\_3\_1" > उसके बाद का शब्द पुरि है ? ';

$html .= '<input type="radio" value="2" name="cond1\_1\_1\_6\_3\_1" > नहीं ';

$html .= '</div>';

}

}

}

elseif($last === 'A' && $\_POST['step'] !== '1\_2' && $\_POST['gender']==='m')

{

$html .= '<div class="step11">';

$html .= '<input type="radio" value="1" name="cond1\_2" > क्या यह आकारान्त धातु है ? ';

$html .= '<input type="radio" value="2" name="cond1\_2" > नहीं ';

$html .= '</div>';

}

elseif($last === 'i' )

{

if ($last\_4 === 'saKi')

{

$html .= '<div class="step11">';

$html .= '<input type="radio" value="1" name="cond1\_3\_1" >प्राधान्य – जैसे कि सुसखा ';

$html .= '<input type="radio" value="2" name="cond1\_3\_1" >उपसर्जनीभूत – जैसे कि परमसखा ';

$html .= '<input type="radio" value="3" name="cond1\_3\_1" >लाक्षणिक – जैसे कि अतिसखिः (सखीमतिक्रान्तः) ';

$html .= '</div>';

}

if (ends($arrWord,array("dvi"),1))

{

$html .= '<div class="step11">';

$html .= '<input type="radio" value="1" name="cond1\_3\_2" >सञ्ज्ञा है या उपसर्जनीभूत है ';

$html .= '<input type="radio" value="2" name="cond1\_3\_2" >सञ्ज्ञा नहीं है और उपसर्जनीभूत नहीं है ';

$html .= '</div>';

}

}

elseif($last === 'I' )

{

if($\_POST['step'] === '1')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_4" > अनदीसञ्ज्ञकः अधातुः ';

$html .= '<input type="radio" value="2" name="cond1\_4" > नदीसञ्ज्ञकः अधातुः ';

$html .= '<input type="radio" value="3" name="cond1\_4" > अनदीसञ्ज्ञकः धातुः (असंयोगपूर्वकः इवर्णः, अनेकाच् अङ्गम्) ';

$html .= '<input type="radio" value="4" name="cond1\_4" > नदीसञ्ज्ञकः धातुः (असंयोगपूर्वकः इवर्णः, अनेकाच् अङ्गम्) ';

$html .= '<input type="radio" value="5" name="cond1\_4" > अनदीसञ्ज्ञकः धातुः (संयोगपूर्वकः इवर्णः / एकाच् अङ्गम् / गतिकारकेतरपूर्वकः) या सुधीशब्दः ';

$html .= '<input type="radio" value="6" name="cond1\_4" > नदीसञ्ज्ञकः धातुः (संयोगपूर्वकः इवर्णः / एकाच् अङ्गम् / गतिकारकेतरपूर्वकः) ';

$html .= '</div>';

}

if($\_POST['step'] === '1\_4\_2')

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_4\_2" > ङ्‍यन्त ';

$html .= '<input type="radio" value="2" name="cond1\_4\_2" > अङ्‍यन्त ';

$html .= '</div>';

}

if($\_POST['step'] === '1\_4\_3')

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_4\_3" > नीधात्वन्त ';

$html .= '<input type="radio" value="2" name="cond1\_4\_3" > सखा – (सखायमिच्छति) ';

$html .= '<input type="radio" value="3" name="cond1\_4\_3" > सखीः, सुखीः, सुतीः, लूनीः, प्रस्तीमीः इत्यादि ';

$html .= '<input type="radio" value="4" name="cond1\_4\_3" > अन्य ';

$html .= '</div>';

}

if($\_POST['step'] === '1\_4\_4')

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_4\_4" > ङ्‍यन्त ';

$html .= '<input type="radio" value="2" name="cond1\_4\_4" > अङ्‍यन्त ';

$html .= '</div>';

}

if($\_POST['step'] === '1\_4\_6')

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_4\_6" > ङ्‍यन्त ';

$html .= '<input type="radio" value="2" name="cond1\_4\_6" > अङ्‍यन्त ';

$html .= '</div>';

}

}

elseif ($last==="U")

{

if($\_POST['step'] === '1')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_5" > अनदीसञ्ज्ञकः अधातुः ';

$html .= '<input type="radio" value="2" name="cond1\_5" > नदीसञ्ज्ञकः अधातुः ';

$html .= '<input type="radio" value="3" name="cond1\_5" > धातुः (असंयोगपूर्वकः ऊवर्णः, अनेकाच् अङ्गम्, गतिकारकपूर्वकः) ';

$html .= '<input type="radio" value="4" name="cond1\_5" > धातुः (संयोगपूर्वकः ऊवर्णः / एकाच् अङ्गम् / गतिकारकेतरपूर्वकः) ';

$html .= '</div>';

}

}

if ( ends($arrWord,array("catur"),0) && $\_POST['step']==='1')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_6\_1" > चतुर्‌ उपसर्जनीभूत है ? ';

$html .= '<input type="radio" value="2" name="cond1\_6\_1" > चतुर्‌ प्रधान है । ';

$html .= '</div>';

}

if ( $last==="m" && $\_POST['step']==='1' && !ends($arrWord,array("idam","idakam"),2))

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_16" > धातु ';

$html .= '<input type="radio" value="2" name="cond1\_16" > अधातु ';

$html .= '</div>';

}

if ( ends($arrWord,$pancan,0) && $\_POST['step']==='1')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_8" > उपसर्जनीभूत ';

$html .= '<input type="radio" value="2" name="cond1\_8" > प्रधान ';

$html .= '</div>';

}

if ( ends($arrWord,array("BrAj"),1) && $\_POST['step']==='1')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_9\_3" > भ्राज्‌ धातु फणादि है ';

$html .= '<input type="radio" value="2" name="cond1\_9\_3" > नहीं ';

$html .= '</div>';

}

elseif ( $last==="j" && $\_POST['step']==='1')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_9" > क्विन्‌ प्रत्ययान्त है ';

$html .= '<input type="radio" value="2" name="cond1\_9" > नहीं ';

$html .= '</div>';

}

if ( ends($arrWord,array("tyad","tad","yad","etad","idam","adas","eka","dvi","idakam",),1) && $\_POST['step']!=='1\_10\_2' )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_10" > त्यदादि सञ्ज्ञा या उपसर्जनीभूत के तौर पर प्रयुक्त हुए हैं ';

$html .= '<input type="radio" value="2" name="cond1\_10" > नहीं ';

$html .= '</div>';

}

if ( ends($arrWord,array("etad","idam","idakam",),1) && $\_POST['step']==='1\_10\_2' )

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_10\_2" > अन्वादेश ';

$html .= '<input type="radio" value="2" name="cond1\_10\_2" > नहीं ';

$html .= '</div>';

}

if ( ends($arrWord,array("asmad","Asmad","yuzmad"),0) && $\_POST['step']==='1' )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_11" > अस्मद् / युष्मद् एकत्ववाची है ';

$html .= '<input type="radio" value="2" name="cond1\_11" > अस्मद् / युष्मद् द्वित्ववाची है ';

$html .= '<input type="radio" value="3" name="cond1\_11" > अस्मद् / युष्मद् बहुत्ववाची है ';

$html .= '</div>';

}

if ( ends($arrWord,array("asmad","Asmad","yuzmad"),2) && $\_POST['step']==='1' )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_12" > अस्मद् / युष्मद् का प्रयोग पद के बाद में नहीं हुआ है । ';

$html .= '<input type="radio" value="2" name="cond1\_12" > अस्मद् / युष्मद् का प्रयोग पाद के प्रारम्भ में हुआ है । ';

$html .= '<input type="radio" value="3" name="cond1\_12" > अस्मद्/ युष्मद् का च, वा, हा, अह या एव से साक्षात् योग है । ';

$html .= '<input type="radio" value="4" name="cond1\_12" > अस्मद्/ युष्मद् का अचाक्षुष ज्ञानार्थ धातु से योग है । ';

$html .= '<input type="radio" value="5" name="cond1\_12" > उपर में से कोई नहीं ';

$html .= '</div>';

}

if ( $\_POST['step']==='1\_12\_5' )

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_12\_5" > अन्वादेश है ';

$html .= '<input type="radio" value="2" name="cond1\_12\_5" > अन्वादेश नहीं है ';

$html .= '</div>';

}

if ( $\_POST['step']==='1\_12\_5\_1' )

{

$html .= '<div id="step33">';

$html .= '<input type="radio" value="1" name="cond1\_12\_5\_1" > अस्मद् / युष्मद् विद्यमानपूर्व प्रथमान्त शब्द के परे हैं ';

$html .= '<input type="radio" value="2" name="cond1\_12\_5\_1" > अस्मद् / युष्मद्‌ विद्यमानपूर्व प्रथमान्त शब्द के परे नहीं हैं ';

$html .= '</div>';

}

if ( $\_POST['step']==='1\_12\_5\_1\_1' )

{

$html .= '<div id="step44">';

$html .= '<input type="radio" value="1" name="cond1\_12\_5\_1\_1" > आमन्त्रित से परे हैं ';

$html .= '<input type="radio" value="2" name="cond1\_12\_5\_1\_1" > आमन्त्रित से परे नहीं है ';

$html .= '</div>';

}

if ( $\_POST['step']==='1\_12\_5\_1\_1\_1' )

{

$html .= '<div id="step55">';

$html .= '<input type="radio" value="1" name="cond1\_12\_5\_1\_1\_1" > विशेष्य से परे समानाधिकरण विशेषण है ';

$html .= '<input type="radio" value="2" name="cond1\_12\_5\_1\_1\_1" > विशेष्य से परे समानाधिकरण विशेषण नहीं है ';

$html .= '</div>';

}

if ( $\_POST['step']==='1\_12\_5\_1\_1\_1\_1' )

{

$html .= '<div id="step66">';

$html .= '<input type="radio" value="1" name="cond1\_12\_5\_1\_1\_1\_1" > बहुवचन है ';

$html .= '<input type="radio" value="2" name="cond1\_12\_5\_1\_1\_1\_1" > बहुवचन नहीं है ';

$html .= '</div>';

}

if ( in\_array($\_POST['step'],array("1","3")) && ends($arrWord,array("aYc","AYc","anc","Anc"),1) )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_13" > क्विन्‌ प्रत्ययान्त है ';

$html .= '<input type="radio" value="2" name="cond1\_13" > क्विन्‌ प्रत्ययान्त नहीं है ';

$html .= '</div>';

}

if ( $\_POST['step']==='1\_13\_1' && ends($arrWord,array("aYc","AYc","anc","Anc"),1) )

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_13\_1" > पूजा के अर्थ में प्रयुक्त है ';

$html .= '<input type="radio" value="2" name="cond1\_13\_1" > पूजा के अर्थ में नहीं है ';

$html .= '</div>';

}

if ( $\_POST['step']==='1' && ends($arrWord,array("at"),0) && !ends($arrWord,array("Bavat"),1) )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_17" > अत्वन्त है ';

$html .= '<input type="radio" value="2" name="cond1\_17" > अत्वन्त नहीं है ';

$html .= '</div>';

}

if ( $\_POST['step']==='1\_17\_2' )

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond1\_17\_2" > अभ्यस्त है ';

$html .= '<input type="radio" value="2" name="cond1\_17\_2" > अभ्यस्त नहीं है ';

$html .= '</div>';

}

if ( $\_POST['step']==='1' && ends($arrWord,array("Bavat"),1) )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_14" > भातेर्डवतुः ';

$html .= '<input type="radio" value="2" name="cond1\_14" > भूधातोः शतृप्रत्ययः ';

$html .= '</div>';

}

if ( $\_POST['step']==='1' && ends($arrWord,array("zaz"),0) )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_18" > प्राधान्य ';

$html .= '<input type="radio" value="2" name="cond1\_18" > उपसर्जनीभूतः ';

$html .= '</div>';

}

if ( $\_POST['step']==='1' && ends($arrWord,array("takz","rakz"),1) )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond1\_15" > ण्यन्त ';

$html .= '<input type="radio" value="2" name="cond1\_15" > अण्यन्त ';

$html .= '</div>';

}

// strIliGga

if ($last==="A" && $\_POST['step'] === '2')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond2\_1" > शब्द आबन्त नहीं है ';

$html .= '<input type="radio" value="2" name="cond2\_1" > शब्द आबन्त है ';

$html .= '</div>';

}

$sarvanamastri = array("sarvA","viSvA","uBA","uBayA","atarA","atamA","anyA","anyatarA","itarA","tvA","nemA","simA","pUrvA","parA","avarA","dakziRA","uttarA","aparA","aDarA","svA","antarA","ekA","dvA",);

$diksamAsa = array('uttarapUrvA','dakziRapUrvA','uttarapaScimA','dakziRapaScimA');

if($\_POST['step'] === '2\_1\_2' && ends($arrWord,$sarvanamastri,1) )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1" > सर्वादयः सञ्ज्ञा के तौर पे प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1" > सर्वादयः उपसर्जनीभूत हैं ';

$html .= '<input type="radio" value="3" name="cond2\_1\_2\_1" > सर्वादयः तृतीया तत्पुरुष समास में प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="4" name="cond2\_1\_2\_1" > सर्वादयः द्वन्द्व समास में प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="5" name="cond2\_1\_2\_1" > सर्वादयः बहुव्रीहि समास में प्रयुक्त हुए हैं ? ';

$html .= '<input type="radio" value="6" name="cond2\_1\_2\_1" > उपर से किसी में नहीं प्रयुक्त हुए हैं ';

$html .= '</div>';

}

if($\_POST['step'] === '2\_1\_2\_1\_5'){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1\_5" > दिक्समास है ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1\_5" > दिक्समास नहीं है ';

$html .= '</div>';

}

if($\_POST['step'] === '2\_1\_2\_1\_6'){

$purva = array("pUrvA","parA","avarA","dakziRA","uttarA","aparA","aDarA");

$sva = array("svA");

$antara = array("antarA");

$datara = array("atarA", "atamA");

$sama = array("samA");

$anyatama = array("anyatamA");

if(ends($arrWord,$purva,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1\_6\_1" > सञ्ज्ञा है या व्यवस्था के अर्थ में नहीं है ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1\_6\_1" > सञ्ज्ञा नहीं है और व्यवस्था के अर्थ में प्रयुक्त हुए हैं । ';

$html .= '</div>';

}elseif(ends($arrWord,$sva,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1\_6\_2" > ज्ञाति या धन के अर्थ में प्रयुक्त हुआ है ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1\_6\_2" > ज्ञाति या धन के अर्थ में प्रयुक्त नहीं हुआ है ';

$html .= '</div>';

}elseif(ends($arrWord,$antara,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1\_6\_3" > बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त हुआ है ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1\_6\_3" > बहिर्योग या उपसंव्यान के अर्थ में प्रयुक्त नहीं हुआ है ';

$html .= '</div>';

}elseif(ends($arrWord,$datara,1) && !ends($arrWord,$anyatama,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1\_6\_4" > डतर / डतम प्रत्यय से शब्द की सिद्धि हुई है ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1\_6\_4" > नहीं ';

$html .= '</div>';

}elseif(ends($arrWord,$sama,1)){

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1\_6\_5" > सम शब्द सर्व के पर्याय में प्रयुक्त हुआ है ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1\_6\_5" > सम शब्द तुल्य के पर्याय में प्रयुक्त हुआ है ';

$html .= '</div>';

}

}

if($\_POST['step'] === '2\_1\_2\_1\_6\_3\_1'){

$html .= '<div id="step33">';

$html .= '<input type="radio" value="1" name="cond2\_1\_2\_1\_6\_3\_1" > उसके बाद का शब्द पुरि है ? ';

$html .= '<input type="radio" value="2" name="cond2\_1\_2\_1\_6\_3\_1" > नहीं ';

$html .= '</div>';

}

if ($last==="i" && ends($arrWord,array("tri"),1) && !ends($arrWord,array("stri"),1) )

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond2\_2\_1" > त्रिशब्द स्त्रीलिङ्ग में है । ';

$html .= '<input type="radio" value="2" name="cond2\_2\_1" > त्रिशब्द स्त्रीलिङ्ग में नहीं है । ';

$html .= '</div>';

}

if ($last==="I" && $\_POST['step']==='2')

{

$html .= '<div id="step11">';

if (ends($arrWord,array("praDI"),1))

{

$html .= '<input type="radio" value="1" name="cond2\_3\_5" > प्रध्यायतीति प्रधी ';

$html .= '<input type="radio" value="2" name="cond2\_3\_5" > प्रकृष्टा धीः ';

}

elseif (ends($arrWord,array("suDI"),1))

{

$html .= '<input type="radio" value="1" name="cond2\_3\_6" > सुष्ठु धीः यस्याः सा / सुष्ठु ध्यायतीति ';

$html .= '<input type="radio" value="2" name="cond2\_3\_6" > सुष्ठु धीः ';

}

elseif (ends($arrWord,array("grAmaRI"),1))

{

}

else

{

$html .= '<input type="radio" value="1" name="cond2\_3" > ङ्‍यन्त, नदीसञ्ज्ञक (गौरी)';

$html .= '<input type="radio" value="2" name="cond2\_3" > अङ्‍यन्त, नदीसञ्ज्ञक (लक्ष्मी) ';

$html .= '<input type="radio" value="3" name="cond2\_3" > ङ्‍यन्त, इयङ्स्थानिक (स्त्री) ';

$html .= '<input type="radio" value="4" name="cond2\_3" > अङ्‍यन्त, इयङ्स्थानिक (श्री) ';

}

$html .= '</div>';

}

if ($last==="U" && $\_POST['step']==='2')

{

$html .= '<div id="step11">';

if (ends($arrWord,array("punarBU","dfnBU","karaBU","kAraBU"),1) )

{

$html .= '<input type="radio" value="1" name="cond2\_4\_4\_1" > समास है । ';

$html .= '<input type="radio" value="2" name="cond2\_4\_4\_1" > समास नहीं है । ';

}

elseif (ends($arrWord,array("varzABU","svayamBU"),1) )

{

}

else

{

$html .= '<input type="radio" value="1" name="cond2\_4" > नदीसञ्ज्ञक (वधूः)';

$html .= '<input type="radio" value="2" name="cond2\_4" > इयङ्स्थानिक (भ्रूः)';

$html .= '<input type="radio" value="3" name="cond2\_4" > न नदीसञ्ज्ञक (खलपूः)';

}

$html .= '</div>';

}

/\* napuMsakaliGga \*/

if (in\_array($last,array("i","I","u","U","f","F","x","X")) && $\_POST['step']==='3')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond3" > भाषितपुंस्क है ';

$html .= '<input type="radio" value="2" name="cond3" > नहीं ';

$html .= '</div>';

}

if (ends($arrWord,array("at","At"),1) && $\_POST['step']==='3')

{

$html .= '<div id="step11">';

$html .= '<input type="radio" value="1" name="cond3\_3" > अभ्यास के परे शतृप्रत्ययान्त शब्द है ';

$html .= '<input type="radio" value="2" name="cond3\_3" > नहीं ';

$html .= '</div>';

}

if ( $\_POST['step']==='3\_3\_2')

{

$html .= '<div id="step22">';

$html .= '<input type="radio" value="1" name="cond3\_3\_2" > अवर्णान्त अङ्ग के परे शतृ प्रत्यय का अवयव है ';

$html .= '<input type="radio" value="2" name="cond3\_3\_2" > नहीं ';

$html .= '</div>';

}

if ( $\_POST['step']==='3\_3\_2\_1')

{

$html .= '<div id="step33">';

$html .= '<input type="radio" value="1" name="cond3\_3\_2\_1" > शप् / श्यन् प्रत्यय के अवर्ण के परे शतृ प्रत्यय का अवयव है ';

$html .= '<input type="radio" value="2" name="cond3\_3\_2\_1" > नहीं ';

$html .= '</div>';

}

//display the output

echo $html;

exit;

?>

## 3. function.php

This file stores all the functions used by ajax.php and subanta.php.

Functions are used for some routine works, where we don’t want to repeat the long code.

List of functions used are as under. Some of the functions are not used in code. They were developed while the code was in nascent stage. Once the code matured, their need was over. They have still been preserved for future reference and future needs.

Actual code for functions are as follows.

<?php

/\* This code is developed by Dr. Dhaval Patel (drdhaval2785@gmail.com) of www.sanskritworld.in and Ms. Sivakumari Katuri.

\* Layout assistance by Mr Marcis Gasuns.

\* Available under GNU licence.

\* Version 1.1 date 17/11/2013

\* The latest source code is available at https://github.com/drdhaval2785/sanskrit

\* For setup, copy and paste sandhi.html, sandhi.php, function.php, mystyle.css, slp-dev.php and dev-slp.php to your localhost and server and run sandhi.html.

\* sandhi.html is the frontend for the code.

\* function.php stores the frequently used functions in this code (The description on how to use the code is there in function.php.

\* slp-dev.php is for converting SLP1 data to Devanagari. dev-slp.php is for converting Devanagari data to SLP1.

\* Mystyle.css is stylesheet where you can change your preferences.

\*/

### /\* Defining grammatical arrays \*/

$shiv=array("a","i","u","-R","f","x","-k","e","o","-N","E","O","-c","h","y","v","r","-w","l","-R","Y","m","N","R","n","-m","J","B","-Y","G","Q","D","-z","j","b","g","q","d","-S","K","P","C","W","T","c","w","t","-v","k","p","-y","S","z","s","-r","h","-l");

$kantha = array("a","A","k","K","g","G","N","h","H"); // kanThya letters

$talu = array("i","I","c","C","j","J","Y","y","S"); // tAlavya letters

$murdha = array("f","F","w","W","q","Q","R","r","z"); // mUrdhanya letters

$danta = array("x","X","t","T","d","D","n","l","s"); // dantavya letters

$oshtha = array("u","U","p","P","b","B","m"); // oSThya letters

$nasika = array("N","Y","R","n","m","M"); // anunAsika letters

$kanthatalu = array ("e","E"); // kanThatAlu letters

$kanthoshtha = array("o","O"); // kanThauSThya letters

$dantoshtha = array("v"); // dantauSThya letters

$sparsha = array("k","K","g","G","N","c","C","j","J","Y","w","W","q","Q","R","t","T","d","D","n","p","P","b","B","m"); // sparza letters

$sprushta = array("k","K","g","G","N","c","C","j","J","Y","w","W","q","Q","R","t","T","d","D","n","p","P","b","B","m",); // spRSTa letters

$ishatsprushta = array("y","r","l","v"); // iSatspRSTa letters

$vivruta = array("S","z","s","h","a","A","i","I","u","U","f","F","x","X","e","E","o","O"); // vivRta letters

$samvruta = array("a"); // saMvRta letter

$vivara = array("k","K","c","C","w","W","t","T","p","P","H","S","z","s"); // vivAra letters

$shvasa = array("k","K","c","C","w","W","t","T","p","P","H","S","z","s"); // zvAsa letters

$aghosha = array("k","K","c","C","w","W","t","T","p","P","H","S","z","s"); // aghoSa letters

$samvara = array("g","G","N","j","J","Y","q","Q","R","d","D","n","b","B","m","y","r","l","v","h"); // saMvAra letters

$nada = array("g","G","N","j","J","Y","q","Q","R","d","D","n","b","B","m","y","r","l","v","h"); // nAda letters

$ghosha = array("g","G","N","j","J","Y","q","Q","R","d","D","n","b","B","m","y","r","l","v","h"); // ghoSa letters

$alpaprana = array("k","g","c","j","w","q","t","d","p","b","y","r","l","v"); // alpaprANa letters

$mahaprana = array("K","G","C","J","W","Q","T","D","P","B","S","z","s","h"); // mahAprANa letters

$ru = array("f","F","x","X"); // R, RR, lR and lRR

$ac = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O",); // vowels - 'ac' letters

$hl = array("k","K","g","G","N","c","C","j","J","Y","w","W","q","Q","R","t","T","d","D","n","p","P","b","B","m","y","r","l","v","S","z","s","h"); // consonants - hal letters

$e = array("e","E","o","O"); // 'ec' letters

$dirgha = array("A","I","U","F","X","e","E","o","O"); // dIrgha letters

$hrasva = array("a","i","u","f","x"); // hrasva letters

$guna = array ("e","o"); // guNa letters

$vruddhi = array("E","O"); // vRddhi letters

$aa = array("a","A"); // a, A

$upasarga = array("pra","prati","api","parA","apa","upa","pari","anu","ava","vi","saM","su","ati","ni","nir","ut","adhi","dur","abhi"); // upasargas

$verbs\_ru = array("fkz","fc","fC","fj","fYj","fR","ft","fd","fD","fn","fP","fBukz","fmP","fS","fz","fh"); // verbs starging with R.

$verbs\_changed = array("kz","c","C","j","Yj","R","t","d","D","n","P","Bukz","mP","S","z","h"); // verbs with removal of R.

$ku = array("k","K","g","G","N"); // kavarga letters

$cu = array("c","C","j","J","Y"); // cavarga letters

$Tu = array("w","W","q","Q","R"); // Tavarga letters

$tu = array("t","T","d","D","n"); // tavarga letters

$pu = array("p","P","b","B","m"); // pavarga letters

$antastha = array("y","r","l","v"); // antaHstha letters

$ushma = array("S","z","s","h"); // uSma letters

$iN = array("i","I","u","U"); // 'iN' pratyAhAra

$pratya = array("aR","ak","ik","uk","eN","ac","ic","ec","Ec","aw","aR","iR","yaR","am","yam","Yam","Nam","yaY","Jaz","Baz","aS","haS","vaS","JaS","jaS","vaS","Cav","yay","may","Jay","Kay","cay","yar","Jar","Kar","car","Sar","al","hal","val","ral","Jal","Sal"); // valid pratyAhAras according to pANinIya vyAkaraNa

$sup = array("su!","Ow","jas","am","O","Sas","wA","ByAm","Bis","Ne","ByAm","Byas","Nasi!","ByAm","Byas","Nas","os","Am","Ni","os","sup"); // sup pratyayas

$acsup = array("O","jas","am","Ow","Sas","wA","Ne","Nasi!","Nas","os","Am","Ni","os"); // sup pratyayas which start with a vowel

$hlsup = array("su!","ByAm","Bis","Byas","sup"); // sup pratyayas which start with a consonant

$prathama = array("su!","O","jas","am","Ow","Sas"); // sup pratyayas of first two vibhaktis

$sarvanama = array("sarva","viSva","uBa","uBaya","atara","atama","anya","anyatara","itara","tvat","tva","nema","sima","tyad","tad","yad","etad","idam","adas","eka","dvi","yuzmad","asmad","Bavat","kim","idakam","etara"); // sarvanAmas

$zasadi = array("Sas","wA","ByAm","Bis","Ne","ByAm","Byas","Nasi!","ByAm","Byas","Nas","os","Am","Ni","os","sup"); // sup pratyayas, starting from zas

$sarvanamasthana = array("su!","O","jas","am","Ow","Si"); // sarvanAmasthAna pratyayas

$yacibham = array("Sas","wA","Ne","Nasi!","Nas","os","Am","os","Ni"); // sup pratyayas which mandate bha saJjJA subject to some conditions

$tRtIyAdiSvaci = array("wA","Ne","Nasi!","Nas","os","Am","Ni","os"); // sup pratyayas tRtIyA vibhakti onwards and starting with vowel

$tRtIyAdi = array("wA","ByAm","Bis","Ne","ByAm","Byas","Nasi!","ByAm","Byas","Nas","os","Am","Ni","os","sup"); // sup pratyayas tRtIyA vibhakti onwards

$eksup = array("su!","am","wA","Ne","Nasi!","Nas","Ni",); // ekavacana sup pratyayas

$dvisup = array("O","Ow","ByAm","ByAm","ByAm","os","os",); // dvivacana sup pratyayas

$bahusup = array("jas","Sas","Bis","Byas","Byas","Am","sup"); // bahuvacana sup pratyayas

$diksamAsa = array("uttarapUrvA","dakziRapUrvA","uttarapaScimA","dakziRapaScimA"); // diksamAsa array

$sarvanamastri = array("sarvA","viSvA","uBA","uBayA","qatarA","qatamA","anyA","anyatarA","itarA","tvA","nemA","simA","pUrvA","parA","avarA","dakziRA","uttarA","aparA","aDarA","svA","antarA","ekA","dvA","kA","idA"); // sarvanAmas feminine gender

$svasrAdi = array("svasf","tisf","catasf","nanAndf","duhitf","yAtf","mAtf"); // svasrAdi gaNa

$tyadadi = array("dvi","tyad","tad","yad","etad","idam","adas","eka","idakam"); // tyadAdi gaNa

$acdir = array("A","A","I","I","U","U","F","F","F","F","e","o","E","O",); // dIrgha of vowels

$tiG = array("tip","tas","Ji"); // tiG pratyayas. Not used right now.

### /\* Function to find pratyAhAra from given two letters \*/

// Enter your letters in the arguments like prat('Jl') will mean pratyAhAra jhal.

function prat($text) // prat for pratyAhAra

{

global $shiv; // incorporating $shiv from arrays.

if ($text === "ra") // for circumventing the imaginary ra pratyAhAra

{

$text = array("r","l");

}

elseif ($text === "yR") // for circumventing the yaN pratyAhAra because of problem of double N

{

$text = array("y","y","v","v","r","r","l","l");

}

else

{

$a = str\_split($text); // separating starting letter and ending letter.

$a[1] = str\_replace($a[1],"-".$a[1],$a[1]); // adding a dash.

$text = array\_slice($shiv,array\_search($a[0],$shiv),array\_search($a[1],$shiv)-array\_search($a[0],$shiv)+1); // taking the necessary pratyAhAra from $shiv.

$b = implode(" ",$text); // creating a single string of pratyAhAra.

$b = str\_replace(" -R","",$b); // removing the end markers of pratyAhAras in between.

$b = str\_replace(" -k","",$b);

$b = str\_replace(" -N","",$b);

$b = str\_replace(" -c","",$b);

$b = str\_replace(" -w","",$b);

$b = str\_replace(" -m","",$b);

$b = str\_replace(" -Y","",$b);

$b = str\_replace(" -z","",$b);

$b = str\_replace(" -S","",$b);

$b = str\_replace(" -v","",$b);

$b = str\_replace(" -y","",$b);

$b = str\_replace(" -r","",$b);

$b = str\_replace(" -l","",$b);

$b = str\_replace("a","a A",$b); // Adding the dIrgha of vowels.

$b = str\_replace("i","i I",$b);

$b = str\_replace("u","u U",$b);

$b = str\_replace("f","f F",$b);

$b = str\_replace("x","x X",$b);

$b = str\_replace("y","y y",$b);

$b = str\_replace("v","v V",$b);

$b = str\_replace("r","r r",$b);

$b = str\_replace("l","l l",$b);

$text = explode(" ",$b); // creating the array of pratyAhAra back.

}

return $text; // returning the result of function to user.

}

### /\* function pratyAhAra check is for checking in regular expressions.

\* In regular expressions we don't treat arrays. usually it is 'aAiIuU'.

\* so this function converts the pratyAhAra to this flat format. \*/

function pc($text) // pratyAhAra check

{

$text = "'".implode("",prat($text))."'"; // e.g. converting array("a","A") to 'aA'.

return $text; // returning the result to the user.

}

### /\* Function one is for replacing one letter in the whole array of $text with another letter \*/

// There are three arguments.

// $a is the array which you want to change.

// $b is the array which will be put in place of the replaced one.

// $merge can take two values. 0 will mean that the whole $text will be replaced with the new replaced values. Used in case of mandatory Adezas.

// 1 will mean that $text will not be replaced, but the replaced values will be added to it. Used in case of optional Adezas.

function one($a,$b,$merge)

{

global $text; // taking $text from the subanta.php or sandhi.php and using it here.

for($z=0;$z<count($text);$z++)

{

$p = $text[$z]; // $p is a word from the array $text.

for($i=0;$i<count($a);$i++)

{

$p = str\_replace($a[$i],$b[$i],$p); // Doing replace in $p.

}

$text1[$z] = $p; // storing $p in array $text1.

}

if ($merge === 0) // if the Adeza is mandatory.

{

$text = $text1; // replaced $text with $text1 (replace one)

$text = array\_unique($text); // keeping only unique values.

$text = array\_values($text); // giving new serial number to the values.

}

if ($merge === 1) // if the Adeza is optional.

{

$text = array\_merge($text,$text1); // merging $text, and the replaced $text1.

$text = array\_unique($text);

$text = array\_values($text);

}

return $text; // giving the result to the user.

}

### /\* Function two is for replacing one letter in the whole array of $text with another letter \*/

// There are five arguments, $a,$b are the arrays which you want to change, $c,$d are the arrays which will be put in place of the replaced one.

// $merge can take five values.

// 0 will mean that the whole $text will be replaced with the new replaced values. Used in case of mandatory Adezas.

// 1 will mean that $text will not be replaced, but the replaced values will be added to it. Used in case of optional Adezas.

// 2 = without + sign, mandatory (apadAnta etc)

// 3 = without + sign, optional

// 4 = with + sign only, mandatory (padAnta etc)

// 5 = with + sign only, optional.

function two($a,$b,$c,$d,$merge) // the comments are the same as one function.

{

global $text;

for ($z=0;$z<count($text);$z++)

{$p = $text[$z];

for($i=0;$i<count($a);$i++)

{

for($j=0;$j<count($b);$j++)

{

if($merge<2)

{

$p = str\_replace($a[$i].$b[$j],$c[$i].$d[$j],$p); // replacement without "+" sign.

$p = str\_replace($a[$i]."+".$b[$j],$c[$i]."+".$d[$j],$p); // replacement for "+" sign.

}

elseif ($merge<4)

{

$p = str\_replace($a[$i].$b[$j],$c[$i].$d[$j],$p); // replacement without "+" sign.

}

else

{

$p = str\_replace($a[$i]."+".$b[$j],$c[$i]."+".$d[$j],$p); // replacement with "+" sign.

}

}

}

$text1[$z] = $p;

}

if (($merge === 0) || ($merge === 2) ||($merge === 4) ) // mandatory Adezas

{

$text2 = $text1;

$text2 = array\_unique($text2);

$text2 = array\_values($text2);

}

if (($merge === 1) || ($merge === 3) ||($merge === 5) ) // optional Adezas.

{

$text2 = array\_merge($text,$text1);

$text2 = array\_unique($text2);

$text2 = array\_values($text2);

}

return $text2;

}

### /\* Function three is for replacing one letter in the whole array of $text with another letter \*/

// There are seven arguments, $a,$b,$c are the arrays which you want to change, $d,$e,$f are the arrays which will be put in place of the replaced one.

// $merge can take two values.

// 0 will mean that the whole $text will be replaced with the new replaced values. Used in case of mandatory Adezas.

// 1 will mean that $text will not be replaced, but the replaced values will be added to it. Used in case of optional Adezas.

function three($a,$b,$c,$d,$e,$f,$merge) // comments are the same as one function.

{global $text;

for ($z=0;$z<count($text);$z++)

{$p = $text[$z];

for($i=0;$i<count($a);$i++)

{

for($j=0;$j<count($b);$j++)

{

for($k=0;$k<count($c);$k++)

{

$p = str\_replace($a[$i].$b[$j].$c[$k],$d[$i].$e[$j].$f[$k],$p);

$p = str\_replace($a[$i]."+".$b[$j].$c[$k],$d[$i]."+".$e[$j].$f[$k],$p);

$p = str\_replace($a[$i].$b[$j]."+".$c[$k],$d[$i].$e[$j]."+".$f[$k],$p);

$p = str\_replace($a[$i]."+".$b[$j]."+".$c[$k],$d[$i]."+".$e[$j]."+".$f[$k],$p);

}

}

}

$text1[$z] = $p;

}

if ($merge === 0)

{

$text = $text1;

$text = array\_unique($text);

$text = array\_values($text);

}

if ($merge === 1)

{

$text = array\_merge($text,$text1);

$text = array\_unique($text);

$text = array\_values($text);

}

return $text;

}

### /\* function flatten is a function to flatten a multidimentional array \*/

// I have borrowed this function from examples in php.net site.

// e.g. array([0]=> a [1]=> array([0] => b [1] => c)) this is flattened in array([0]=>a [1]=>b [2]=>c).

/\*function flatten(array $array) // $array is the multidimentional array which you want to flatten.

{

$return = array();

array\_walk\_recursive($array, function($a) use (&$return) { $return[] = $a; });

return $return;

}\*/ // This code is immeculate. But unfortunately the servers don't support PHP5.4.

function flatten($array) {

if (!is\_array($array)) {

// nothing to do if it's not an array

return array($array);

}

$result = array();

foreach ($array as $value) {

// explode the sub-array, and add the parts

$result = array\_merge($result, flatten($value));

}

return $result;

}

### /\* function savarna is to find out savarNa of a character from the given pratyAhAra \*/

// $inarray is the array for which you are looking for savarNa.

// $array is the array in which you are looking for savarNa.

// e.g. savarNa(array("a"),prat('ac')) will give savarNa of letter 'a' from the pratyAhAra 'ac'.

// Known issue - words having two Asyas are not treated properly.

function savarna($inarray,$array)

{

for($z=0;$z<count($inarray);$z++)

{

$text = $inarray[$z];

global $kantha,$talu,$murdha,$danta,$oshtha,$nasika,$kanthatalu,$kanthoshtha,$dantoshtha,$sprushta,$ishatsprushta,$vivruta,$samvruta,$aghosha,$alpaprana,$ghosha,$mahaprana,$ac,$udatta,$anudatta,$svarita,$shvasa,$nada,$vivara,$samvara,$hl,$ru,$e; // bringing the arrays in function

// defining an array for sthAna

$i=0;

if (in\_array($text,$kantha)) { $sthana[$i] = $kantha; $i++; }

if (in\_array($text,$talu)) { $sthana[$i] = $talu; $i++; }

if (in\_array($text,$murdha)) { $sthana[$i] = $murdha; $i++; }

if (in\_array($text,$danta)) { $sthana[$i] = $danta; $i++; }

if (in\_array($text,$oshtha)) { $sthana[$i] = $oshtha; $i++; }

if (in\_array($text,$nasika)) { $sthana[$i] = $nasika; $i++; }

if (in\_array($text,$kanthatalu)) { $sthana[$i] = $kanthatalu; $i++; }

if (in\_array($text,$kanthoshtha)) { $sthana[$i] = $kanthoshtha; $i++;}

if (in\_array($text,$dantoshtha)) { $sthana[$i] = $dantoshtha; $i++; }

// defining an array for Abhyantara prayatna for consonants

$j=0;

if (in\_array($text,$sprushta)) { $abhyantara[$j] = $sprushta; $j++; }

if (in\_array($text,$ishatsprushta)) { $abhyantara[$j] = $ishatsprushta; $j++; }

if (in\_array($text,$vivruta)) { $abhyantara[$j] = $vivruta; $j++; }

if (in\_array($text,$samvruta)) { $abhyantara[$j] = $samvruta; $j++; }

// defining an array for bAhya prayatna for consonants and prANa

$k=0;

if (in\_array($text,$aghosha)) { $ghosh[$k] = $aghosha; $k++; }

if (in\_array($text,$alpaprana)) { $prana[$k] = $alpaprana; $k++; }

if (in\_array($text,$ghosha)) { $ghosh[$k] = $ghosha; $k++; }

if (in\_array($text,$mahaprana)) { $prana[$k] = $mahaprana; $k++; }

// defining an array for bAhya prayatna of vowels

$u=0;

if (in\_array($text,$ac)) { $svar[$u] = $udatta; $u++; }

// Finding out intersections of sthAna, Abhyantara prayatna and bAhya prayatnas of the given letter and the given array.

if(empty($sthana)===FALSE)

{

$sthanasamya = array\_intersect(flatten($sthana),$array); // finding the intersection of sthAna

}

if(empty($abhyantara)===false)

{

$abhyantarasamya = array\_intersect(flatten($abhyantara),$array); // finding the intersection of Abhyantara prayatna

}

if(empty($ghosh)===FALSE)

{

$ghoshasamya = array\_intersect(flatten($ghosh),$array); // finding the intersection of bAhya prayatna

}

if(empty($prana)===FALSE)

{

$pranasamya = array\_intersect(flatten($prana),$array); // finding the intersection of prANa.

}

if(empty($svar)===false)

{

if(in\_array($text,$ac)) { $svarasamya = array\_intersect(flatten($svar),$array,$ac); // finding the svarasAmya

}

}

$l = array\_intersect($sthanasamya,$abhyantarasamya,$ghoshasamya,$pranasamya); // $l is the case in which sthAna, Abhyantaraprayatna, ghoSa and prANa all are matching. The best match.

$m = array\_intersect($sthanasamya,$abhyantarasamya,$ghoshasamya); // $m is the case in which sthAna, Abhyantaraprayatna and ghoSa are matching. The second best match.

$n = array\_intersect($sthanasamya,$abhyantarasamya); // $n is the case in which sthAna and Abhyantaraprayatna are matching. The third best match.

$o = array\_intersect($sthanasamya,$abhyantarasamya,$pranasamya); // $o is the case in which sthAna, Abhyantaraprayatna and prANa are matching. The second best match.

$p = array\_intersect($sthanasamya,$ghoshasamya); // $p is the case in which sthAna and ghoSa are matching. The third best match.

$q = array\_intersect($sthanasamya,$pranasamya); // $q is the case in which sthAna and prANa are matching. The third best match.

// Defining savarNas for consonants

if(in\_array($text,$hl))

{

if(empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($ghoshasamya)===false&&empty($pranasamya)===FALSE&&empty($l)===false) // four match

{

$savarna = implode(", ",$l);

}

elseif (empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($ghoshasamya)===false&&empty($m)===false) // three match

{

$savarna = implode(", ",$m);

}

elseif (empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($pranasamya)===false&&empty($o)===false) // three match

{

$savarna = implode(", ",$o);

}

elseif (empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($n)===false) // two match

{

$savarna = implode(", ",$n);

}

elseif (empty($sthanasamya)===false&&empty($ghoshasamya)===false&&empty($p)===false) // two match

{

$savarna = implode(", ",$p);

} elseif (empty($sthanasamya)===false&&empty($pranasamya)===false&&empty($q)===false) // two match

{

$savarna = implode(", ",$q);

}

else // no match

{

$savarna = implode(", ",$sthanasamya);

}

}

// defining savarNas for vowels

else

{

if (in\_array($text,$ru)||in\_array($text,$e))

{// patch for $ru

for($i=0;$i<4;$i++)

{

if ($text === $ru[$i])

{

if (in\_array($text,$array))

{ $savarna = "f, F, x, X"; }

else

{ $savarna = ""; }

}

// patch for non sAvarNya of e,E,o,O

elseif ($text === $e[$i])

{

if (in\_array($text,$array))

{ $savarna = $text; }

else

{ $savarna = ""; }

}

}

}

else

{// In case of other vowels.

$savarna = implode(", ",$sthanasamya);

}

}

// stores that savarNa letter in memory.

$arr[$z] = $savarna; // We got the savarNa for the first letter of the array. The loop is repeated for all members of the array.

}

return $arr; // returning the desired savarNa array.

}

### /\* function display will show the text on screen. \*/

// There are three arguments. 0 will simply display the message.

// 1 will show an additional message in dvitva.

// 2 is not used in the code. Useful where there were two optional forms.

// 3 is for aGgAdhikAra.

// 4 is for yaNaH pratiSedho vAcyaH in case of yaN.

// 5 is for sarvAdeza.

// 6 is for bhasya adhikAra.

// 7 is for padasya, padAt, anudAttaM sarvamapAdAdau adhikAra.

// new messages can be added if the message is of repeated nature.

function display($n)

{

global $text; // bringing $text from subanta.php

if ($n === 1) // sending special messages.

{

echo "<p class = hn>Please note: Wherever there is dvitva, it is optionally negated by sarvatra zAkalyasya. (8.4.51)</p>";

echo "<p class = hn>द्वित्व का सर्वत्र सर्वत्र शाकल्यस्य (८.४.५१) से पाक्षिक निषेध होता है ।</p>";

}

if ($n === 2)

{

global $text1; $text2 = $text; $text = $text1;

}

if ($n === 3)

{

echo "<p class = pa>yasmAtpratyayavidhistadAdi pratyaye'Ggam (2.4.13) and aGgasya (6.4.1) </p>";

echo "<p class = pa>यस्मात्प्रत्ययविधिस्तदादि प्रत्ययेऽङ्गम्‌ (२.४.१३) तथा अङ्गस्य (६.४.१) </p>";

}

if ($n === 4)

{

echo "<p class = hn>N.B.: yaNaH pratiSedho vAcyaH (vA 4806) prevents application of saMyogAntasya lopaH (8.2.23) </p>";

echo "<p class = hn>यणः प्रतिषेधो वाच्यः (वा ४८०६) से संयोगान्तस्य लोपः (८.२.२३) का निषेध होता है ।</p>";

}

if ($n === 5)

{

echo "<p class = hn>N.B.: anekAlzitsarvasya (1.1.55) mandates sarvAdeza. </p>";

echo "<p class = hn>अनेकाल्शित्सर्वस्य (१.१.५५) से सर्वादेश होता है ।</p>";

}

if ($n === 6)

{

echo "<p class = pa>yasmAtpratyayavidhistadAdi pratyaye'Ggam (2.4.13) aGgasya (6.4.1) and bhasya (6.4.129) :</p>";

echo "<p class = pa>यस्मात्प्रत्ययविधिस्तदादि प्रत्ययेऽङ्गम्‌ (२.४.१३), अङ्गस्य (६.४.१) तथा भस्य (६.४.१२९) :</p>";

}

if ($n === 7)

{

echo "<p class = pa>padasya (8.1.16), padAt (8.1.17) and anudAttaM sarvamapAdAdau (8.1.18) :</p>";

echo "<p class = pa>पदस्य (८.१.१६), पदात्‌ (८.१.१७) तथा अनुदात्तं सर्वमपादादौ (८.१.१८) :</p>";

}

for($i=1;$i<count($text)+1;$i++) // for all members of the $text array

{

echo "<p class = form>$i - ".convert($text[$i-1])."</p>"; // showing the output to the browser. e.g. 1. rAmaH. $i is for numbering. function convert converts the output into devanAgarI.

}

echo "<hr>"; // drawing a line after all the words are displayed. Thus two sUtras are separated by a separator line.

if ($n === 2) { $text1 = $text; $text = $text2; } // not useful because the display(2) is not used in the code. If it is used, this will work.

}

### /\* function dvitva will be used to duplicate a letter \*/

// It has six arguments, first four arguments are the arrays which need to be replaced.

// $location is to specify which of these 4 consecutive arrays is to be duplicated. e.g. 2 will mean that the second member will be duplicated.

// $merge is to specify whether to replace (0) or add (1) to the existing array. 0 is used for mandatory dvitva and 1 is used for optional dvitva.

// I am not able to elucidate this code much, because it is a borrowed code. I have kept it as the lender gave.

function dvitva ($kantha,$talu,$murdha,$oshtha,$location,$merge)

{

global $text;

// this is an array of the input values to use below

// get all possible combianations of $kantha+$talu+$murdha

$combinations = array();

foreach($kantha as $k) // "a","k","g","n","h"

{

foreach($talu as $t) // "i","c","j","y","s"

{

foreach($murdha as $m) // "f","w","q","r","z"

{

foreach ($oshtha as $o)

{

$combinations[] = $k.$t.$m.$o;

}

}

}

}

$values1 = array();

foreach ($text as $stti => $string) // 'aifkcwh', 'aifkcwhsz', 'kim', 'aif'

{

$values = array($string);

reset($values);

// loop through $values using an array pointer

// while the current array pointer position is not null/false

while(current($values)!==false)

{

// on the first iteration, $values will be the inputted string

// from $strings\_to\_test... our example is "aifkcwh"

$value = current($values);

if ($merge === 1)

{

$values1[] = current($values);

}

// for each possible combination of $kantha.$talu.$murdha

// let's say our first combination is "aif"

foreach($combinations as $ci => $combination)

{

// look and see if the current value we are looking

// at ("aifkcwh") contains this combination string

if (strpos($value,$combination)!==false)

{

// if it does... we perform the string mutation

// "aif" does exist in the string "aifkcwh"

// get the second letter in the combination string

// echo $combination. "These are combinations </br>";

$single = $combination[$location-1]; // i

// double that second letter

$post = substr($combination,$location-1);

// echo $post. "These are posts </br>";

$pre = chop($combination,$post);

// echo $pre. "These are pres </br>";

// create a new string that is the combination with the

// second letter doubled...

$newcom = $pre.$combination[$location-1].$post;

// echo $newcom."These are newcoms </br> ";

// replace that string in $values so that

// aifkcwh becomes aiifkcwh

$newval = str\_replace($combination, $newcom, $value); // aiifkcwh

// does the new value ("aiifkcwh") exist in $values?

// have we already recorded this mutation?

if (!in\_array($newval,$values1))

{

// if not... append it to the array $values

$values1[] = $newval;

// now, values would go from being = array([0]=>'aifkcwh');

// to array ([ 0] => 'aifkcwh', [1] => 'aiifkcwh' );

}

else

{

}

}

} // <-- end of the foreach statement, this will go through all combinations

// in our combinations array for this particular value which is currently aifkcwh

// next($values) increments the array pointer so that we move to the next

// value in the $values array. since we just added a value,

// $values now contains array ([ 0] => 'aifkcwh', [1] => 'aiifkcwh' );

// before this statement index 0, current($values) == 'aifkcwh'

next($values);

// after this statement index 1, current($values) == 'aiifkcwh'

// for the next loop, we will test this string for all the combinations

// if there is no next value, the `while` loop will end

}

// after we have gone through every possible combination for "aifkcwh",

// we will have something like this:

/\*

Array

(

[0] => aifkcwh

[1] => aiifkcwh

[2] => aifkccwh

[3] => aiifkccwh

)

\*/

// and we add that to the $output array that contains an index for

// each input string, which contains all possible mutations of that string

$output[$string] = $values1;

}

$output = flatten($output);

$output = array\_unique($output);

$output = array\_values($output);

return $output;

}

### /\* function flat is used to make an array useful for regular expressions \*/

function flat($array)

{

$array = "'".implode("",$array)."'";

return $array;

}

### /\* function blank will return an array having n blank members. blank(2) === array("",""); \*/

function blank($n)

{

$array = array();

while(count($array)< $n+1)

{

array\_push($array,""); // adding blank ""s in the array.

}

return $array;

}

### /\* function checkarray is used to see whether the sequence of $a,$b,$c,$d is found in any of the members of $text \*/

function checkarray($a,$b,$c,$d) // usually sub function serves the purpose. Only once or twice this function is used where we have to test 4 consecutive letters to satisfy the demand of the rule.

{

global $text;

$array = $text;

$combinations = array();

foreach ($a as $aa)

{

foreach ($b as $bb)

{

foreach ($c as $cc)

{

foreach ($d as $dd)

{

$combinations[] = $aa.$bb.$cc.$dd; // creating all permutations and combinations of given 4 arrays and storing it.

}

}

}

}

$counter = array();

for ($i=0;$i<count($array);$i++)

{

for($j=0;$j<count($combinations);$j++)

{

if(strpos($array[$i],$combinations[$j]) !== false)

{

$counter = 1; // if the occurence is found in any of the $text members, $counter is made 1. Otherwise it will remain 0.

break;

}

else

{

$counter = 0;

}

}

if ($counter === 1)

{

break;

}

else

{

$counter = 0;

}

}

return $counter; // returns 1 for match found. 0 for no match found.

}

### /\* function sub is a modified version of checkarray. It searches for the consecutive occurences of $a,$b,$c in the members of array $text.

\* $repeat 0 will mean that the sequence doesnt matter. 1 will mean that the sequence has to be the same.

\*/

function sub($a,$b,$c,$repeat)

{

global $text;

$needle = array();

// for different length and all combinations

if($repeat !== 1)

{

foreach ($a as $aa)

{

foreach ($b as $bb)

{

foreach ($c as $cc)

{

$needle[]=$aa.$bb.$cc; // creating all permutations and combinations.

$needle[]=$aa."+".$bb.$cc; // adding "+" at all possible locations.

$needle[]=$aa.$bb."+".$cc;

$needle[]=$aa."+".$bb."+".$cc;

}

}

}

}

// for similar length arrays and ordered combinations.

if ($repeat === 1)

{

for($i=0;$i<count($a);$i++)

{

$needle[] = $a[$i].$b[$i].$c[$i]; // creating linear patterns.

$needle[] = $a[$i]."+".$b[$i].$c[$i]; // adding "+" at all possible locations.

$needle[] = $a[$i].$b[$i]."+".$c[$i];

$needle[] = $a[$i]."+".$b[$i]."+".$c[$i];

}

}

/\*\*\* map with preg\_quote \*\*\*/

$needle = array\_map('preg\_quote', $needle);

/\*\*\* loop of the array to get the search pattern \*\*\*/

global $first;

foreach ($needle as $pattern)

{

if (($repeat <2 && count(preg\_grep("/$pattern/", $text)) >0) || ($repeat ===2 && strpos(strrev($first), strrev($pattern)) === 0) || ($repeat ===3 && strpos($first,$pattern) === 0) || ($repeat ===4 && strpos(strrev($second), strrev($pattern)) === 0) || ($repeat ===5 && strpos($second,$pattern) === 0))

{

$can = 1; // match found

break;

}

else

{

$can = 0; // match not found

}

}

if ($can === 1)

{

return true;

}

else

{

return false;

}

}

### /\* function arr to search the occurence of a pattern in any of the member of an array \*/

// $text - the array to search for the occurence of pattern

// $a - pattern in regular expression.

function arr($text,$a)

{

foreach ($text as $value)

{

if (preg\_match($a,$value)) // if the regular expression matches the value of array member.

{

$count[] = 1; // match found

break;

}

else

{

$count[] = 0; // match not found

}

}

if (in\_array(1,$count))

{

return true; // if match found

}

else

{

return false; // if match not found

}

}

### /\* function to remove the last n letters from a word in the $text array and replace them with another word \*/

function last($a,$b,$merge)

{

global $text; // bringing $text from subanta.php.

foreach ($text as $value)

{

for($i=0;$i<count($a);$i++)

{ $c = str\_replace("\\","",$a[$i]);

if (preg\_match('/('.$a[$i].')$/',$value) || preg\_match('/('.$c.')$/',$value))

{

$value1[] = substr($value,0,-strlen($a[$i])).$b[$i]; // removing the last letter and substituting it with the corresponding letter of $b.

if ($merge === 1)

{

$value1[] = $value;

}

}

else

{

$value1[] = $value;

}

}

if (array\_unique($value1)===array($value))

{

$value2[] = array\_unique($value1); // in case there is no change.

}

elseif ($merge === 1)

{

$value2[] = array\_unique($value1); // in case there is optional Adeza.

}

else

{

$value2[] = array\_diff($value1,array($value)); // in case there is mandatory Adeza.

}

$value1 = array(); // reseting the $value1 for future use.

}

$text = flatten($value2); // storing the result of $value2 in $text.

$text = array\_unique($text); // keeping only unique members

$text = array\_values($text); // reserialize the numbers.

return $text; // return to the user.

}

### /\* function to remove the first n letters from a word in the $text array and replace them with another word \*/

function first($a,$b,$merge)

{

global $text;

foreach ($text as $value)

{

for($i=0;$i<count($a);$i++)

{

if (preg\_match('/('.$a[$i].')$/',$value))

{

$value1[] = $b[$i].substr($value,strlen($a[$i])); // This function is the same as last function, except that the replacement here is preceded by the member of $b, unlike followed by member of $b in last function.

if ($merge === 1)

{

$value1[] = $value;

}

}

else

{

$value1[] = $value;

}

}

if (array\_unique($value1)===array($value))

{

$value2[] = array\_unique($value1);

}

elseif ($merge === 1)

{

$value2[] = array\_unique($value1);

}

else

{

$value2[] = array\_diff($value1,array($value));

}

$value1 = array();

}

$text = flatten($value2);

$text = array\_unique($text);

$text = array\_values($text);

return $text;

}

### /\* vriddhi function \*/

function vriddhi($a)

{ $ac = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O",);

$acreplace = array("A","A","E","E","O","O","Ar","Ar","Al","Al","E","O","E","O",);

foreach ($a as $value)

{

$out[] = str\_replace($ac,$acreplace,$value); // replacing the vowels with their corresponding vRddhi letters.

}

return $out;

}

### /\* guna function \*/

function guna($a)

{ $ac = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O",);

$acreplace = array("a","a","e","e","o","o","ar","ar","al","al","e","o","e","o",);

foreach ($a as $value)

{

$out[] = str\_replace($ac,$acreplace,$value); // replacing the vowels with their corresponding guNa letters.

}

return $out;

}

### /\* dirgha function \*/

function dirgha($a)

{ $ac = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O",);

$acreplace = array("A","A","I","I","U","U","F","F","F","F","e","o","E","O",);

foreach ($a as $value)

{

$out[] = str\_replace($ac,$acreplace,$value); // replacing the vowels with their corresponding dIrgha letters.

}

return $out;

}

### /\* anekAca function \*/

// function to test whether the word is ekAc or anekAc i.e. whether there is only one vowel in it or more than one vowel in it.

function anekAca($a)

{

$ac = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O",); // vowel list.

$b = preg\_split('/['.flat($ac).']/',$a); // splitting the word by vowels.

if (count($b)>2) // if there are more than 2 pieces, that means there are more than one vowel mandatorily.

{

return true; // returns true to the user.

}

else

{

return false; // returns false to the user.

}

}

### /\* ends function \*/

// to find if there is any member of array which end in a particular member of another array.

// 0 -> doesn't start with but ends with it. 1 -> ends with it or is equal to it. 2 -> is equal to it.

function ends($a,$b,$n)

{

foreach ($a as $aa)

{

foreach ($b as $bb)

{

if ($n===0)

{

if (strpos($aa,$bb)>0 && strpos(strrev($aa),strrev($bb)) ===0) // strpos($aa,$bb)>0 means that it doesn't start with it. the next means that it ends with it.

{

$can[] = 1;

}

else

{

$can[] = 0;

}

}

if ($n===1)

{

if (strpos($aa,$bb) >= 0 && strpos(strrev($aa),strrev($bb)) ===0) // strpos($aa,$bb) >= 0 means that it starts with it or is included in it. the next means that it ends with it.

{

$can[] = 1;

}

else

{

$can[] = 0;

}

}

if ($n===2)

{

if ($aa===$bb) // they are the same.

{

$can[] = 1;

}

else

{

$can[] = 0;

}

}

}

}

if (in\_array(1,$can))

{

return true; // if any time $can becomes 1, it means that match is found.

}

else

{

return false; // match not found.

}

}

### /\* function Ti to remove the Ti \*/

// $merge 0 - mandatory Adezas, 1 - optional Adezas.

function Ti($merge)

{

global $text;

$te1 = '/(['.pc('ac').'])(['.pc('hl').']\*)([+])/'; // vowel - unlimited combinations of consonants - '+' combination.

$te2 = '$3'; // keep only "+" and remove the first two members.

foreach ($text as $value)

{

$val[] = preg\_replace($te1,$te2,$value); // removing the first two members.

}

if ($merge===0) // mandatory Adezas

{

$text = $val;

}

if ($merge === 1) // optional Adezas

{

$text = array\_merge($text,$val);

}

$val=array();

$text = array\_unique($text);

$text = array\_values($text);

return $text; // giving the result to the user.

}

### /\* function mit \*/

// $pattern is the pattern to search in the word to add mit pratyaya.

// $b is the addition to upadha

// $merge 0 is replacement, 1 is optional.

function mit($pattern,$b,$merge)

{ global $text;

$te1 = '/(['.pc('ac').'])(['.pc('hl').']\*)([+])/';

$te2 = '$1'.$b.'$2$3'; // Adding the mit Agama after the last vowel. 'midaco'ntyAtparaH'.

foreach ($text as $value)

{

if (preg\_match($pattern,$value))

{

$val[] = preg\_replace($te1,'$1n$2$3',$value); // executing mit Agama.

}

else

{

$val[] = $value;

}

}

if ($merge===0)

{

$text = $val;

}

if ($merge === 1)

{

$text = array\_merge($text,$val);

}

$val=array();

$text = array\_unique($text);

$text = array\_values($text);

return $text;

}

### /\* function samprasarana \*/

function samprasarana($a,$merge)

{

$yan = array("y","v","r","l");

$yanik = array("i","u","f","x"); // list of samprasAraNa for yaN.

foreach ($a as $value)

{

$b[] = str\_replace($yan,$yanik,$a); // doing samprasARaNa.

}

return $b;

}

### /\* function itcheck \*/

// $a to contain the array of it markers to check.

// $b. 0 for it. 1 for itprakriti. 2 for itpratyaya

function itcheck($a,$b)

{

global $text; global $it; global $itprakriti; global $itpratyaya;

if ($b===0)

{

if(count(array\_intersect($it,$a))>0) // finds in $it.

{

return TRUE;

}

else

{

return FALSE;

}

}

if ($b===1)

{

if(count(array\_intersect($itprakriti,$a))>0) // finds in $itprakriti

{

return TRUE;

}

else

{

return FALSE;

}

}

if ($b===2)

{

if(count(array\_intersect($itpratyaya,$a))>0) // finds in $itpratyaya

{

return TRUE;

}

else

{

return FALSE;

}

}

}

### /\* function it (to find an it marker). The function adds the marker to its arrays. \*/

// $pattern is the pattern to look for

// $a 0 for finding it in the whole word. 1 for it in prakriti. 2 for it in pratyaya.

function it($pattern)

{

global $text; global $it; global $itprakriti; global $itpratyaya;

foreach ($text as $value)

{

$b = preg\_split($pattern,$value,null,PREG\_SPLIT\_DELIM\_CAPTURE); // spliting the string by the pattern.

for($i=1;$i<((count($b)+1)/2);$i++)

{

$b[$i\*2-1]= str\_replace("!","",$b[$i\*2-1]); // removing !

$b[$i\*2-1]= str\_replace("+","",$b[$i\*2-1]); // removing +

$it = array\_merge($it,array($b[$i\*2-1])); // adding the it marker to the array $it.

$it = array\_unique($it);

$it = array\_values($it);

}

}

foreach ($text as $value)

{

$c = explode("+",$value);

$b = preg\_split($pattern,$c[0],null,PREG\_SPLIT\_DELIM\_CAPTURE);

for($i=1;$i<((count($b)+1)/2);$i++)

{

$b[$i\*2-1]= str\_replace("!","",$b[$i\*2-1]);

$b[$i\*2-1]= str\_replace("+","",$b[$i\*2-1]);

$itprakriti = array\_merge($itprakriti,array($b[$i\*2-1])); // adding it to array $itprakriti

$itprakriti = array\_unique($itprakriti);

$itprakriti = array\_values($itprakriti);

}

}

foreach ($text as $value)

{

$c = explode("+",$value);

$pattern = str\_replace("[+]","^",$pattern);

$b = preg\_split($pattern,$c[1],null,PREG\_SPLIT\_DELIM\_CAPTURE);

for($i=1;$i<((count($b)+1)/2);$i++)

{

$b[$i\*2-1]= str\_replace("!","",$b[$i\*2-1]);

$b[$i\*2-1]= str\_replace("+","",$b[$i\*2-1]);

$itpratyaya = array\_merge($itpratyaya,array($b[$i\*2-1])); // adding it to array $itpratyaya

$itpratyaya = array\_unique($itpratyaya);

$itpratyaya = array\_values($itpratyaya);

}

}

}

### /\* function to find out it markers when they can occur only in the pratayayas. \*/

function itprat($pattern)

{

global $text; global $it; global $itprakriti; global $itpratyaya;

foreach ($text as $value)

{

$c = explode("+",$value);

$b = preg\_split($pattern,$c[1],null,PREG\_SPLIT\_DELIM\_CAPTURE);

for($i=1;$i<((count($b)+1)/2);$i++)

{

$b[$i\*2-1]= str\_replace("!","",$b[$i\*2-1]);

$b[$i\*2-1]= str\_replace("+","",$b[$i\*2-1]);

$itpratyaya = array\_merge($itpratyaya,array($b[$i\*2-1])); // adding it to array $itpratyaya

$itpratyaya = array\_unique($itpratyaya);

$itpratyaya = array\_values($itpratyaya);

}

}

}

/\* function antya to do antyAdeza \*/

// $a - array where we want to do antyAdeza. $rep - replacement in place of the last letter.

function antya($a,$rep)

{

foreach ($a as $value)

{

$value1[] = substr($value,0,-1).$rep;

}

return $value1;

}

### /\* Functions which are not used in the code \*/

/\* Function f to find the nth letter in the word \*/

function f($text,$n) // Not used in code.

{

$p = str\_split($text); // splits the word in its letters.

$text = $p[$n-1]; // finds the nth letter. 1st letter is $p[0], 2nd $p[1]... nth is $p[$n-1]

return $text; // returning the result to the user.

}

/\* find string after removing 'n' letters from beginning of the word (sa for string after) \*/

function sa($text,$n) // Not used in code.

{

$p = str\_split($text);

$a = "";

for ($q=0;$q<$n;$q++)

{ $text = ltrim($text,$p[$q]); }

return $text;

}

/\* find 'n'th letter from the end in a word (r for reverse direction) \*/

function r($text,$n) // Not used in code

{

$p = str\_split($text);

$text = $p[count($p)-$n];

return $text;

}

/\* find the string remaining after removing n characters from the end (sb for string before) \*/

function sb($text, $n) // Not used in code.

{

$p = str\_split($text);

$a= "";

for ($q=0;$q<$n;$q++)

{$text = chop($text,$p[count($p)-$q-1]);}

return $text;

}

/\* function lopa is not used in the code because it is not stable \*/

function lopa ($kantha,$talu,$murdha,$oshtha,$location,$merge)

{

global $text;

// this is an array of the input values to use below

// get all possible combianations of $kantha+$talu+$murdha

//$combinations = get\_string\_combinations();

$combinations = array();

foreach($kantha as $k) // "a","k","g","n","h"

{

foreach($talu as $t) // "i","c","j","y","s"

{

foreach($murdha as $m) // "f","w","q","r","z"

{

foreach ($oshtha as $o)

{

$combinations[] = $k.$t.$m.$o;

}

}

}

}

//print\_r($combinations);

$values1 = array();

foreach ($text as $stti => $string) // 'aifkcwh', 'aifkcwhsz', 'kim', 'aif'

{

$values = array($string);

reset($values);

// loop through $values using an array pointer

// while the current array pointer position is not null/false

while(current($values)!==false)

{

// on the first iteration, $values will be the inputted string

// from $strings\_to\_test... our example is "aifkcwh"

$value = current($values);

if ($merge === 1)

{

$values1[] = current($values);

}

// for each possible combination of $kantha.$talu.$murdha

// let's say our first combination is "aif"

foreach($combinations as $ci => $combination)

{

// look and see if the current value we are looking

// at ("aifkcwh") contains this combination string

if (strpos($value,$combination)!==false)

{ $posterior = substr($value,strpos($value,$combination)+$location+2);

$previous = chop($value,$posterior);

$previous = substr($previous,0,strlen($previous));

$newval = $previous.$posterior;

// echo $value."<br>".$combination."</br>".$previous."</br>".$posterior."</br>";

// have we already recorded this mutation?

if (!in\_array($newval,$values1))

{

// if not... append it to the array $values

$values1[] = $newval;

// now, values would go from being = array([0]=>'aifkcwh');

// to array ([ 0] => 'aifkcwh', [1] => 'aiifkcwh' );

}

else

{

}

}

} // <-- end of the foreach statement, this will go through all combinations

// in our combinations array for this particular value which is currently aifkcwh

next($values);

}

$output[$string] = $values1;

}

// print\_r($output);

$output = flatten($output);

$output = array\_unique($output);

$output = array\_values($output);

return $output;

}

/\* function nosavarna is not used in the code. It is for finding the array of savarNas without the letter itself. \*/

function nosavarna($c)

{

global $ac;

$i = array("i","I");$u = array("u","U");$f = array("f","F","x","X");

if ( $c === $i[0] ) {$non = array\_diff($ac, $i);}

if ( $c === $i[1] ) {$non = array\_diff($ac, $i);}

if ( $c === $u[0] ) {$non = array\_diff($ac, $u);}

if ( $c === $u[1] ) {$non = array\_diff($ac, $u);}

if ( $c === $f[0] ) {$non = array\_diff($ac, $f);}

if ( $c === $f[1] ) {$non = array\_diff($ac, $f);}

if ( $c === $f[2] ) {$non = array\_diff($ac, $f);}

if ( $c === $f[3] ) {$non = array\_diff($ac, $f);}

return $non;

}

/\* function get\_string\_combinations is a subset of the function dvitva and lopa. In modified version, this code segment is redundant. \*/

function get\_string\_combinations()

{

$kantha = array("a","k","g","n","h");

$talu = array("i","c","j","y","s");

$murdha = array("f","w","q","r","z");

$combinations = array();

foreach($kantha as $k) // "a","k","g","n","h"

{

foreach($talu as $t) // "i","c","j","y","s"

{

foreach($murdha as $m) // "f","w","q","r","z"

{

$combinations[] = $k.$t.$m;

}

}

}

// this gives us an array if 125 items

/\*

$combinations =

Array

(

[0] => aif

[1] => aiw

[2] => aiq

[3] => air

[4] => aiz

....

[121] => hsw

[122] => hsq

[123] => hsr

[124] => hsz

)

\*/

return $combinations;

}

/\* function merge is not used in the code \*/

function merge($text,$text1)

{

$text = array\_merge($text,$text1);

$text = array\_unique($text);

$text = array\_values($text);

return $text;

}

/\* function blank1 will return an array having n members of $a. blank("a",2) === array("a","a"); Not used in code. \*/

function blank1($a,$n)

{

$array = array();

while(count($array)< $n+1)

{

array\_push($array,$a);

}

return $array;

}

/\* Function to find savarna of a given letter from the given array. Not used in code. \*/

function sl($text,$array) // Known issue - words having two Asyas.

{

global $kantha,$talu,$murdha,$danta,$oshtha,$nasika,$kanthatalu,$kanthoshtha,$dantoshtha,$sprushta,$ishatsprushta,$vivruta,$samvruta,$aghosha,$alpaprana,$ghosha,$mahaprana,$ac,$udatta,$anudatta,$svarita,$shvasa,$nada,$vivara,$samvara,$hl,$ru,$e;

// defining an array for sthAna

$i=0;

if (in\_array($text,$kantha)) { $sthana[$i] = $kantha; $i++; }

if (in\_array($text,$talu)) { $sthana[$i] = $talu; $i++; }

if (in\_array($text,$murdha)) { $sthana[$i] = $murdha; $i++; }

if (in\_array($text,$danta)) { $sthana[$i] = $danta; $i++; }

if (in\_array($text,$oshtha)) { $sthana[$i] = $oshtha; $i++; }

if (in\_array($text,$nasika)) { $sthana[$i] = $nasika; $i++; }

if (in\_array($text,$kanthatalu)) { $sthana[$i] = $kanthatalu; $i++; }

if (in\_array($text,$kanthoshtha)) { $sthana[$i] = $kanthoshtha; $i++;}

if (in\_array($text,$dantoshtha)) { $sthana[$i] = $dantoshtha; $i++; }

// defining an array for Abhyantara prayatna for consonants

$j=0;

if (in\_array($text,$sprushta)) { $abhyantara[$j] = $sprushta; $j++; }

if (in\_array($text,$ishatsprushta)) { $abhyantara[$j] = $ishatsprushta; $j++; }

if (in\_array($text,$vivruta)) { $abhyantara[$j] = $vivruta; $j++; }

if (in\_array($text,$samvruta)) { $abhyantara[$j] = $samvruta; $j++; }

// defining an array for bAhya prayatna for consonants

$k=0;

if (in\_array($text,$aghosha)) { $ghosh[$k] = $aghosha; $k++; }

if (in\_array($text,$alpaprana)) { $prana[$k] = $alpaprana; $k++; }

if (in\_array($text,$ghosha)) { $ghosh[$k] = $ghosha; $k++; }

if (in\_array($text,$mahaprana)) { $prana[$k] = $mahaprana; $k++; }

// defining an array for bAhya prayatna of vowels

$u=0;

if (in\_array($text,$ac)) { $svar[$u] = $udatta; $u++; }

// Finding out intersections of sthAna, Abhyantara prayatna and bAhya prayatnas of the given letter and the given array.

if(empty($sthana)===FALSE)

{

$sthanasamya = array\_intersect(flatten($sthana),$array);

//echo "The letters in the pratyAhAra with same sthAna (Asya) as the letter input are: ".implode(",",$sthanasamya)."</br>";

}

if(empty($abhyantara)===false)

{

$abhyantarasamya = array\_intersect(flatten($abhyantara),$array);

//echo "The letters in the pratyAhAra with the same Abhyantara prayatna as the letter input are: ".implode(",",$abhyantarasamya)."</br>";

}

if(empty($ghosh)===FALSE)

{

$ghoshasamya = array\_intersect(flatten($ghosh),$array);

//echo "The letters in the pratyAhAra with the same ghoSa as the letter input are: ".implode(",",$ghoshasamya)."</br>";

}

if(empty($prana)===FALSE)

{

$pranasamya = array\_intersect(flatten($prana),$array);

//echo "The letters in the pratyAhAra with the same prANa as the letter input are: ".implode(",",$pranasamya)."</br>";

}

if(empty($svar)===false)

{

if(in\_array($text,$ac))

{ $svarasamya = array\_intersect(flatten($svar),$array,$ac);

//echo "The letters in the pratyAhAra with the same udAtta/anudAtta/svarita as the letter input are: ".implode(",",$svarasamya)."</br>";

}

}

if(empty($sthanasamya)===false && empty($abhyantarasamya)===false && empty($ghoshasamya)===false && empty($pranasamya)===false)

{ $l = array\_intersect($sthanasamya,$abhyantarasamya,$ghoshasamya,$pranasamya); }

if(empty($sthanasamya)===false && empty($abhyantarasamya)===false && empty($ghoshasamya)===false)

{ $m = array\_intersect($sthanasamya,$abhyantarasamya,$ghoshasamya); }

if(empty($sthanasamya)===false && empty($abhyantarasamya)===false)

{ $n = array\_intersect($sthanasamya,$abhyantarasamya); }

if(empty($sthanasamya)===false && empty($abhyantarasamya)===false && empty($pranasamya)===false)

{ $o = array\_intersect($sthanasamya,$abhyantarasamya,$pranasamya); }

if(empty($sthanasamya)===false && empty($ghoshasamya)===false)

{ $p = array\_intersect($sthanasamya,$ghoshasamya); }

if(empty($sthanasamya)===false && empty($pranasamya)===false)

{ $q = array\_intersect($sthanasamya,$pranasamya); }

// Defining savarNas for consonants

if(in\_array($text,$hl))

{

if(empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($ghoshasamya)===false&&empty($pranasamya)===FALSE&&empty($l)===false)

{//echo "four match";

$savarna = implode(", ",$l);

}

elseif (empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($ghoshasamya)===false&&empty($m)===false)

{//echo "three match";

$savarna = implode(", ",$m);

}

elseif (empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($pranasamya)===false&&empty($o)===false)

{//echo "three match";

$savarna = implode(", ",$o);

}

elseif (empty($sthanasamya)===false&&empty($abhyantarasamya)===false&&empty($n)===false)

{//echo "Two match";

$savarna = implode(", ",$n);

}

elseif (empty($sthanasamya)===false&&empty($ghoshasamya)===false&&empty($p)===false)

{//echo "Two match";

$savarna = implode(", ",$p);

} elseif (empty($sthanasamya)===false&&empty($pranasamya)===false&&empty($q)===false)

{//echo "Two match";

$savarna = implode(", ",$q);

}

else

{//echo "no match";

$savarna = implode(", ",$sthanasamya);

}

}

// defining savarNas for vowels

else

{

if (in\_array($text,$ru)||in\_array($text,$e))

{// patch for $ru

for($i=0;$i<4;$i++)

{

if ($text === $ru[$i])

{

if (in\_array($text,$array))

{ $savarna = "f, F, x, X"; }

else

{ $savarna = ""; }

}

// patch for non sAvarNya of e,E,o,O

elseif ($text === $e[$i])

{

if (in\_array($text,$array))

{ $savarna = $text; }

else

{ $savarna = ""; }

}

}

}

else

{// In case of other vowels.

$savarna = implode(", ",$sthanasamya);

}

if (in\_array($text,$array))

{

$savarna = $text;

}

}

// giving output to the browser for savarNa letter

//echo "The savarna letter of '".$text."' among the given pratyAhAra is: ".$savarna;

// stores that savarNa letter in memory.

return $savarna;

}

/\* Function one1 is for replacing one letter in the whole array of $text with another letter. Not used in code \*/

// There are three arguments, $a is the array which you want to change, $b is the array which will be put in place of the replaced one.

// $merge can take two values. 0 will mean that the whole $text will be replaced with the new replaced values. Used in case of mandatory Adezas.

// 1 will mean that $text will not be replaced, but the replaced values will be added to it. Used in case of optional Adezas.

function one1($text,$a,$b,$merge)

{

for($i=0;$i<count($a);$i++)

{

$text = str\_replace($a[$i],$b[$i],$text);

}

return $text;

}

/\* Function two is for replacing one letter in the whole array of $text with another letter. Not used in code \*/

// There are five arguments, $a,$b are the arrays which you want to change, $c,$d are the arrays which will be put in place of the replaced one.

// $merge can take two values. 0 will mean that the whole $text will be replaced with the new replaced values. Used in case of mandatory Adezas.

// 1 will mean that $text will not be replaced, but the replaced values will be added to it. Used in case of optional Adezas.

function two1($text,$a,$b,$c,$d,$merge)

{

for($i=0;$i<count($a);$i++)

{

for($j=0;$j<count($b);$j++)

{

$text = str\_replace($a[$i].$b[$j],$c[$i].$d[$j],$text);

}

}

return $text;

}

/\* Function three1 is for replacing one letter in the whole array of $text with another letter. Not used in code \*/

// There are seven arguments, $a,$b,$c are the arrays which you want to change, $d,$e,$f are the arrays which will be put in place of the replaced one.

// $merge can take two values. 0 will mean that the whole $text will be replaced with the new replaced values. Used in case of mandatory Adezas.

// 1 will mean that $text will not be replaced, but the replaced values will be added to it. Used in case of optional Adezas.

function three1($text,$a,$b,$c,$d,$e,$f,$merge)

{

for($i=0;$i<count($a);$i++)

{

for($j=0;$j<count($b);$j++)

{

for($k=0;$k<count($c);$k++)

{

$text = str\_replace($a[$i].$b[$j].$c[$k],$d[$i].$e[$j].$f[$k],$text);

}

}

}

return $text;

}

/\* An attempt to create an all encompassing function

\* name is panini

\* three arrays for checking,

\* one for adding additional checking in addition to sub function

\* one for adding whether pratyaya / pada

\* one for the 0/1/2/3/4 etc of sub function,

\* three for the substituted,

\* three for substitution,

\* one for 0/1 of one/two/three function,

\* one for english sutra name,

\* one for english sutra number

\* one for hindi sutra name,

\* one for hindi sutra number

\*

\*/

?>

## 4. mystyle.css

This is stylesheet for HTML display. Borrowed from Mr. Marcis Gasuns.

Only some of its elements are used in the code.

Rest I have kept intact.

The code is as follows:

/\* This code is borrowed from Mr. Marcis Gasuns. \*/

body {

font-family: "Arial Unicode MS, Sans-Serif";

font-size: 16pt;

font-style: normal;

font-weight: normal;

color: #333333;

background: white;

padding-left: 5px;

}

/\* hymn name \*/

h3 {

font-style: italic;

color: #ff0000;

}

p {

float: inherit;

margin: 0 0 0 0;

}

table {

width: 800px;

}

div {

color: #8b0000;

background-color: #d0d0d0;

font-size: 16pt;

font-family: "Siddhanta";

padding-right: 40px;

left: 20px;

}

/\* vidhi sUtra \*/

/\* We use this to style vidhi sUtras. \*/

p.sa {

color: #8b0000;

background-color: #d0d0d0;

font-size: 16pt;

font-family: "Siddhanta"; /\* Default font is kept Siddhanta for proper display of upadhmAnIya and jihvAmUlIya. \*/

padding-right: 40px;

left: 20px;

}

/\* notes \*/

/\* We use this to style notes \*/

p.hn {

color: black;

background-color: #ffffcc;

font-family: "Siddhanta";

padding-left: 0px;

}

/\* sutras etc except vidhi sutra \*/

/\* most probably paribhASAs, saJjJA sUtra etc are recorded in this \*/

p.pa {

background: #ccffcc;

font-family: "Siddhanta";

}

/\* Error message \*/

/\* using red color to give error messages to user \*/

p.red {

color: black;

background-color: #ff0033;

font-family: "Siddhanta";

padding-top: 2px;

padding-bottom: 2px;

}

/\* the next styles are not used in code. kept reserved for future usages. \*/

/\* rv01.001.01 \*/

p.stamp {

font-size: 14pt;

font-weight: bold;

float: none;

padding-top: 2px;

padding-bottom: 2px;

}

p.comm {

font-style: italic;

}

/\* elizarenkova \*/

p.form {

padding-top: 4px;

color: black;

background: white;

font-family: "Siddhanta";

padding-left: 3px;

}

/\* geldner \*/

p.de {

color: #5b2e00;

background: white;

font-family: "Charter Indologique";

padding-left: 6px;

padding-right: 50px;

}

/\* griffith \*/

p.en {

color: #2c577b;

background: white;

font-family: "Charter Indologique";

padding-left: 9px;

padding-right: 50px;

}

## 5. script.js

This is a javascript file which makes subanta.html interactive. Whenever the user clicks on specified radio buttons, this script sends information to ajax.php and checks whether there is any special additional message required to be displayed to the user for additional information gathering. It shows the results of ajax.php back to user in the same subanta.html file.

The code used is as follows:

$(document).ready(function() {

// Check for the blur event

var response = '';

$.getData = function(){

$.ajax({

type: "POST", // data is transferred by POST

url: "ajax.php", // data is sent to ajax.php

async : false,

data: $( "form" ).serialize()

})

.done(function( data ) {

response = data; // serialises data got from form of subanta.html

});

}

$( document ).on( "click", "input:radio[name=gender]", function() {// if the user clicks on gender radio button.

if($('input:radio[name=gender]:checked').val() === "m"){ // if the checked gender is "m"

$('#step').val('1'); // send value of 'step' as '1' to ajax.php

$.getData(); // sends data to ajax.php

$('#step1').html(response); // gets the output from ajax.php and displays it on subanta.html in div class 'step1'. Similar for all the rest of code.

}

});

$( document ).on( "click", "input:radio[name=cond1\_1\_1]", function() {

if($('input:radio[name=cond1\_1\_1]:checked').val() === '5'){

$('#step').val('1\_1\_1\_5');

$.getData();

$('#step2').html(response);

}else if($('input:radio[name=cond1\_1\_1]:checked').val() === '6'){

$('#step').val('1\_1\_1\_6');

$.getData();

$('#step2').html(response);

}else{

$('#step2').html(''); // output in div class step2

}

});

$( document ).on( "click", "input:radio[name=cond1\_1\_1\_6\_3]", function() {

if($('input:radio[name=cond1\_1\_1\_6\_3]:checked').val() == '1'){

$('#step').val('1\_1\_1\_6\_3\_1');

$.getData();

$('#step3').html(response);

}else{

$('#step3').html(''); // output in div class step3. These steps are levels of user feedback.

}

});

$( document ).on( "click", "input:radio[name=cond1\_4]", function() {

if($('input:radio[name=cond1\_4]:checked').val() == '2'){

$('#step').val('1\_4\_2');

$.getData();

$('#step2').html(response);

}else if($('input:radio[name=cond1\_4]:checked').val() == '3'){

$('#step').val('1\_4\_3');

$.getData();

$('#step2').html(response);

}else if($('input:radio[name=cond1\_4]:checked').val() == '4'){

$('#step').val('1\_4\_4');

$.getData();

$('#step2').html(response);

}else if($('input:radio[name=cond1\_4]:checked').val() == '6'){

$('#step').val('1\_4\_6');

$.getData();

$('#step2').html(response);

}else{

$('#step2').html('');

}

});

$( document ).on( "click", "input:radio[name=cond1\_9]", function() {

if($('input:radio[name=cond1\_9]:checked').val() === '1'){

$('#step').val('1\_9\_1');

$.getData();

$('#step2').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_10]", function() {

if($('input:radio[name=cond1\_10]:checked').val() === '2'){

$('#step').val('1\_10\_2');

$.getData();

$('#step2').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_12]", function() {

if($('input:radio[name=cond1\_12]:checked').val() === '5'){

$('#step').val('1\_12\_5');

$.getData();

$('#step2').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_12\_5]", function() {

if($('input:radio[name=cond1\_12\_5]:checked').val() === '1'){

$('#step').val('1\_12\_5\_1');

$.getData();

$('#step3').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_12\_5\_1]", function() {

if($('input:radio[name=cond1\_12\_5\_1]:checked').val() === '1'){

$('#step').val('1\_12\_5\_1\_1');

$.getData();

$('#step4').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_12\_5\_1\_1]", function() {

if($('input:radio[name=cond1\_12\_5\_1\_1]:checked').val() === '1'){

$('#step').val('1\_12\_5\_1\_1\_1');

$.getData();

$('#step5').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_12\_5\_1\_1\_1]", function() {

if($('input:radio[name=cond1\_12\_5\_1\_1\_1]:checked').val() === '1'){

$('#step').val('1\_12\_5\_1\_1\_1\_1');

$.getData();

$('#step6').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_13]", function() {

if($('input:radio[name=cond1\_13]:checked').val() === '1'){

$('#step').val('1\_13\_1');

$.getData();

$('#step2').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond1\_17]", function() {

if($('input:radio[name=cond1\_17]:checked').val() === '2'){

$('#step').val('1\_17\_2');

$.getData();

$('#step2').html(response);

}

});

/\* strIliGga \*/

$( document ).on( "click", "input:radio[name=gender]", function() {

if($('input:radio[name=gender]:checked').val() === 'f'){

$('#step').val('2');

$.getData();

$('#step1').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond2\_1]", function() {

if($('input:radio[name=cond2\_1]:checked').val() == '2'){

$('#step').val('2\_1\_2');

$.getData();

$('#step2').html(response);

}else{

$('#step2').html('');

}

});

$( document ).on( "click", "input:radio[name=cond2\_1\_2\_1]", function() {

if($('input:radio[name=cond2\_1\_2\_1]:checked').val() === '5'){

$('#step').val('2\_1\_2\_1\_5');

$.getData();

$('#step3').html(response);

}else if($('input:radio[name=cond2\_1\_2\_1]:checked').val() === '6'){

$('#step').val('2\_1\_2\_1\_6');

$.getData();

$('#step3').html(response);

}else{

$('#step3').html('');

}

});

$( document ).on( "click", "input:radio[name=cond2\_1\_2\_1\_6\_3]", function() {

if($('input:radio[name=cond2\_1\_2\_1\_6\_3]:checked').val() == '1'){

$('#step').val('2\_1\_2\_1\_6\_3\_1');

$.getData();

$('#step4').html(response);

}else{

$('#step4').html('');

}

});

/\* napuMsakaliGga \*/

$( document ).on( "click", "input:radio[name=gender]", function() {

if($('input:radio[name=gender]:checked').val() === "n"){

$('#step').val('3');

$.getData();

//$('.innerStep').html('');

$('#step1').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond3\_3]", function() {

if($('input:radio[name=cond3\_3]:checked').val() == '2'){

$('#step').val('3\_3\_2');

$.getData();

$('#step2').html(response);

}

});

$( document ).on( "click", "input:radio[name=cond3\_3\_2]", function() {

if($('input:radio[name=cond3\_3\_2]:checked').val() == '1'){

$('#step').val('3\_3\_2\_1');

$.getData();

$('#step3').html(response);

}

});

});

## 6. subanta.html

subanta.html is the frontend of the code. The user enters his words for declention here. It is integrally linked with script.js and ajax.php. After getting necessary information from user, the data is passed to subanta.php for processing according to SK.

The code used is as follows:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html>  
<head>  
<!--... Defining UTF-8 as our default character set, so that devanagari is displayed properly. -->  
<meta charset="UTF-8">  
<!--... Defining CSS -->  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
<!--... including jquery. -->  
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js" ></script>  
<!--... script.js is javascript which we use to make the frontend interective for user feedback. Whenever the user clicks the radio button specified in script.js, the data is sent to ajax.php and then output from ajax.php is shown in the frontend. This process may continue as long as you want. -->  
<script src="script.js" ></script>  
</head>  
<body>  
  
<div>  
<!--... submitting data to subanta.php. -->  
<form action="subanta.php" method = "get" id="frm">  
<!--... first is the input word, which the user enters. -->  
प्रकृतिः : <input type="text" name="first" id="first"><br>  
<!--... Defining gender of the word. -->  
<input type="radio" name="gender" value="m">पुँल्लिङ्ग  
<input type="radio" name="gender" value="f">स्त्रीलिङ्ग  
<input type="radio" name="gender" value="n">नपुंसकलिङ्ग</br>  
<!--... Defining transliteration of the word. Default is kept as Devanagari. User may change it according to his wish. -->  
<input type="radio" name="tran" value="Devanagari" checked>Devanagari  
<input type="radio" name="tran" value="IAST">IAST  
<input type="radio" name="tran" value="SLP1">SLP1<br>  
<!--... Divs to record the output from ajax.php. step1, step2, step3 etc are the levels. see ajax.php to decide where ajax will send its output. Maximum user feedback level for our coding was 6. So, kept it 6. It can be easily added if need be. -->  
<div id="response">  
 <div id="step1" class="innerStep"></div>  
 <div id="step2" class="innerStep"></div>  
 <div id="step3" class="innerStep"></div>  
 <div id="step4" class="innerStep"></div>  
 <div id="step5" class="innerStep"></div>  
 <div id="step6" class="innerStep"></div>  
</div>  
<!--... Submit button -->  
<input type="submit">  
<!--... Sending the step id to subanta.php hidden. -->  
<input type="hidden" name="step" id="step" value="0" />  
</form>  
</div>  
  
<!--... Information about the code. -->  
<p class = hn>  
Developed by: Dr. Dhaval Patel & Sivakumari Katuri.  
Layout assistance by: Mr. Marcis Gasuns.<br>  
Available under <a href="http://www.gnu.org/copyleft/gpl.html" traget="\_blank">GNU licence</a>. <br>  
Note: The source code for this program is available on Github <a href = "https://github.com/drdhaval2785/sanskrit" target="\_blank">here.</a><br>  
For those who find github clumsy, latest code can be accessed manually <a href="http://lanover.com/lan/sanskrit/" target="\_blank">here</a>.<br>  
For those who want to install the code for offline usage, instruction can be seen in readme file <a href="https://github.com/drdhaval2785/sanskrit" target="\_blank">here</a>.<br>   
Places where we take user feedback can be accessed <a href="http://lanover.com/lan/sanskrit/ajax%20requirement.docx" target="\_blank">here</a>. <br>  
For better rendering of output, use <a href = "http://svayambhava.org/index.php/en/" target="\_blank">Siddhanta font.</a></p>  
<!--... Disclaimer for IAST -->  
For <a href = "http://en.wikipedia.org/wiki/International\_Alphabet\_of\_Sanskrit\_Transliteration">IAST scheme</a>, please use the lowercase letters.</p>  
<!--... SLP1 letter set. code works primarily on SLP1. Other transliterations are first converted to SLP1 and then processed. -->  
<p>For SLP1, use this scheme for entering words: </p>  
aAiIuUfFxXeEoOMH</br>  
kKgGN</br>  
cCjJY</br>  
wWqQR</br>  
tTdDn</br>  
pPbBm</br>  
yrlvSzsh;  
</body>  
</html>

## 7. subanta.php

This is the main crux of the code. This is the operative part of the code. All information gathered by subanta.html, script.js and ajax.php are gathered and processed by this code. This is a step by step derivation program for vibhaktis. The code is fully illustrated to show what is happenning in the code.

The code used is as follows.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="mystyle.css">

</link>

</meta>

</head>

<body>

<?php

//header('Content-type: text/html; charset=utf-8');

/\* This code is developed by Dr. Dhaval Patel (drdhaval2785@gmail.com) of www.sanskritworld.in and Ms. Sivakumari Katuri.

\* Layout assistance by Mr Marcis Gasuns.

\* Available under GNU licence.

\* Version 2.0 date 4 September 2014

\* The latest source code is available at https://github.com/drdhaval2785/sanskrit

\* Acknowledgements: I extend my heartfelt thanks to Ananda Loponen for the code to convert devanagari and various sanskrit transliterations. That can be accessed at http://www.ingmardeboer.nl/php/diCrunch.php?act=help.

\* I also extend my gratitude to gloomy.penguin of stackoverflow.com, who helped me create dvitva and lopa functions, without which I would be handicapped.

\* For setup, copy and paste subanta.php, subanta.html, script.js, ajax.php, function.php, mystyle.css, slp-dev.php and dev-slp.php to your localhost and server and run subanta.html.

\* subanta.html is the frontend for the code.

\* ajax.php and script.js are codes which asks for user feedback for particular words.

\* function.php stores the frequently used functions in this code (The description on how to use the code is there in function.php).

\* subanta.php is the code which actually gives the output of the word derivation.

\* slp-dev.php is for converting SLP1 data to Devanagari.

\* dev-slp.php is for converting Devanagari data to SLP1.

\* Mystyle.css is stylesheet where you can change your preferences.

\* The code uses jquery.

\* The description part uses Howard Kyoto protocol.

\* The coding uses SLP1 transliteration.

\*/

### /\* Including arrays and functions \*/

include "function.php"; // includes the file function.php which is collection of functions used in this code.

include "slp-dev.php"; // includes code for conversion from SLP to devanagari,

include "dev-slp.php"; // includes code for devanagari to SLP.

### /\* hides error reports. \*/

error\_reporting(0);

### /\* set execution time to an hour \*/

ini\_set('max\_execution\_time', 36000);

/\* set memory limit to 1000 MB \*/

ini\_set("memory\_limit","1000M");

### /\* Reading from the HTML input. \*/

$first = $\_GET["first"]; // word entered by the user.

//$second = $\_GET['second']; // This has been bracketed because we are taking second member automatically.

$tran = $\_GET['tran']; // "Devanagari" - devanagari, "IAST" - IAST, "SLP1" - SLP1 transliteration.

$gender = $\_GET['gender']; // "m" - male. "f" - female. "n" - neuter.

### /\* Defining the variables used in the code and their default values .

\* If there is no change in the execution of subanta.php, the default values are operated.

\*/

$pada = "pratyaya"; // "pada" - pada saJjJA. "pratyaya" - not pada.

$nadi = 0; // 0 - no nadI saJjJA. 1 - nadI saJjJA

$GI = 0; // 0 - no Gyantatva. 1 - Gyantantva.

$Ap = 0; // 0 - no Abantatva. 1 - Abantatva.

$taddhita = 0; // 0 - not taddhita. 1 - taddhita.

$dhatu = 0; // 0 - no dhAtu. 1 - dhAtu.

$eranekaca = 0; // 0 - no application of eranekAco. 1 - application of eranekAco.

$tri = "m"; // for tricaturoH striyAM tisRcatasR (7.2.99). "m" - word tri is pulliGga. "f" - word tri is strIliGga.

$nityastri = 0; // for nitya strIliGga. 0 - not nityastrIliGga. 1 - nityastrIliGga.

$ekajuttarapada = 0; // 0 - no application of ekAjuttarapade NaH. 1 - application of ekAjuttarapade NaH.

$bhashitapumska = 0; // 0 - not bhASitapuMska. 1 - bhASitapuMska.

$anvadesha = 0; // 0 - no anvAdeza. 1 - anvAdeza.

$samasa = 0; // 0 - no samAsa. 1 - samAsa

$pradhana = 0; // 0 - no pradhAnatva in samAsa. 1 - pradhAnatva in samAsa

$Jit = 0; // 0 - no 'J' as it marker. 1 - 'J' as it marker. e.g. ho hanterJNinneSu 7.3.54

$Nit = 0; // 0 - no 'N' as it marker. 1 - 'N' as it marker.

$kvin = 0; // 0 - no kvin pratyaya. 1 - kvin pratyaya.

$kvip = 0; // 0 - no kvip pratyaya. 1 - kvip pratyaya.

$asmadpada = 2; // 0 - niSedha. 1 - nitya. 2 - vibhASA. used to decide whether the conversion of asmad / yuSmad -> me / te, nau / vAm etc will happen or not.

$bhavat = 0; // 0 - bhAterDavatu. 1 - bhU+zatR.

$abhyasta = 0; // 0 - not abhyasta. 1 - abhyasta.

$shatR = 0; // 0 - not zatR pratyaya. 1 - zatR pratyaya.

$Nyanta = 0; // 0 - not Nyanta, 1 - Nyanta.

$san = 0; // 0 - non san, 1 - san. 'san' is used to create nAmadhAtus.

$vasu = 0; // 0 - no vasvanta, 1 - vasvanta.

$shap = 0; // 0 - no zap pratyaya. 1 - zap pratyaya.

$shyan = 0; // 0 - no zyan pratyaya. 1 - zyan pratyaya.

$tRcvat=0; // 0 - not tRjvat. 1 - tRjvat. e.g. tRjvat kroSTuH.

$R = array(); // creating an array where we can store whether the word has 'R' as it marker.

$num = array(); // creating an array where we can store whether the word has 'num' Agama.

$it = array(); // Creating an array where we can store it markers.

$itprakriti = array(); // creating an array where we can store it markers of prakRti.

$itpratyaya = array(); // creating an array where we can store it markers of pratyayas.

$samp = array(); // creating an array where we can store whethere samprasAraNa has happened or not.

// rest of the variables will be defined at their particular occurence in the code.

### /\* a for loop for entering all sup pratyayas one by one. Sambuddhi is at the last after sup. \*/

$sup1= array("su!","O","jas","am","Ow","Sas","wA","ByAm","Bis","Ne","ByAm","Byas","Nasi!","ByAm","Byas","Nas","os","Am","Ni","os","sup","su!","O","jas"); // the last three members are for sambodhana forms.

for ($w=0;$w<count($sup1);$w++) // running the loop till $sup1 is exhausted.

{

$second=$sup1[$w]; // defining the second word as su!, O, jas etc.

### /\* Code for converting from IAST to SLP1 \*/

// defining IAST letters.

$iast = array("a","ā","i","ī","u","ū","ṛ","ṝ","ḷ","ḹ","e","ai","o","au","ṃ","ḥ","kh","ch","ṭh","th","ph","gh","jh","ḍh","dh","bh","ṅ","ñ","ṇ","k","c","ṭ","t","p","g","j","ḍ","d","b","n","m","y","r","l","v","s","h","ś","ṣ",);

// defining SLP1 letters.

$slp = array("a","A","i","I","u","U","f","F","x","X","e","E", "o","O", "M","H","K", "C", "W", "T", "P","G", "J", "Q", "D","B", "N","Y","R","k","c","w","t","p","g","j","q","d","b","n","m","y","r","l","v","s","h","S","z",);

if (preg\_match('/[āĀīĪūŪṛṚṝṜḷḶḹḸṃṂḥḤṭṬḍḌṅṄñÑṇṆśŚṣṢV]/',$first) || preg\_match('/[āĀīĪūŪṛṚṝṜḷḶḹḸṃṂḥḤṭṬḍḌṅṄñÑṇṆśŚṣṢV]/',$second)) // if there is IAST letters in the input, change them to SLP1

{

$first = str\_replace($iast,$slp,$first);

$second = str\_replace($iast,$slp,$second);

}

if ($tran === "IAST") // if the user says that the input is IAST - change it to SLP1.

{

$first = str\_replace($iast,$slp,$first);

$second = str\_replace($iast,$slp,$second);

}

### /\* Code for converting from devanagari - SLP1 \*/

//This is innocuous. Therefore even running without the selection in dropdown menu.

$first = json\_encode($first);

$first = str\_replace("\u200d","",$first); // removing whitespace

$first = str\_replace("\u200c","",$first); // removing whitespace

$first = json\_decode($first);

$second = json\_encode($second);

$second = str\_replace("\u200d","",$second); // removing whitespace

$second = str\_replace("\u200c","",$second); // removing whitespace

$second = json\_decode($second);

$first = convert1($first); // converting to SLP1

$second = convert1($second); // converting to SLP1

$fo = $first; // remembering the original prakRti. Sometimes we need to know what was the original prakRti.

$so = $second; ; // remembering the original pratyayas. Sometimes we need to know what was the original pratyaya.

### /\* displaying the data back to the user \*/

echo "<p class = red >You entered: ".convert($fo)." + ".convert($so)." <a href = subanta.html>Go Back</a></p>";

echo "</br>";

### /\* for sambodhana vibhaktis display \*/

if ($w>20)

{

echo "<p class = red >This is sambuddhi form.</p>";

echo "</br>";

if ($w===21)

{

$sambuddhi=1; // 0 - no sambuddhi. 1 - sambuddhi.

}

else

{

$sambuddhi=0;

}

}

else

{

$sambuddhi=0;

}

### /\* preprocessing for the sup pratyayas. \*/

// Joining the two input words

if ($second === "") // if there is no pratyaya. This doesn't happen in subanta generation. But kept it for other uses.

{

$input = $input = ltrim(chop($first));

}

elseif ($first === "") // if there is no prakRti. This doesn't happen in subanta generation. But kept it for other uses, like sandhi etc.

{

$input = ltrim(chop($second));

}

else // this option is used for subanta generation. $input is 'prakRti'+'pratyaya'.

{

$input = ltrim(chop($first."+".$second));

}

### /\* main coding part starts from here. Based on Siddhantakaumudi text. \*/

### /\* Defining an array $text. \*/

//Here we will store the output after the process of sUtras. The first member is $input.

// The reason behind creating an array and not keeping it a string is - sometimes the sUtras give 2 / more optional outputs. In that case, it is impossible to manage the string.

// Right now what will happen is that 1 member -> 2 members by some sUtra. For next sUtra, we will take these two members one by one and store their results in $text itself.

$text = array();

$text[] = $input; // Defining first member of the array as $input (combined words first and second)

### /\* displaying general information about the sup vibhaktis \*/

### /\* arthavadadhAturapratyayaH prAtipadikam (1.2.45), kRttaddhitasamAsAzca (1.2.46), pratyayaH (3.1.1), parazca (3.1.2), GyAppradipadikAt (4.1.1), svaujasamauTCaSTAbhyAmbhisGebhyAmbhyasGasibhyAmbhyasGasosAmGyossup (4.1.2), vibhaktizca (1.4.104), supaH (1.4.103) \*/

// For future sUtras also, code for each sUtra is separated by /\* xyz.... \*/.

if (in\_array($so,$sup) && $pada==="pratyaya") // Here conditions for application of sUtra is mentioned.

{

echo "<p class = pa >arthavadadhAturapratyayaH prAtipadikam (1.2.45), kRttaddhitasamAsAzca (1.2.46), pratyayaH (3.1.1), parazca (3.1.2), GyAppradipadikAt (4.1.1), svaujasamauTCaSTAbhyAmbhisGebhyAmbhyasGasibhyAmbhyasGasosAmGyossup (4.1.2), vibhaktizca (1.4.104) and supaH (1.4.103) :</p>"; // the class 'pa' is used for sUtras, which are not vidhisUtras.

echo "<p class = pa >अर्थवदधातुरप्रत्ययः प्रातिपदिकम्‌ (१.२.४५), कृत्तद्धितसमासाश्च (१.२.४६), प्रत्ययः (३.१.१), परश्च (३.१.२), ङ्‍याप्प्रातिपदिकात्‌ (४.१.१), स्वौजसमौट्छष्टाभ्याम्भिस्ङेभ्याम्भ्यस्ङसिभ्याम्भ्यस्ङसोसाम्ङ्‍योस्सुप्‌ (४.१.२), विभक्तिश्च (१.४.१०४) तथा सुपः (१.४.१०३) :</p>";

display(0); // see function.php for documentation for display function.

}

### /\* dvyekayordvivacanaikavacane (1.4.22) \*/

if ( (in\_array($so,$eksup)||in\_array($so,$dvisup) ) && $pada==="pratyaya") // For $eksup, $dvisup etc. arrays which are not mentioned in subanta.php, please see function.php.

{

echo "<p class = pa >dvyekayordvivacanaikavacane (1.4.22) :</p>";

echo "<p class = pa >द्व्येकयोर्द्विवचनैकवचने (१.४.२२) :</p>";

display(0);

}

### /\* bahuSu bahuvacanam (1.4.21) \*/

if (in\_array($so,$bahusup) && $pada==="pratyaya")

{

echo "<p class = pa >bahuSu bahuvacanam (1.4.21) :</p>";

echo "<p class = pa >बहुषु बहुवचनम्‌ (१.४.२१) :</p>";

display(0);

}

### /\* sarUpANAmekazeSa ekavibhaktau (1.2.64) \*/

if ( (in\_array($so,$bahusup)||in\_array($so,$dvisup) ) && $pada==="pratyaya")

{

echo "<p class = pa >sarUpANAmekazeSa ekavibhaktau (1.2.64) :</p>";

echo "<p class = pa >सरूपाणामेकशेष एकविभक्तौ (१.२.६४) :</p>";

display(0);

}

### /\* ekavacanaM sambuddhiH (2.3.49) \*/

if ( $sambuddhi === 1 && $so==="su!" && $pada==="pratyaya")

{

echo "<p class = pa >ekavacanaM sambuddhiH (2.3.49) :</p>";

echo "<p class = pa >एकवचनं सम्बुद्धिः (२.३.४९) :</p>";

display(0);

}

### /\* nityabahuvacanAnta special messages \*/

$nityabahuvacana = array("kati","tri","catur","paYcan","saptan","azwan","navan","daSan","ap"); // list of words which are used in bahuvacanam always. New can be added.

if (in\_array($fo,$nityabahuvacana) && !in\_array($so,$bahusup) && in\_array($so,$sup))

{

echo "<p class = red >the word you entered is a nitya bahuvacanAnta word. Please check again.</p>"; // class red is for potential error messages.

echo "<p class = red >आपने जो शब्द दिया है, वह नित्य बहुवचनान्त है । कृपया जाँच कीजिए ।</p>";

display(0);

}

### /\* nityadvivacanAnta special messages \*/

$nityadvivacana = array("dvi"); // list of words which are used in dvivacanam always.

if (in\_array($fo,$nityadvivacana) && !in\_array($so,$dvisup) && in\_array($so,$sup))

{

echo "<p class = red >the word you entered is a nitya dvivacanAnta word. Please check again.</p>";

echo "<p class = red >आपने जो शब्द दिया है, वह नित्य द्विवचनान्त है । कृपया जाँच कीजिए ।</p>";

display(0);

}

### /\* tyadAdi sambodhana special messages \*/

if (in\_array($fo,$tyadadi) && $so==="su!" && $sambuddhi===1)

{

echo "<p class = red >tyadAdi don't have sambodhana.</p>";

echo "<p class = red >त्यदादि का संबोधन नहीं होता है ।</p>";

display(0);

}

### /\* defining sarvanama status \*/

// $sarvafinal. 1 - obligatory. 2 - optional. 0 - no sarvanamasaJjJA.

### /\* vibhASAprakaraNe tIyasya GitsUpasaMkhyAnam (vA 242) \*/

if (in\_array($fo,array("dvitIyA","tftIyA")) && in\_array($so,array("Ne","Nasi!","Nas","Ni")))

{

$sarvafinal = 2;

echo "<p class = sa >By vibhASA dvitIyAtRtIyAbhyAm (7.3.115) :</p>"; // class sa is for vidhisUtras.

echo "<p class = hn >According to siddhAntakaumudI, this sUtra can be done away with. </p>"; // class hn is for some detailed note on the sUtra under question.

echo "<p class = sa >विभाषा द्वितीयातृतीयाभ्याम्‌ (७.३.११५) :</p>";

echo "<p class = hn >कौमुदी के मत में यह सूत्र त्याग करना शक्य है, तीयस्य ङित्सूपसङ्ख्यानात्‌ । </p>";

display(0);

}

elseif (in\_array($fo,array("dvitIya","tftIya")) && in\_array($so,array("Ne","Nasi!","Nas","Ni")))

{

$sarvafinal = 2;

echo "<p class = pa >By vibhASAprakaraNe tIyasya GitsUpasaMkhyAnam (vA 242) :</p>";

echo "<p class = pa >विभाषाप्रकरणे तीयस्य ङित्सूपसङ्ख्यानम्‌ (वा २४२) :</p>";

display(0);

}

elseif ( in\_array($\_GET['cond1\_1\_1'],array("1","2")) || in\_array($\_GET['cond2\_1\_2\_1'],array("1","2"))) // For these $\_GET, pleas see ajax.php and ajax requirements.docx. the numbers refer to their documentation in docx file.

{

$sarvafinal = 0;

echo "<p class = pa >saJjJopasarjanIbhUtAstu na sarvAdayaH (vA 225) :</p>";

echo "<p class = pa >सञ्ज्ञोपसर्जनीभूतास्तु न सर्वादयः (वा २२५) :</p>";

display(0);

}

elseif ( $\_GET['cond1\_1\_1']==="3" || $\_GET['cond2\_1\_2\_1']==="3")

{

$sarvafinal = 0;

echo "<p class = pa >tRtIyAsamAse (1.1.30) :</p>";

echo "<p class = hn >This prevents application of sarvAdIni sarvanAmAni. Therefore, there is no sarvanAmasaJjJA applicable here.</p>";

echo "<p class = pa >तृतीयासमासे (१.१.३०) :</p>";

echo "<p class = hn >अत्र प्रथमं सर्वादीनि सर्वनामानि इति सर्वनामसंज्ञायां प्राप्तायां तृतीयासमासे इति सर्वनामसंज्ञानिषेधः ।</p>";

display(0);

}

elseif ( $\_GET['cond1\_1\_1']==="4" || $\_GET['cond2\_1\_2\_1']==="4")

{

$sarvafinal = 0;

echo "<p class = pa >dvandve ca (1.1.31) :</p>";

echo "<p class = hn >This prohibits application of sarvAdIni sarvanAmAni. </p>";

echo "<p class = pa >द्वन्द्वे च (१.१.३१) :</p>";

echo "<p class = hn >सर्वादीनि सर्वनामानि इति सर्वनामसज्ञायां प्राप्तायाम् अनेन निषेधः ।</p>";

display(0); $dvandveca=1; // 0 - dvandve ca will not apply. 1 - dvandve ca will apply.

}

elseif ( $\_GET['cond1\_1\_1\_5']==="1" || $\_GET['cond2\_1\_2\_1\_5']==="1" )

{

$sarvafinal = 2;

echo "<p class = pa >vibhASA diksamAse bahuvrIhau (1.1.28) :</p>";

echo "<p class = pa >विभाषा दिक्समासे बहुव्रीहौ (१.१.२८) :</p>";

display(0);

}

elseif ( $\_GET['cond1\_1\_1\_5']==="2" || $\_GET['cond2\_1\_2\_1\_5']==="2" )

{

$sarvafinal = 0;

echo "<p class = pa >na bahuvrIhau (1.1.29) :</p>";

echo "<p class = pa >न बहुव्रीहौ (१.१.२९) :</p>";

display(0);

}

elseif ( $\_GET['cond1\_1\_1\_6\_1']==="2" || $\_GET['cond2\_1\_2\_1\_6\_1']==="2" )

{

$sarvafinal = 1;

echo "<p class = pa >pUrvaparAvaradakSiNottarAparAdharANi vyavasthAyAmasaJjJAyAm (ga sU 1) :</p>";

echo "<p class = pa >पूर्वपरावरदक्षिणोत्तरापराधराणि व्यवस्थायामसञ्ज्ञायाम्‌ (ग सू १) :</p>";

display(0);

$purvapara=1; // 0 - pUrvaparA... will not apply. 1 - will apply

$sarvanama=array\_merge($sarvanama,array("pUrva","para","avara","dakziRa","uttara","apara","aDara","pUrvA","parA","avarA","dakziRA","uttarA","aparA","aDarA"));

}

elseif ( $\_GET['cond1\_1\_1\_6\_1']==="1" || $\_GET['cond2\_1\_2\_1\_6\_1']==="1" )

{

$sarvafinal = 0;

}

elseif ( $\_GET['cond1\_1\_1\_6\_2']==="2" || $\_GET['cond2\_1\_2\_1\_6\_2']==="2" )

{

$sarvafinal = 1;

echo "<p class = pa >svamajJAtidhanAkhyAyAm (ga sU 2) :</p>";

echo "<p class = pa >स्वमज्ञातिधनाख्यायाम्‌ (ग सू २) :</p>";

$sva=1; // 0 - svamajJAtidhanAkhyAyAm (1.1.35) doesn't apply. 1 - svamajJAtidhanAkhyAyAm (1.1.35) applies.

display(0);

$sarvanama=array\_merge($sarvanama,array("sva","svA")); // For most of the cases, sva / svA are not sarvanAma. In this particular case, they become sarvanAma.

}

elseif ( $\_GET['cond1\_1\_1\_6\_2']==="1" || $\_GET['cond2\_1\_2\_1\_6\_2']==="1" )

{

$sarvafinal = 0;

$sva=0;

}

elseif ( $\_GET['cond1\_1\_1\_6\_3\_1']==="1" || $\_GET['cond2\_1\_2\_1\_6\_3\_1']==="1" )

{

$sarvafinal = 0;

echo "<p class = pa >'antaraM bahiryogopasaMvyanayoH' (ga sU 3) iti gaNasUtre 'apuri' iti vaktavyam (vA 240) :</p>";

echo "<p class = pa >'अन्तरं बहिर्योगोपसंव्यानयोः' (ग सू ३) इति गणसूत्रे 'अपुरि' इति वक्तव्यम्‌ (वा २४०) :</p>";

$antara=0; // 0 - antaraM bahiryogopasaMvyAnayoH (1.1.36) doesn't apply. 1 - antaraM bahiryogopasaMvyAnayoH (1.1.36) applies.

display(0);

}

elseif ( $\_GET['cond1\_1\_1\_6\_3\_1']==="2" || $\_GET['cond2\_1\_2\_1\_6\_3\_1']==="2" )

{

$sarvafinal = 1;

echo "<p class = pa >antaraM bahiryogopasaMvyAnayoH (ga sU 3) :</p>";

echo "<p class = pa >अन्तरं बहिर्योगोपसंव्यानयोः (ग सू ३) :</p>";

$antara=1;

display(0);

$sarvanama=array\_merge($sarvanama,array("antara","antarA"));

}

elseif ( $\_GET['cond1\_1\_1\_6\_3']==="2" || $\_GET['cond2\_1\_2\_1\_6\_3']==="2" )

{

$sarvafinal = 0;

$antara=0;

}

elseif ( $\_GET['cond1\_1\_1\_6\_4']==="1" || $\_GET['cond2\_1\_2\_1\_6\_4']==="1" )

{

$sarvafinal = 1;

}

elseif ( $\_GET['cond1\_1\_1\_6\_4']==="2" && !in\_array($fo,array("anyatara","anyatarA")) && $\_GET['cond2\_1\_2\_1\_6\_4']==="2")

{

$sarvafinal = 0;

$sarvanama=array\_diff($sarvanama,array("atara","atama","atarA","atamA")); echo "hi";

}

elseif (ends(array($fo),array("anyatama","anyatamA"),1))

{

$sarvafinal = 0;

echo "<p class = pa >As anyatama is not enumerated in sarvAdi and anyatara is specifically enumerated, anyatama doesn't have sarvanAma saJjJA. </p>";

echo "<p class = pa >सर्वादिगण में अन्यतर का गणन हुआ है और अन्यतम का नहीं है, इसलिए अन्यतम की सर्वनाम सञ्ज्ञा नहीं है ।</p>";

display(0); $sarvanama=array\_diff($sarvanama,array("atama","atamA"));

}

elseif ( $\_GET['cond1\_1\_1\_6\_5']==="1" || $\_GET['cond2\_1\_2\_1\_6\_5']==="1" )

{

$sarvafinal = 1;

echo "<p class = pa >samaH sarvaparyAyaH.</p>";

echo "<p class = pa >समः सर्वपर्यायः ।</p>";

display(0);

$sarvanama=array\_merge($sarvanama,array("sama","samA"));

}

elseif ( $\_GET['cond1\_1\_1\_6\_5']==="2" || $\_GET['cond2\_1\_2\_1\_6\_5']==="2" )

{

$sarvafinal = 0;

echo "<p class = pa >tulyaparyAyastu neha gRhyate.</p>";

echo "<p class = pa >तुल्यपर्यायस्तु नेह गृह्यते ।</p>";

display(0);

}

elseif ( $\_GET['cond1\_1\_1']==="6" || $\_GET['cond2\_1\_2\_1']==="6" || in\_array($fo,array("idam","tyad")) )

{

$sarvafinal = 1;

}

elseif ( ($\_GET['cond1\_16']==="2" || $\_GET['cond1\_7']==="2") && in\_array($fo,$sarvanama) )

{

$sarvafinal = 1;

}

elseif ( $\_GET['cond1\_10']==="2" )

{

$sarvafinal = 1;

}

elseif ( ends(array($fo),array("tvat","tyad","tad","yad","etad","idam","adas","yuzmad","asmad","kim","idakam"),1) ) // prAdhAnya and gauNatva pending.

{

$sarvafinal = 1;

}

else

{

$sarvafinal = 0;

}

### /\* sarvAdIni sarvanAmAni (1.1.27) \*/

if ($sarvafinal !==0)

{

if (in\_array($fo,$sarvanama)||in\_array($fo,$sarvanamastri))

{

echo "<p class = pa >sarvAdIni sarvanAmAni (1.1.27) :</p>";

echo "<p class = pa >सर्वादीनि सर्वनामानि (१.१.२७) :</p>";

display(0);

}

/\* tadantasyApIyaM saJjJA | dvandve ca iti jJApakAt \*/

if ( ends(array($fo),$sarvanama,0)||ends(array($fo),$sarvanamastri,0))

{

echo "<p class = pa >sarvAdIni sarvanAmAni (1.1.27) and tadantasyApIyaM saJjJA. dvandve ca iti jJApakAt :</p>";

echo "<p class = pa >सर्वादीनि सर्वनामानि (१.१.२७) तथा तदन्तस्यापीयं सञ्ज्ञा । द्वन्द्वे च इति ज्ञापकात्‌ :</p>";

display(0);

}

}

### /\* defininig eranekAca \*/

if ($\_GET['cond1\_4'] === "1") // for conditions, please see ajax requirements.docx

{

$nadi = 0;

$eranekaca=0;

}

elseif ($\_GET['cond1\_4\_2'] === "1")

{

$eranekaca=0;

$nadi=1;

$GI=1;

}

elseif ($\_GET['cond1\_4\_2'] === "2")

{

$eranekaca=0;

$nadi=1;

$GI=0;

}

elseif ($\_GET['cond1\_4\_3'] === "1")

{

$eranekaca=1;

$nadi=0;

$nI=1;

$dhatu=1;

}

elseif ($\_GET['cond1\_4\_3'] === "2")

{

$eranekaca=1;

$nadi=0;

$dhatu=1;

// anaGNitva, uttvam extra.

}

elseif ($\_GET['cond1\_4\_3'] === "3")

{

$eranekaca=1;

$nadi=0;

$dhatu=1;

// khyatyAtparasya extra.

}

elseif ($\_GET['cond1\_4\_3'] === "4")

{

$eranekaca=1;

$nadi=0;

$dhatu=1;

}

elseif ($\_GET['cond1\_4\_4'] === "1")

{

$eranekaca=1;

$nadi=1;

$GI=1;

$dhatu=1;

}

elseif ($\_GET['cond1\_4\_4'] === "2")

{

$eranekaca=1;

$nadi=1;

$GI=0;

$dhatu=1;

}

elseif ($\_GET['cond1\_4\_5'] === "1")

{

$eranekaca=0;

$nadi=0;

$nI=1;

$dhatu=1;

}

elseif ($\_GET['cond1\_4\_5'] === "2")

{

$eranekaca=0;

$nadi=0;

$nI=0;

$dhatu=1;

$GI=0;

}

elseif ($\_GET['cond1\_4'] === "5")

{

$eranekaca=0;

$nadi=0;

$dhatu=1;

$GI=0;

}

elseif ($\_GET['cond1\_4'] === "6")

{

$eranekaca=0;

$nadi=1;

$dhatu=1;

}

### /\* defining oH supi variable \*/

if ($\_GET['cond1\_5'] === "1")

{

$nadi = 0;

$eranekaca=0;

}

elseif ($\_GET['cond1\_5'] === "2")

{

$eranekaca=0;

$nadi=1;

}

elseif ($\_GET['cond1\_5'] === "3")

{

$eranekaca=1;

$dhatu=1;

}

elseif ($\_GET['cond1\_5'] === "4")

{

$eranekaca=0;

$dhatu=1;

}

### /\* GI definition \*/

if ($\_GET['cond2\_3']==="1")

{

$GI=1;

$nadi=1;

}

elseif ($\_GET['cond2\_3']==="2")

{

$GI=0;

$nadi=1;

}

elseif ($\_GET['cond2\_3']==="3")

{

$GI=1;

$eranekaca=0;

$nadi=1;

}

elseif ($\_GET['cond2\_3']==="4")

{

$GI=0;

$eranekaca=0;

//$nadi=1; // Not sure

$dhatu=1; // for zrI.

}

### /\* makArAnta pulliGga dhAtu definition \*/

if (ends(array($fo),array("m"),1) && $\_GET['cond1\_16']==="1")

{

$dhatu=1;

}

### /\* idam, idakam anvAdeza definition \*/

if ( $\_GET['cond1\_7']==="1" || $\_GET['cond1\_10\_2']==="1" || $\_GET['cond2\_5']==="1" || $\_GET['cond3\_2']==="1" )

{

$sarvafinal=1;

$anvadesha=1;

}

### /\* nakArAnta pulliGga ShanvAat definition \*/

if ( $\_GET['cond1\_8']==="1")

{

$samasa=1;

$pradhana=0;

}

elseif ( $\_GET['cond1\_8']==="2")

{

$samasa=1;

$pradhana=1;

}

### /\* jakArAnta pulliGga kvin pratyaya definition \*/

if ( $\_GET['cond1\_9']==="1" && $\_GET['cond1\_9\_1']==="1" )

{

$dhatu=1;

$kvin=1;

$samasa=0;

$yuj=1; // 0 - yujerasamAse (7.1.71) won't apply. 1 - yujerasamAse (7.1.71) will apply.

}

elseif ( $\_GET['cond1\_9']==="1" || $\_GET['cond1\_9\_1']==="2" )

{

$dhatu=1;

$kvin=1;

$samasa=0;

$yuj=0;

}

else

{

$yuj=0;

}

### /\* tyadAdi gauNatva definition \*/

if ( $\_GET['cond1\_10']==="1" )

{

$noatvasatva=1; // 0 - tyadAdi are upasarjanIbhUta / saJjJA. 1 - tyadAdi are not upasarjanIbhUta / saJjJA i.e. prAdhAnya.

}

else

{

$noatvasatva=0;

}

### /\* asmad / yuSmad -> nau / naH etc definition \*/

if ($\_GET['cond1\_12']==="1" && in\_array($w,array(3,4,5,9,10,11,15,16,17)) )

{

$asmadpada=0;

echo "<p class = pa >Because the words asmad / yuSmad are not after a 'pada' , there won't be tvA/mA, nau/vAm, nas/vas, te/me Adezas:</p>";

echo "<p class = pa >'पदात्‌' नहीं होने से अस्मद्‌ / युष्मद्‌ के त्वा/मा, नौ/वाम्‌, नस्‌/वस्‌, ते/मे आदेश नहीं होंगे ।:</p>";

display(0);

}

elseif($\_GET['cond1\_12']==="2" && in\_array($w,array(3,4,5,9,10,11,15,16,17)) )

{

$asmadpada=0;

echo "<p class = pa >Because the words asmad / yuSmad are used in the start of a 'pAda', there won't be tvA/mA, nau/vAm, nas/vas, te/me Adezas:</p>";

echo "<p class = pa >पाद के प्रारम्भ में प्रयुक्त होने से अस्मद्‌ / युष्मद्‌ के त्वा/मा, नौ/वाम्‌, नस्‌/वस्‌, ते/मे आदेश नहीं होंगे ।:</p>";

display(0);

}

elseif($\_GET['cond1\_12']==="3" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=0;

echo "<p class = pa >By na cavAhAhaivayukte (8.1.24):</p>";

echo "<p class = hn >This prevents tvA/mA, nau/vAm, nas/vas, te/me Adezas.</p>";

echo "<p class = pa >न चवाहाहैवयुक्ते (८.१.२४):</p>";

echo "<p class = hn >इससे त्वा/मा, नौ/वाम्‌, नस्‌/वस्‌, ते/मे आदेश नहीं होंगे ।</p>";

display(0);

}

elseif($\_GET['cond1\_12']==="4" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=0;

echo "<p class = pa >By pazyArthaizcAnAlocane (8.1.25):</p>";

echo "<p class = hn >This prevents tvA/mA, nau/vAm, nas/vas, te/me Adezas.</p>";

echo "<p class = pa >पश्यार्थैश्चानालोचने (८.१.२५):</p>";

echo "<p class = hn >इससे त्वा/मा, नौ/वाम्‌, नस्‌/वस्‌, ते/मे आदेश नहीं होंगे ।</p>";

display(0);

}

elseif($\_GET['cond1\_12\_5\_1\_1\_1\_1']==="1" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=2;

echo "<p class = pa >By vibhASitaM vizeSavacane (8.1.74) :</p>";

echo "<p class = hn >bhASya makes tvA/mA, nau/vAm, nas/vas, te/me Adezas optional by 'bahuvacanamiti vakSyAmi'.</p>";

echo "<p class = pa >विभाषितं विशेषवचने (८.१.७४) :</p.>";

echo "<p class = hn >भाष्य 'बहुवचनमिति वक्ष्यामि' से त्वा/मा, नौ/वाम्‌, नस्‌/वस्‌, ते/मे विभाषा आदेश बताता है ।</p>";

display(0);

}

elseif($\_GET['cond1\_12\_5\_1\_1\_1\_1']==="2" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=1;

echo "<p class = pa >By nAmantrite samAnAdhikaraNe sAmAnyavacanam (8.1.73) :</p>";

echo "<p class = pa >नामन्त्रिते समानाधिकरणे सामान्यवचनम्‌ (८.१.७३) :</p.>";

display(0);

}

elseif($\_GET['cond1\_12\_5\_1\_1\_1']==="2" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=0;

echo "<p class = pa >By AmantritaM pUrvamavidyamAnavat (8.1.72) :</p>";

echo "<p class = hn >By sAmantritam (2.3.48) :</p>";

echo "<p class = pa >आमन्त्रितं पूर्वमविद्यमानवत्‌ (८.१.७२) :</p.>";

echo "<p class = hn >सामन्त्रितम्‌ (२.३.४८) :</p.>";

display(0);

}

elseif($\_GET['cond1\_12\_5\_1\_1']==="2" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=2;

echo "<p class = pa >By sapUrvAyAH prathamAyA vibhASA (8.1.26) :</p>";

echo "<p class = pa >सपूर्वायाः प्रथमाया विभाषा (८.१.२६) :</p.>";

display(0);

}

elseif($\_GET['cond1\_12\_5\_1']==="2" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=1;

echo "<p class = pa >By ete vAMnAvAdaya AdezA ananvAdeze vA vaktavyAH (vA 4717). anvAdeze tu nityaM syuH. :</p>";

echo "<p class = pa >एते वांनावादय आदेशा अनन्वादेशे वा वक्तव्याः (वा ४७१७). अन्वादेशे तु नित्यं स्युः । :</p.>";

display(0);

}

elseif($\_GET['cond1\_12\_5']==="2" && in\_array($w,array(3,4,5,9,10,11,15,16,17)))

{

$asmadpada=2;

echo "<p class = pa >By ete vAMnAvAdaya AdezA ananvAdeze vA vaktavyAH (vA 4717) :</p>";

echo "<p class = pa >एते वांनावादय आदेशा अनन्वादेशे वा वक्तव्याः (वा ४७१७) :</p.>";

display(0);

}

### /\* ancu definitions \*/

if ($\_GET['cond1\_13\_1']==="1" )

{

$kvin=1;

$nance=1; // 0 - nAJceH pUjAyAm (6.4.30) will not apply. 1 - nAJceH pUjAyAm (6.4.30) will apply.

$ancu=1;

}

elseif ($\_GET['cond1\_13\_1']==="2" )

{

$kvin=1;

$nance=0;

$ancu=1; // 0 - no ancu verb. 1 - ancu verb.

}

elseif ($\_GET['cond1\_13']==="2")

{

$kvin=0;

$ancu=0;

}

### /\* defining goanc \*/

if (in\_array($fo,array("goanc","goaYc")) )

{

$kvin=1;

$nance=1;

$goanc=1; // 0 - no goanc word. 1 - the word is goanc.

}

else

{

$goanc=0;

}

### /\* defining Nyanta \*/

if ($\_GET['cond1\_15']==="1")

{

$Nyanta=1;

}

### /\* defining dhatus \*/

// This is the crude way in which dhAtu saJjJA is given without user interface. In practice it is difficult to decide whether the word ends with a dhAtu or not. So made enumeration.

if (sub(array("supis","sutus","suhiMs","gir","pur","sajuz","ASiz"),blank(0),blank(0),0))

{

$dhatu=1;

}

### /\* defining kvip \*/

// It is difficult to know whether kvip pratyay has happened or not, because it is elided completely. so manually enumerated the instances where SK uses kvip pratyayAnta words.

if (sub(array("DvaMs"),blank(0),blank(0),0))

{

$kvip=1;

$dhatu=1;

}

if (sub(array("beBid","cecCid","pipaWiz"),blank(0),blank(0),0))

{

$kvip=1;

}

### /\* strIliGga definitions \*/

### /\* defining Abantatva \*/

if (ends(array($fo),array("A"),1) && $gender==="f" && $\_GET['cond2\_1']==="1")

{

$Ap=0;

$dhatu=1;

}

elseif (ends(array($fo),array("A"),1) && $gender==="f")

{

$Ap=1;

}

else

{

$Ap=0;

}

### /\* nityastrIliGga definition \*/

$strilist = array("strI","SrI"); // list of nityastrIliGga words. Add other words to make it exhaustive.

if (ends(array($fo),$strilist,1))

{

$nityastri=1;

}

### /\* praDI definition \*/

if ($\_GET['cond2\_3\_5']==='1')

{

$dhatu=1;

$eranekaca=2;

$GI=0;

$nadi=2;

echo "<p class = pa >This word is declined like 'lakSmI' according to vRttikAra, and like its musculine counterpart according to kaiyaTa.</p>";

echo "<p class = pa >प्रधीशब्दस्य तु वृत्तिकारानां मते लक्ष्मीवद्रूपम्‌ । कैयटमते तु पुंवद्रूपम्‌ । :</p>";

display(0);

}

if ($\_GET['cond2\_3\_5']==='2')

{

$dhatu=1;

$eranekaca=1;

}

### /\* sulU definition \*/

if (in\_array($fo,array("sulU")) && $gender==="n")

{

$dhatu=1;

$eranekaca=1;

}

### /\* suDI definition \*/

if ($\_GET['cond2\_3\_6']==='1')

{

$dhatu=1;

$GI=0;

$nityastri=1;

echo "<p class = pa >This word is declined like 'zrI' according to vRttikAra, and like its musculine counterpart according to kaiyaTa.</p>";

echo "<p class = pa >सुष्ठु धीर्यस्याः, सुष्ठु ध्यायतीति वेति विग्रहे तु वृत्तिकारमते सुधीः श्रीवत्‍ । मतान्तरे तु पुंवत्‌ ।</p>";

display(0);

}

if ($\_GET['cond2\_3\_5']==='2')

{

$dhatu=1;

$GI=0;

$nityastri=1;

echo "<p class = pa >This word is declined like 'zrI'.</p>";

echo "<p class = pa >'सुष्ठु धीः' इति विग्रहे तु श्रीवदेव ।।</p>";

display(0);

}

### /\* UkArAnta strIliGga - defining \*/

if ($\_GET['cond2\_4']==='1')

{

$nityastri=1;

}

elseif ($\_GET['cond2\_4']==='2')

{

$dhatu=1;

$eranekaca=0;

$nityastri=1;

}

elseif ($\_GET['cond2\_4']==='3')

{

$dhatu=1;

$eranekaca=1;

$nityastri=0;

}

if ($gender==="f" && ends(array($fo),array("varzABU","svayamBU"),1))

{

$dhatu=1;

}

### /\* punarBU definition \*/

if ($\_GET['cond2\_4\_4\_1']==='1')

{

}

elseif ($\_GET['cond2\_4\_4\_1']==='2')

{

$dhatu=1;

$eranekaca=0;

}

### /\* bhASitapuMska definition \*/

if ($\_GET['cond3']==='1')

{

$bhashitapumska=1;

}

### /\* tRtIyAdiSu bhASitapuMskaM puMvadgAlavasya (7.1.74) \*/

if ($bhashitapumska===1 && in\_array($so,$tRtIyAdiSvaci) ) // see function.php for $tRtIyAdiSvaci.

{

echo "<p class = pa >By tRtIyAdiSu bhASitapuMskaM puMvadgAlavasya (7.1.74) :</p>";

echo "<p class = pa >तृतीयादिषु भाषितपुंस्कं पुंवद्गालवस्य (७.१.७४):</p>";

display(0);

}

### /\* defining kRt pratyayas \*/

### /\* kRdatiG (3.1.93) \*/

if ($pada === "pratyaya" && !in\_array($so,$tiG) && $dhatu===1)

{

echo "<p class = pa >By kRdatiG (3.1.93) :</p>";

echo "<p class = pa >कृदतिङ्‌ (३.१.९३) :</p>";

display(0);

}

### /\* Defining pada and bham \*/

### /\* suDanapuMsakasya (1.1.43) \*/

if ($gender !== "n" && in\_array($so,$sarvanamasthana))

{

echo "<p class = pa >By suDanapuMsakasya (1.1.43) :</p>";

echo "<p class = pa >सुडनपुंसकस्य (१.१.४३) :</p>";

display(0);

}

### /\* bahuguNavatuDati saGkhyA (1.1.28) \*/

### /\* Dati ca (1.1.25) \*/

if (in\_array($fo,array("bahu","guRa")))

{

$sankhya = 1; // 0 - no saGkhyA saJjJA. 1 - saGkhyA saJjJA.

echo "<p class = pa >By bahuguNavatuDati saGkhyA (1.1.28) :</p>";

echo "<p class = pa >बहुगुणवतुडति सङ्ख्या (१.१.२८) :</p>";

display(0);

}

else

{

$sankhya = 0;

}

if (in\_array($fo,array("kati")))

{

$sankhya = 1;

$shat = 1; // 0 - no SaT saJjJA. 1 - SaT saJjJA.

echo "<p class = pa >By bahuguNavatuDati saGkhyA (1.1.28) and Dati ca (1.1.25) :</p>";

echo "<p class = pa >बहुगुणवतुडति सङ्ख्या (१.१.२८) तथा डति च (१.१.२५) :</p>";

display(0);

}

else

{

$sankhya = 0;

$shat = 0;

}

### /\* SNAntA Sat (1.1.24) \*/

if (arr($text,'/[zn][+]/') && ($sankhya===1 || ends(array($fo),array("paYcan","zaz","saptan","zwan","navan","daSan"),1)) && ($samasa===0 || $samasa===1 && $pradhana===1) && $\_GET['cond1\_18']!=="2")

// for function arr see function.php.

{

$shat = 1;

echo "<p class = pa >By SNAntA Sat (1.1.24) :</p>";

echo "<p class = pa >ष्णान्ता षट्‌ (१.१.२४) :</p>";

display(0);

}

### /\* defining "SaT" \*/

if ($\_GET['cond1\_18']===2)

{

$shat=0;

}

### /\* same in all lingas - special messages \*/

$samaliGga = array("asmad","asmat","yuzmad","yuzmat"); // list of words which are same in all three genders.

if (in\_array($fo,$samaliGga) || $shat===1)

{

echo "<p class = pa >asmad, yuSmad and words having SaT saJjJA have same forms in all three genders.</p>";

echo "<p class = pa >अस्मद्युष्मद्‍षट्सञ्ज्ञकाः त्रिषु सरूपाः ।</p>";

display(0);

}

### /\* checking for presence of aJcu verb. \*/

if (sub(array("aYc","AYc","anc","Anc"),blank(0),blank(0),0))

{

$ancu=1;

}

else

{

$ancu=0;

}

### /\* zatRvat finder \*/

if(sub(array("pfzad","bfhat","mahat","jagat",),blank(0),blank(0),0) || $\_GET['cond1\_14']==="2" )

{

$shatR = 1;

$it = array\_merge($it,array("S","f")); // adding two it markers 'z' and 'R', because zatR pratyaya has z and R as it marker. Similarly for next cases also.

$itprakriti = array\_merge($itprakriti,array("S","f"));

}

elseif ($\_GET['cond3\_3']==="1" )

{

$shatR = 1;

$it = array\_merge($it,array("S","f"));

$itprakriti = array\_merge($itprakriti,array("S","f"));

$abhyasta=1;

}

elseif ($\_GET['cond3\_3\_2\_1']==="2")

{

$shatR = 1; $dhatu=1;

$it = array\_merge($it,array("S","f"));

$itprakriti = array\_merge($itprakriti,array("S","f"));

$AcCInadyo = 1;

}

elseif ($\_GET['cond3\_3\_2\_1']==="1")

{

$shatR = 1; $dhatu=1;

$it = array\_merge($it,array("S","f"));

$itprakriti = array\_merge($itprakriti,array("S","f"));

$shap = 1;

$shyan = 1;

}

### /\* atvanta finder \*/

if ($\_GET['cond1\_17']==="1" || $\_GET['cond1\_14']==="1" ) // atvanta and ugit.

{

$atu = 1;

$it = array\_merge($it,array("u"));

$itprakriti = array\_merge($itprakriti,array("u"));

}

elseif( in\_array($fo,array("veDas",)) ) // This is atvanta, but not ugit.

{

$atu=1;

}

elseif( in\_array($fo,array("suvas","gras","glas")))

{

$dhatu=1;

$atu=1;

}

else

{

$atu=0;

}

/\*if(sub(array("atu!+"),blank(0),blank(0),0) || ( $bhavat===0 && sub(array("Bavat"),blank(0),blank(0),0)) )

{

$atu = 1;

if (sub(array("atu!+"),blank(0),blank(0),0)|| ( $bhavat===0 && sub(array("Bavat"),blank(0),blank(0),0)))

{

$it = array\_merge($it,array("u"));

$itprakriti = array\_merge($itprakriti,array("u"));

}

}

else

{

$atu = 0;

}\*/

### /\* na SaTsvasrAdibhyaH (4.1.10) \*/

// bracketed because deals with strIpratyayas.

if ($shat===1 || in\_array($fo,$svasrAdi))

{

echo "<p class = sa >na SaTsvasrAdibhyaH (4.1.10) :</p>";

echo "<p class = sa >न षट्स्वस्रादिभ्यः (४.१.१०) :</p>";

display(0); $Ap === 0; $GI === 0; $SaTsvasrAdi=1;

} else {$SaTsvasrAdi=0; }

### /\* striyAM ca (7.1.96) \*/

if ($gender === "f" && sub(array("krozwu"),array("+"),blank(0),0))

{

$text = two(array("krozwu"),array("+"),array("krozwf"),array("+"),0); // for function two - please see function.php.

echo "<p class = sa >striyAM ca (7.1.96) :</p>";

echo "<p class = sa >स्त्रियां च (७.१.९७) :</p>";

display(3);

}

### /\* Rnnebhyo Gip (4.1.5) \*/

if ($gender === "f" && sub(array("f","n"),array("+"),blank(0),0) && $SaTsvasrAdi===0)

{

$text = two(array("f","n"),array("+"),array("f+I","n+I"),array("+"),0);

echo "<p class = sa >By Rnnebhyo Gip (4.1.5) :</p>";

echo "<p class = sa >ऋन्नेभ्यो ङीप्‌ (४.१.५) :</p>";

display(3);

$text = two(array("f+I"),array("+"),array("rI"),array("+"),0);

echo "<p class = sa >By iko yaNaci (6.1.77) :</p>";

echo "<p class = sa >इको यणचि (६.१.७७) :</p>";

display(3);

$GI=1;

$nadi=1;

}

### /\* dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118) \*/

if ($dhatu===1 && in\_array($fo,array("dfnBU","karaBU","kAraBU","punarBU")) && in\_array($so,$sup))

{

$dRnkar=1; // 0 - the word doesn't belong to dRnbhU, karabhU etc. 1 - the word belongs to dRnbhU, karabhU etc.

} else {$dRnkar=0; }

### /\* yU stryAkhyau nadI (1.4.3) and prathamaliGgagrahaNaJca (vA 1036) \*/

// the vArtika is not coded perfectly. Also stryAkhyo is pending.

if ($gender === "f" && !in\_array($fo,array("stri")) && arr($text,'/[iu][+][N]/') && in\_array($so,array("Ne","Nasi!","Nas","Ni")) && $dRnkar===0) // this was commented because it gave nadisaJjJA to priyatri - priyAH tisraH yasya saH.

{

echo "<p class = pa >By Giti hrasvazca (1.4.6) :</p>";

echo "<p class = pa >ङिति ह्रस्वश्च (१.४.६) :</p>";

display(0);

$nadi = 2; // Giti vA.

}

if ($gender === "f" && !in\_array($fo,array("strI")) && arr($text,'/[IU][+][N]/') && $nityastri===1 && in\_array($so,array("Ne","Nasi!","Nas","Ni")) && $dRnkar===0 && $dhatu===1 && ( $eranekaca===0 && anekAca($fo) || ends(array($fo),array("BrU"),1)) ) // this was commented because it gave nadisaJjJA to priyatri - priyAH tisraH yasya saH.

//if ($gender === "f" && !in\_array($fo,array("strI")) && (arr($text,'/[iuIU][+][N]/') && $nityastri===1) && in\_array($so,array("Ne","Nasi!","Nas","Ni")) && $dRnkar===0)

{

echo "<p class = pa >By Giti hrasvazca (1.4.6) :</p>";

echo "<p class = pa >ङिति ह्रस्वश्च (१.४.६) :</p>";

display(0);

$nadi = 2; // Giti vA.

}

elseif ( $gender==="f" && !in\_array($fo,array("stri","strI")) && $dhatu===1 && arr($text,'/[IU][+][A][m]$/') && $nityastri===1 && $dRnkar===0 && $dhatu===1 && ( $eranekaca===0 && anekAca($fo) || ends(array($fo),array("BrU"),1)))

{

echo "<p class = pa >By vA'mi (1.4.5) :</p>";

echo "<p class = pa >वाऽऽमि (१.४.५) :</p>";

display(0);

$nadi = 2;

}

//elseif ( $gender==="f" && !in\_array($fo,array("stri","strI")) && $dhatu===1 && arr($text,'/[IU][+]/') && $dRnkar===0 && $eranekaca!==2 && !in\_array($so,array("su!","ByAm","Bis","sup")))

elseif ( $gender==="f" && !in\_array($fo,array("stri","strI")) && $dhatu===1 && arr($text,'/[IU][+]/') && $dRnkar===0 && $eranekaca!==2 )

{

echo "<p class = pa >By neyaGuvaGsthAnAvastrI (1.4.4) :</p>";

echo "<p class = pa >नेयङुवङ्स्थानावस्त्री (१.४.४) :</p>";

display(0);

$nadi = 0;

}

elseif ( (( arr($text,'/[IU][+]/') && $nityastri===1) || in\_array($fo,array("bahuSreyasI","aticamU"))))

{

if (in\_array($fo,array("bahuSreyasI","aticamU")))

{

echo "<p class = pa >By yU stryAkhyau nadI (1.4.3) and prathamaliGgagrahaNaJca (vA 1036) :</p>";

echo "<p class = pa >यू स्त्र्याख्यौ नदी (१.४.३) तथा प्रथमलिङ्गग्रहणञ्च (वा १०३६) :</p>";

display(0);

}

else

{

echo "<p class = pa >By yU stryAkhyau nadI (1.4.3) :</p>";

echo "<p class = pa >यू स्त्र्याख्यौ नदी (१.४.३) :</p>";

display(0);

}

$nadi = 1;

}

### /\* abhyasta definition \*/

if ($\_GET['cond1\_17\_2']==="1")

{

$abhyasta=1;

$shatR=1;

}

### /\* jakSityAdayaH SaT (6.1.6) \*/

if (sub(array("jakzat","jAgrat","daridrat","SAsat","cakAsat","dIDyat","vevyat"),blank(0),blank(0),0))

{

$abhyasta=1;

$jaksat=1; // 0 - doesn't belong to jakSityAdi. 1 - belongs to jakSityAdi.

echo "<p class = pa >By jakSityAdayaH SaT (6.1.6) :</p>";

echo "<p class = pa >जक्षित्यादयः षट्‍ (६.१.६) :</p>";

display(0);

} else { $jaksat=0; } // when the variables which are not initialised are used, we keep this else box. Otherwise for future uses, $jaksat will be null and PHP will send notices to browser.

### /\* ubhe abhyastam (6.1.5) \*/

if ($abhyasta===1 && $jaksat===0)

{

echo "<p class = pa >By ubhe abhyastam (6.1.5) :</p>";

echo "<p class = pa >उभे अभ्यस्तम्‌ (६.१.५) :</p>";

display(0);

}

### /\* tyadAdiSu dRSo'nAlocane kaJca (3.2.60) \*/

if ( sub($tyadadi,array("dfS"),blank(0),0) )

{

echo "<p class = pa >By tyadAdiSu dRSo'nAlocane kaJca (3.2.60) :</p>";

echo "<p class = pa >त्यदादिषु दृशोऽनालोचने कञ्च (३.२.६०) :</p>";

display(0); $kvin=1;

}

### /\* kvin pratyaya from asRj \*/

if ( $fo==="asfj" && in\_array($so,$sup) )

{

echo "<p class = pa >asRj has kutva at padAnta, because of mandate of kvin pratyaya after sRj. </p>";

echo "<p class = pa >असृजः पदान्ते कुत्वम्‌, सृजेः क्विनो विधानात्‌ ।</p>";

display(0); $kvin=1;

}

### /\* no kvin pratyaya from viSvasRj \*/

if ( $fo==="viSvasfj" && in\_array($so,$sup) )

{

echo "<p class = pa >'rajjusRDbhyAm' usage in bhASya under 'sRjidRSoH' sUtra mandates that there will be no kutva here.</p>";

echo "<p class = pa >विश्वसृड्‌ इत्यादौ तु कुत्वं न । 'सृजिदृशोः' इति सूत्रे 'रज्जुसृड्भ्याम्‌' इति भाष्यप्रयोगात्‌ ।</p>";

display(0); $kvin=0;

}

### /\* spRSo'nudake kvin (3.2.58) \*/

if ( sub(array("spfS"),array("+"),blank(0),0) )

{

echo "<p class = pa >By spRSo'nudake kvin (3.2.58) :</p>";

echo "<p class = pa >स्पृशोऽनुदके क्विन्‌ (३.२.५८) :</p>";

display(0); $kvin=1;

}

### /\* Defining $vasu \*/

if ( sub(array("vidvas","sedivas","uzivas","Suzruvas","upeyivas","anASvas"),array("+"),blank(0),0) )

{

$text = two(array("vidvas","sedivas","uzivas","Suzruvas","upeyivas","anASvas"),array("+"),array("vidvasu!","sedivasu!","uzivasu!","Suzruvasu!","upeyivasu!","anASvasu!"),array("+"),0);

echo "<p class = pa >This is a vasvanta word.</p>";

echo "<p class = pa >यह एक वस्वन्त शब्द है ।</p>";

display(0); $vasu=1;

}

### /\* Defining ugit \*/

if ( sub(array("DvaMs","sraMs"),array("+"),blank(0),0) )

{

$text = two(array("DvaMs","sraMs"),array("+"),array("DvaMsu!","sraMsu!"),array("+"),0);

echo "<p class = pa >This is an udit word.</p>";

echo "<p class = pa >यह एक उदित्‌ शब्द है ।</p>";

display(0);

}

### /\* dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118) \*/

if ($dhatu===1 && in\_array($fo,array("dfnBU","karaBU","kAraBU","punarBU")) && (in\_array($so,array("Ne","Nasi!","Nas","Ni")) || ($sambuddhi===1 && $so==="su!") ) && arr($text,'/[U][+]['.pc('ac').']/'))

{

echo "<p class = pa >By dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118), yaN bars application of iyaG,uvaG. Therefore, nadIkAryas will happen. :</p>";

$nadi=1;

if (in\_array($fo,array("karaBU","kAraBU")))

{

echo "<p class = hn >There is pAThabheda here. Some hold that there is dIrgha kAra word here. SK has adopted both the readings, therefore we have kept them optional.</p>";

$nadi=2;

}

echo "<p class = pa >दृन्करपुनःपूर्वस्य भुवो यण्‌ वक्तव्यः (वा ४११८) द्वारा प्राप्त यण्‌ - इयङ्‌/उवङ्‌ का बाधन करता है । अतः नदीकार्य होंगे ।</p>";

if (in\_array($fo,array("karaBU","kAraBU")))

{

echo "<p class = hn >दीर्घपाठे करपूर्वस्य उवङेव । ह्रस्वपाठे करपूर्वस्य यणेव इति विवेकः ।</p>";

}

display(0);

}

### /\* adDDatarAdibhyaH paJcabhyaH (7.1.25) and ekatarAtpratiSedho vaktavyaH (vA 4287) \*/

elseif (sub(array("ekatara"),array("+"),array("su!","am"),0) && $gender==="n")

{

echo "<p class = pa >ekatarAtpratiSedho vaktavyaH (vA 4287) :</p>";

echo "<p class = pa >एकतरात्प्रतिषेधो वक्तव्यः (वा ४२८७) :</p>";

display(0); $ekatara=1; // 0 - the word is not ekatara. 1 - the word is ekatara. This will be useful in adDDatarAdibhyaH paJcabhyaH (7.1.25).

} else {$ekatara=0; }

if (sub(array("atara","atama","anya","anyatara","itara"),array("+"),array("su!","am"),0) && $gender==="n" && $ekatara===0)

{

$text = two(array("a+",),array("su!","am"),array("a+"),array("adq","adq"),0);

echo "<p class = sa >By adDDatarAdibhyaH paJcabhyaH (7.1.25) :</p>";

echo "<p class = sa >अद्ड्डतरादिभ्यः पञ्चभ्यः (७.१.२५) :</p>";

display(3);

$Dit = 1; // 0 - 'D' is not it. 1 - 'D' is it.

$adD=1; // 0 - adD suffix has not been applied. 1 - adD suffix is applied.

} else {$adD = 0; $Dit =0;}

### /\* maghavA bahulam (6.4.128) \*/

if (sub(array("maGavan"),array("+"),blank(0),0) && in\_array($so,$sup))

{

$text = two(array("maGavan"),array("+"),array("maGavatf!"),array("+"),1);

echo "<p class = sa >By maghavA bahulam (6.4.128) :</p>";

echo "<p class = sa >मघवा बहुलम्‌ (६.४.१२८) :</p>";

display(3);

$it = array\_merge($it,array("f"));

$itprakriti = array\_merge($itprakriti,array("f"));

}

### /\* arvaNastrasAvanaJa (6.4.127) \*/

if (sub(array("arvan"),array("+"),blank(0),0) && in\_array($so,$sup) && $fo!=="anarvan" && $so!=="su!")

{

$text = two(array("arvan"),array("+"),array("arvatf!"),array("+"),0);

echo "<p class = sa >By arvaNastrasAvanaJa (6.4.127) :</p>";

echo "<p class = sa >अर्वणस्त्रसावनञः (६.४.१२७) :</p>";

display(3);

$it = array\_merge($it,array("f"));

$itprakriti = array\_merge($itprakriti,array("f"));

}

if (sub(array("arvan"),array("+"),blank(0),0) && in\_array($so,$sup) && ($fo=="anarvan" || $so!=="su!"))

{

echo "<p class = pa >'tR' Adeza of arvaNastrasAvanaJa (6.4.127) doesnt apply here. </p>";

echo "<p class = pa >अर्वणस्त्रसावनञः (६.४.१२७) का 'तृ' आदेश यहाँ नहीं होता है ।</p>";

display(0);

}

### /\* patch for dRz \*/

if (sub(array("dfS"),array("+"),blank(0),0))

{

$kvip=1;

$kvin=1;

}

### /\* RtvigdadhRksragdiguSNigaJcuyujikruJcAM ca (3.2.59) \*/

if (sub(array("ftvij","daDfz","sraj","diS","zRih","aYcu","yuj","kruYc","anc","Anc","aYc","AYc","krunc"),array("+"),blank(0),0))

{

$kvin=1;

}

if (sub(array("ftvij","daDfz","sraj","diS","zRih","aYcu","yuj","kruYc","anc","Anc","aYc","AYc","krunc"),array("+"),blank(0),0) && $kvin===1 )

{

echo "<p class = sa >By RtvigdadhRksragdiguSNigaJcuyujikruJcAM ca (3.2.59) :</p>";

echo "<p class = sa >ऋत्विग्दधृक्स्रग्दिगुष्णिगञ्चुयुजिक्रुञ्चां च (३.२.५९) :</p>";

display(3);

}

### /\* rAyo hali (7.2.85) \*/

if (sub(array("rE"),array("+"),$hlsup,0) && in\_array($so,$hlsup) && !($gender==="n" && $so==="su!") )

{

$text = two(array("rE"),array("+"),array("rA"),array("+"),0);

echo "<p class = sa >By rAyo hali (7.2.85) :</p>";

echo "<p class = sa >रायो हलि (७.२.८५) :</p>";

display(3);

$rayo=1; // 0 - rAyo hali has not applied. 1 - rAyo hali has applied. Useful in hrasvo napuMsake prAtipadikasya (1.2.47).

} else {$rayo = 0; }

### /\* hrasvo napuMsake prAtipadikasya (1.2.47) \*/

$achrasva= array("a","a","i","i","u","u","f","f","x","x","i","u","i","u",);

if (sub($ac,array("+"),blank(0),0) && $gender==="n" && in\_array($so,$sup) && $rayo===0)

{

if (sub(array("e","o","E","O"),array("+"),blank(0),0))

{

$text = two($ac,array("+"),$achrasva,array("+"),0);

echo "<p class = sa >By hrasvo napuMsake prAtipadikasya (1.2.47) :</p>";

echo "<p class = pa >By eca igghrasvAdeze (1.1.47) :</p>";

echo "<p class = sa >ह्रस्वो नपुंसके प्रातिपदिकस्य (१.२.४७) :</p>";

echo "<p class = pa >एच इग्घ्रस्वादेशे (१.२.४७) :</p>";

display(0);

echo "<p class = pa >This word is not bhASitapuMska. </p>";

echo "<p class = pa >यह शब्द भाषितपुंस्क नहीं है ।</p>";

display(0);

}

elseif($bhashitapumska===1 && in\_array($so,$tRtIyAdiSvaci))

{

$text = two($ac,array("+"),$achrasva,array("+"),1);

echo "<p class = sa >By hrasvo napuMsake prAtipadikasya (1.2.47) :</p>";

echo "<p class = sa >ह्रस्वो नपुंसके प्रातिपदिकस्य (१.२.४७) :</p>";

display(0);

}

else

{

$text = two($ac,array("+"),$achrasva,array("+"),0);

echo "<p class = sa >By hrasvo napuMsake prAtipadikasya (1.2.47) :</p>";

echo "<p class = sa >ह्रस्वो नपुंसके प्रातिपदिकस्य (१.२.४७) :</p>";

display(0);

}

}

### /\* ato'm (7.1.24) \*/

if (sub(array("a"),array("+"),array("su!","am"),0) && $gender==="n" && $adD ===0)

{

$text = two(array("a+",),array("su!","am"),array("a+"),array("am","am"),0);

echo "<p class = sa >By ato'm (7.1.24) :</p>";

echo "<p class = sa >अतोऽम्‌ (७.१.२४) :</p>";

display(3); $atom=1;

} else { $atom =0; }

### /\* defining whether the first word is asmad / yuzmad \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,$sup))

{

$asmad=1; // 0 - the word doesn't end with asmad / yuSmad. 1 - the word ends with asmad / yuSmad.

}

else

{

$asmad=0;

}

### /\* Whole replacements for asmad / yuSmad \*/

### /\* yuSmadasmadoH SaSThIcaturthIdvitIyAsthayorvAMnAvau (8.1.20) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($w,array(4,10,16)) && $asmadpada>0 )

{

if($asmadpada===2)

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("Ow","ByAm","os"),array("inO","nO","vAm"),array("","",""),1);

}

else

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("Ow","ByAm","os"),array("inO","nO","vAm"),array("","",""),0);

}

echo "<p class = sa >By yuSmadasmadoH SaSThIcaturthIdvitIyAsthayorvAMnAvau (8.1.20) :</p>";

echo "<p class = sa >युष्मदस्मदोः षष्ठीचतुर्थीद्वितीयास्थयोर्वांनावौ (८.१.२०) :</p>";

display(7);

}

### /\* bahuvacanasya vasnasau (8.1.21) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && (in\_array($so,array("Sas","Am")) || (in\_array($so,array("Byas")) && $w===11)) && $asmadpada>0 )

{

if ($asmadpada===2)

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("Sas","Byas","Am"),array("inas","nas","vas"),array("","",""),1);

}

else

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("Sas","Byas","Am"),array("inas","nas","vas"),array("","",""),0);

}

echo "<p class = sa >By bahuvacanasya vasnasau (8.1.21) :</p>";

echo "<p class = sa >बहुवचनस्य वस्नसौ (८.१.२१) :</p>";

display(7);

}

### /\* temayAvekavacanasya (8.1.22) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("Ne","Nas")) && $asmadpada>0 )

{

if ($asmadpada===2)

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("Ne","Nas"),array("ime","me","te"),array("",""),1);

}

else

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("Ne","Nas"),array("ime","me","te"),array("",""),0);

}

echo "<p class = sa >By temayAvekavacanasya (8.1.22) :</p>";

echo "<p class = sa >तेमयावेकवचनस्य (८.१.२२) :</p>";

display(7);

}

### /\* tvAmau dvitIyAyAH (8.1.23) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("am")) && $asmadpada>0 )

{

if ($asmadpada===2)

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("am"),array("imA","mA","tvA"),array(""),1);

}

else

{

$text = two(array("yasmad+","asmad+","yuzmad+"),array("am"),array("imA","mA","tvA"),array(""),0);

}

echo "<p class = sa >By tvAmau dvitIyAyAH (8.1.23) :</p>";

echo "<p class = sa >त्वामौ द्वितीयायाः (८.१.२३) :</p>";

display(7);

}

### /\* pratyaya Adeza for yuSmad / asmad \*/

### /\* yuSmadasmadbhyAM Gaso'z (7.1.27) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("Nas")))

{

$text = two(array("+"),array("Nas"),array("+"),array("a"),0);

echo "<p class = sa >By yuSmadasmadbhyAM Gaso'z (7.1.27) :</p>";

echo "<p class = sa >युष्मदस्मद्भ्यां ङसोऽश्‍ (७.१.२७) :</p>";

display(3);

}

### /\* Geprathamayoram (7.1.28) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("Ne","su!","Ow","O","jas","am")))

{

$text = two(array("+"),array("Ne","su!","Ow","O","jas","am"),array("+"),array("am","am","am","am","am","am"),0);

echo "<p class = sa >By Geprathamayoram (7.1.28) :</p>";

echo "<p class = sa >ङेप्रथमयोरम्‌ (७.१.२८) :</p>";

display(3);

}

### /\* zaso na (7.1.29) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("Sas")))

{

$text = two(array("+"),array("Sas"),array("+"),array("ns"),0);

echo "<p class = sa >By zaso na (7.1.29) :</p>";

echo "<p class = hn >This sUtra prevents application of Geprathamayoram. Also the Adeza is of the first letter by AdeH parasya.</p>";

echo "<p class = sa >शसो न (७.१.२९) :</p>";

echo "<p class = hn >यह सूत्र ङेप्रथमयोरम्‌ का बाध करता है । आदेश आदेः परस्य सूत्र से अकार का है ।</p>";

display(3);

}

### /\* bhyaso bhyam (7.1.30) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("Byas")) && $w===11)

{

$text = two(array("+"),array("Byas"),array("+"),array("aByam"),0);

echo "<p class = sa >By bhyaso bhyam (7.1.30) :</p>";

echo "<p class = sa >भ्यसो भ्यम्‌ (७.१.३०) :</p>";

display(3);

}

### /\* paJcamyA at (7.1.32) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("Byas")) && $w===14)

{

$text = two(array("+"),array("Byas"),array("+"),array("at"),0);

echo "<p class = sa >By paJcamyA at (7.1.32) :</p>";

echo "<p class = sa >पञ्चम्या अत्‌ (७.१.३२) :</p>";

display(3);

}

### /\* ekavacanasya ca (7.1.32) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("Nasi!")))

{

$text = two(array("+"),array("Nasi!"),array("+"),array("at"),0);

echo "<p class = sa >By ekavacanasya ca (7.1.32) :</p>";

echo "<p class = sa >एकवचनस्य च (७.१.३२) :</p>";

display(3);

}

### /\* prakRti Adezas for asmad / yuSmad \*/

### /\* yo'ci (7.2.89) \*/

if (sub(array("asmad","yuzmad",),array("+"),blank(0),0) && in\_array($so,array("wA","Ni","os")) )

{

$text = two(array("asmad","yuzmad",),array("+"),array("asmay","yuzmay",),array("+"),0);

echo "<p class = sa >By yo'ci (7.2.89) :</p>";

echo "<p class = sa >योऽचि (७.२.८९) :</p>";

display(3);

}

### /\* yuSmadasmadoranAdeze (7.2.86) \*/

if (sub(array("asmad","yuzmad"),array("+"),$tRtIyAdi,0) && in\_array($so,array("ByAm","Bis","sup")) )

{

$text = two(array("asmad","yuzmad",),array("+"),array("asmaA","yuzmaA"),array("+"),0);

echo "<p class = sa >By yuSmadasmadoranAdeze (7.2.86) :</p>";

echo "<p class = sa >युष्मदस्मदोरनादेशे (७.२.८६) :</p>";

display(3);

}

### /\* dvitIyAyAJca (7.2.87) \*/

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("am","Ow","Sas")))

{

$text = two(array("asmad","yuzmad",),array("+"),array("asmaA","yuzmaA"),array("+"),0);

echo "<p class = sa >By dvitIyAyAJca (7.2.87) :</p>";

echo "<p class = pa >maparyantasya (7.2.91) :</p>";

echo "<p class = sa >द्वितीयायाञ्च (७.२.८७) :</p>";

display(3);

}

### /\* zeSe lopaH (7.2.90) \*/

// strIliGga of asmad and yuSmad still pending to be clarified.

if (sub(array("asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("su!","jas","Ne","Byas","Nasi!","Nas","Am")))

{

$text = two(array("asmad","yuzmad"),array("+"),array("asma","yuzma"),array("+"),0);

echo "<p class = sa >By zeSe lopaH (7.2.90) :</p>";

echo "<p class = sa >शेषे लोपः (७.२.९०) :</p>";

display(3);

}

### /\* maparyantasya Adezas for asmad / yuSmad \*/

### /\* yUyavayau jasi (7.2.93) \*/

if (sub(array("asm","yuzm"),array(""),blank(0),0) && $asmad===1 && in\_array($so,array("jas")))

{

$text = one(array("yasm","asm","yuzm"),array("ivaya","vaya","yUya"),0); // for function one please see function.php.

echo "<p class = sa >By yUyavayau jasi (7.2.93) :</p>";

echo "<p class = pa >maparyantasya (7.2.91) :</p>";

echo "<p class = sa >यूयवयौ जसि (७.२.९३) :</p>";

echo "<p class = pa >मपर्यन्तस्य (७.२.९१) :</p>";

display(3);

}

### /\* tvAhau sau (7.2.94) \*/

if (sub(array("asm","yuzm"),array(""),blank(0),0) && $asmad===1 && in\_array($so,array("su!")))

{

$text = one(array("asm","yuzm"),array("aha","tva"),0);

echo "<p class = sa >By tvAhau sau (7.2.94) :</p>";

echo "<p class = pa >maparyantasya (7.2.91) :</p>";

echo "<p class = sa >त्वाहौ सौ (७.२.९४) :</p>";

echo "<p class = pa >मपर्यन्तस्य (७.२.९१) :</p>";

display(3);

}

### /\* tubhyamahyau Gayi (7.2.95) \*/

if (sub(array("asm","yuzm"),array(""),blank(0),0) && $asmad===1 && in\_array($so,array("Ne")))

{

$text = one(array("yasm","asm","yuzm"),array("imahya","mahya","tuBya"),0);

echo "<p class = sa >By tubhyamahyau Gayi (7.2.95) :</p>";

echo "<p class = pa >maparyantasya (7.2.91) :</p>";

echo "<p class = sa >तुभ्यमह्यौ ङयि (७.२०९५) :</p>";

echo "<p class = pa >मपर्यन्तस्य (७.२.९१) :</p>";

display(3);

}

### /\* tavamamau Gasi (7.2.96) \*/

if (sub(array("asm","yuzm"),array(""),blank(0),0) && $asmad===1 && in\_array($so,array("Nas")))

{

$text = one(array("yasm","asm","yuzm"),array("imama","mama","tava"),0);

echo "<p class = sa >By tavamamau Gasi (7.2.96) :</p>";

echo "<p class = pa >maparyantasya (7.2.91) :</p>";

echo "<p class = sa >तवममौ ङसि (७.२.९६) :</p>";

echo "<p class = pa >मपर्यन्तस्य (७.२.९१) :</p>";

display(3);

}

### /\* tvamAvekavacane (7.2.97) \*/

if (sub(array("asm","yuzm"),array(""),blank(0),0) && $asmad===1 && (in\_array($so,array("am","wA","Nasi!","Ni")) && (in\_array($fo,array("asmad","yuzmad")) ) || $\_GET['cond1\_11']==="1") )

{

$text = one(array("yasm","yuzm","asm"),array("ima","tva","ma"),0);

echo "<p class = sa >By tvamAvekavacane (7.2.97) :</p>";

echo "<p class = pa >maparyantasya (7.2.91) :</p>";

echo "<p class = sa >त्वमावेकवचने (७.२.९७) :</p>";

echo "<p class = pa >मपर्यन्तस्य (७.२.९१) :</p>";

display(3);

}

### /\* yuvAvau dvivacane (7.2.92) \*/

if (sub(array("asm","yuzm"),array(""),blank(0),0) && $asmad===1 && ((in\_array($so,$dvisup) && in\_array($fo,array("asmad","yuzmad"))) || $\_GET['cond1\_11']==="2") )

{

$text = one(array("asm","yuzm"),array("Ava","yuva"),0);

echo "<p class = sa >By yuvAvau dvivacane (7.2.92) :</p>";

echo "<p class = pa >maparyantasya (7.2.91) :</p>";

echo "<p class = sa >युवावौ द्विवचने (७.२.९२) :</p>";

echo "<p class = pa >मपर्यन्तस्य (७.२.९१) :</p>";

display(3);

}

### /\* ato guNe patch \*/

// Because ato guNe sometimes interferes with akaH savarNe dIrghaH, it is treated specially wherever it is applicable.

if ($asmad===1 && sub(array("aa","a+aByam","a+at","tava+a","mama+a","Ava+aByam","yuva+aByam","Ava+at","yuva+at"),blank(0),blank(0),0))

{

$text = one(array("aa",),array("a"),0);

$text = two(array("asma+aByam","yuzma+aByam","ma+aByam","tva+aByam","Ava+aByam","yuva+aByam"),array(""),array("asm+aByam","yuzm+aByam","m+aByam","tv+aByam","Av+aByam","yuv+aByam"),array(""),0);

$text = two(array("asma+at","yuzma+at","tva+at","ma+at","tava+a","mama+a","Ava+at","yuva+at"),array(""),array("asm+at","yuzm+at","tvat","mat","tava","mama","Av+at","yuv+at"),array(""),0);

echo "<p class = sa >By ato guNe (6.1.96) :</p>";

echo "<p class = sa >अतो गुणे (६.१.९६) :</p>";

display(3);

}

### /\* prathamAyAzca dvivacane bhASAyAm (7.2.88) \*/

if (sub(array("Avad","yuvad","imad","itvad","asmad","yuzmad"),array("+"),blank(0),0) && in\_array($so,array("O")) && $asmad===1)

{

$text = two(array("Avad","yuvad","imad","itvad","asmad","yuzmad"),array("+"),array("AvaA","yuvaA","imaA","itvaA","asmaA","yuzmaA"),array("+"),0);

echo "<p class = sa >By prathamAyAzca dvivacane bhASAyAm (7.2.88) :</p>";

echo "<p class = sa >प्रथमायाश्च द्विवचने भाषायाम्‌ (७.२.९१) :</p>";

display(3);

}

### /\* aGgakArye kRte punarnAGgakAryam (pa 93) \*/

if (sub(array("vaya","yUya"),array("+"),array("am"),0) && $so==="jas" )

{

echo "<p class = pa >aGgakArye kRte punarnAGgakAryam (pa 93) prevents application of jasaH zI.</p>";

echo "<p class = pa >अङ्गकार्ये कृते पुनर्नाङ्गकार्यम्‌ (प ९३) से जसः शी का प्रतिषेध होता है ।</p>";

display(0); $nojas=1;

} else { $nojas=0; }

### /\* idamo maH (7.2.108) \*/

if (sub(array("idam","idakam"),array("+"),blank(0),0) && $so==="su!")

{

echo "<p class = sa >By idamo maH (7.2.108) :</p>";

echo "<p class = sa >इदमो मः (७.२.१०८) :</p>";

display(3);

### /\* ido'y puMsi (7.2.111) \*/

if (sub(array("idam","idakam"),array("+"),blank(0),0) && $so==="su!" && $gender==="m")

{

$text = two(array("idam","idakam"),array("+"),array("ayam","ayakam"),array("+"),0);

echo "<p class = sa >By ido'y puMsi (7.2.111) :</p>";

echo "<p class = sa >इदोऽय्‌ पुंसि (७.२.१११) :</p>";

display(3);

}

$idamoma=1; // 0 - idamo maH has not applied. 1 - idamo maH has applied. useful in tyadAdInAmaH.

} else { $idamoma=0; }

### /\* yaH sau (7.2.110) \*/

if (sub(array("idam","idakam"),array("+"),blank(0),0) && $so==="su!" && $gender==="f")

{

$text = two(array("idam"),array("+"),array("iyam"),array("+"),0);

echo "<p class = sa >By yaH sau (7.2.110) :</p>";

echo "<p class = sa >यः सौ (७.२.११०) :</p>";

display(3);

}

### /\* anvAdeze napuMsake enadvaktavyaH (vA 1569) \*/

if ($gender==="n" && sub(array("idam+","etad+","idakam+",),blank(0),blank(0),0) && in\_array($fo,array("idam","etad","idakam")) && $anvadesha===1 && in\_array($so,array("am")))

{

$text = one(array("idam+","etad+","idakam+",),array("enad+","enad+","enad",),0);

echo "<p class = sa >By anvAdeze napuMsake enadvaktavyaH (vA 1569) :</p>";

echo "<p class = sa >अन्वादेशे नपुंसके एनद्वक्तव्यः (वा १५६९) :</p>";

display(0);

}

### /\* svamornapuMsakAt (7.1.23) \*/

if ( $gender==="n" && in\_array($so,array("su!","am")) && $atom===0 && $adD ===0 )

{

$text = two(array("+"),array("su!","am"),array("+"),array("",""),0);

echo "<p class = sa >By svamornapuMsakAt (7.1.23) :</p>";

echo "<p class = sa >स्वमोर्नपुंसकात्‌ (७.१.२३) :</p>";

display(3);

$svamo = 1; // 0 - su / am have not been elided. 1 - su / am have been elided. Useful in many places like na lumatAGgasya etc.

$pada="pada"; // when these su / am are elided, the word gets pada saJjJA, because there is no pratyaya remaining after that.

} else { $svamo = 0; }

### /\* kimaH kaH (7.2.103) \*/

if (sub(array("kim","kaka"),array("+"),blank(0),0) && !arr($text,'/[k][i][m][+]$/') && in\_array($so,$sup))

{

$text = two(array("kim","kaka"),array("+"),array("ka","ka"),array("+"),0);

echo "<p class = sa >By kimaH kaH (7.2.103) :</p>";

echo "<p class = sa >किमः कः (७.२.१०३) :</p>";

display(3); $sarvafinal=1;

}

if (sub(array("kim","kaka"),array("+"),blank(0),0) && arr($text,'/[k][i][m][+]$/') && in\_array($so,$sup))

{

echo "<p class = pa >na lumatA'Ggasya prevents application of kimaH kaH (7.2.103) :</p>";

echo "<p class = pa >न लुमताऽङ्गस्य से किमः कः (७.२.१०३) का बाध होता है ।</p>";

display(0);

}

### /\* Rnnebhyo GIp (4.1.5) \*/

// Pending, because it deals with two pratyayas. Right now, if user wants he can write the Gyanta itself and check.

### /\* aSTana A vibhaktau (7.2.84) \*/

if ( sub(array("zwan"),array("su!"),blank(0),0) && $\_GET['cond1\_8']==="1")

{

$text = two(array("zwan"),array("su!"),array("zwaA"),array("su!"),1);

echo "<p class = sa >By aSTana A vibhaktau (7.2.84) :</p>";

echo "<p class = hn >Atva of aSTana A vibhaktau is optional, as shown by 'aSTano dIrghAt'.</p>";

echo "<p class = sa >अष्टन आ विभक्तौ (७.२.८४)</p>";

echo "<p class = hn >अष्टन आ विभक्तौ से हुआ आत्व वैकल्पिक है । अष्टनो दीर्घात्‌ इस सूत्र में दीर्घग्रहण के ज्ञापक से । </p>";

display(3);

$astana=1; // 0 - this sUtra has not applied. 1 - aSTana A vibhaktau has applied. useful in na lopaH prAtipadikAntasya.

}

elseif ( sub(array("zwan"),$hlsup,blank(0),0) || (sub(array("zwan"),array("jas","Sas"),blank(0),0) && $\_GET['cond1\_8']==="2") )

{

$text = two(array("zwan"),$hlsup,array("zwaA"),$hlsup,1);

$text = two(array("zwan"),array("jas","Sas"),array("zwaA"),array("jas","Sas"),1);

echo "<p class = sa >By aSTana A vibhaktau (7.2.84) :</p>";

echo "<p class = hn >Atva of aSTana A vibhaktau is optional, as shown by 'aSTano dIrghAt'.</p>";

echo "<p class = sa >अष्टन आ विभक्तौ (७.२.८४)</p>";

echo "<p class = hn >अष्टन आ विभक्तौ से हुआ आत्व वैकल्पिक है । अष्टनो दीर्घात्‌ इस सूत्र में दीर्घग्रहण के ज्ञापक से । </p>";

display(3);

$astana=1;

} else { $astana=0; }

### /\* aSTAbhya auz (7.1.21) \*/

if ( sub(array("zwaA"),array("jas","Sas"),blank(0),0) && ( ($samasa ===1 && $pradhana===1) || $samasa===0 ))

{

$text = two(array("zwaA"),array("jas","Sas"),array("zwaA"),array("O","O"),0);

echo "<p class = sa >By aSTAbhya auz (7.1.21) :</p>";

echo "<p class = sa >अष्टाभ्य औश्‌ (७.१.२१) :</p>";

display(3);

}

### /\* pratyayasya lukzlulupaH (1.1.61) \*/

### /\* SaDbhyo luk (7.1.22) \*/

if ( $shat === 1 && in\_array($so,array("jas","Sas")) && ( ($samasa ===1 && $pradhana===1) || $samasa===0 ))

{

$text = two(array("+"),array("jas","Sas"),array("+"),blank(2),0);

echo "<p class = sa >By SaDbhyo luk (7.1.22) :</p>";

echo "<p class = sa >षड्भ्यो लुक्‌ (७.१.२२) :</p>";

echo "<p class = pa >pratyayasya lukzlulupaH (1.1.61) </p>";

echo "<p class = pa >प्रत्ययस्य लुक्श्लुलुपः (१.१.६१) </p>";

display(3);

$luk = 1; // 0 - luk has not happened. 1 - luk has happened.

$pada="pada"; // when pratyayas are elided, the word gets pada saJjJA, because there is no pratyaya remaining after that.

} else { $luk = 0; }

### /\* pratyayalope pratyayalakSaNam (1.1.62) and na lumatAGgasya (1.1.63) \*/

if ($luk === 1 )

{

echo "<p class = hn >pratyayalope pratyayalakSaNam (1.1.62) is overridden by na lumatAGgasya (1.1.63) :</p>";

echo "<p class = hn >प्रत्ययलोपे प्रत्ययलक्षणम्‌ (१.१.६२) का न लुमताङ्गस्य (१.१.६३) से बाध हुआ है । :</p>";

display(0);

}

if ($svamo===1 && $gender==="n" && $so==="su!" && $sambuddhi===1)

{

echo "<p class = hn >pratyayalope pratyayalakSaNam (1.1.62) is overridden by na lumatAGgasya (1.1.63) optionally.</p>";

echo "<p class = hn >प्रत्ययलोपे प्रत्ययलक्षणम्‌ (१.१.६२) का न लुमताङ्गस्य (१.१.६३) से बाध हुआ है । यह बाध अनित्य है । :</p>";

display(0);

}

### /\* jazzasoH ziH (7.1.20) \*/

if ($gender === "n" && $pada=== "pratyaya" && in\_array($so,array("jas","Sas")) && $luk===0)

{

$text = last(array("jas","Sas"),array("Si","Si"),0);

echo "<p class = sa >By jazzasoH ziH (7.1.20) :</p>";

echo "<p class = sa >जश्शसोः शिः (७.१.२०) :</p>";

echo "<p class = hn >N.B. anekAlzitsarvasya mandates sarvAdeza :</p>";

echo "<p class = hn >अनेकाल्शित्सर्वस्य से सर्वादेश होता है । :</p>";

display(3);

$shi = 1; // 0 - zi Adeza has not happened. 1 - zi Adeza has happened.

} else { $shi = 0; }

### /\* zi sarvanAmasthAnam (1.1.42) \*/

if ($shi===1 )

{

echo "<p class = pa >zi sarvanAmasthAnam (1.1.42) :</p>";

echo "<p class = pa >शि सर्वनामस्थानम्‌ (१.१.४२) :</p>";

display(0);

}

### /\* declaring sarvanamasthana1 variable \*/

if ( ( in\_array($so,$sarvanamasthana) && $gender!=="n") || $shi===1)

{

$sarvanamasthana1 = 1;

}

else

{

$sarvanamasthana1 = 0;

}

### /\* am sambuddhau (7.1.99) \*/

if ($so==="su!" && sub(array("catur","anaquh"),blank(0),blank(0),0) && $sambuddhi===1)

{

$text = one(array("catur+","anaquh+"),array("catuar+","anaquah+"),0);

echo "<p class = sa >By am sambuddhau (7.1.99) :</p>";

echo "<p class = sa >अम्‌ सम्बुद्धौ (७.१.९९) :</p>";

display(3);

}

### /\* puMso'suG (7.1.89) \*/

if ($sarvanamasthana1===1 && sub(array("puMs"),blank(0),blank(0),0))

{

$text = one(array("puMs+"),array("pumas+"),0);

echo "<p class = sa >By puMso'suG (7.1.89) :</p>";

echo "<p class = sa >पुंसोऽसुङ्‌ (७.१.८९) :</p>";

display(3);

$it=array\_merge($it,array("u")); // puMs is an ugit word.

$itprakriti=array\_merge($itprakriti,array("u"));

}

### /\* yujerasamAse (7.1.71) \*/

if ( sub(array("yuj"),$sup,blank(0),0) && $fo==="yuj" && $sarvanamasthana1===1 && $samasa===0 && $yuj===1)

{

$text = one(array("yuj",),array("yunj"),0);

echo "<p class = sa >By yujerasamAse (7.1.71) :</p>";

echo "<p class = sa >युजेरसमासे (७.१.७१) :</p>";

display(3);

}

### /\* A sarvanAmnaH (6.3.91) \*/

if ( sub($sarvanama,array("dfg","dfS","avatu"),blank(0),0) )

{

$text = two($tyadadi,array("dfS"),antya($tyadadi,"A"),array("dfS"),0);

echo "<p class = sa >By A sarvanAmnaH (6.3.91) :</p>";

echo "<p class = sa >आ सर्वनाम्नः (६.३.९१) :</p>";

display(0); $Asarva=1; // 0 - A sarvanAmnaH has not applied. 1 - A sarvanAmnaH has applied.

} else { $Asarva=0; }

### /\* defining tri variable for tisRcatasR \*/

if ($\_GET["cond2\_2\_1"]==="1" )

{

$tri="f";

}

else

{

$tri="m";

}

### /\* tricaturoH striyAM tisRcatasR (7.2.99) \*/

if ( (($tri === "f" && ends(array($fo),array("tri"),1)) || (ends(array($fo),array("catur"),1) && $gender==="f" ) ) && !ends(array($fo),array("stri"),1) && in\_array($so,$sup))

{

echo "<p class = sa >By tricaturoH striyAM tisRcatasR (7.2.99) :</p>";

if ($svamo===1)

{

echo "<p class = hn >By anityatva of na lumatAGgasya paribhASA.</p>";

$text = one(array("tri+","catur+"),array("tisf+","catasf+"),1);

}

else

{

$text = one(array("tri+","catur+"),array("tisf+","catasf+"),0);

}

echo "<p class = sa >त्रिचतुरोः स्त्रियां तिसृचतस्रू (७.२.९९) :</p>";

if ($svamo===1)

{

echo "<p class = hn >न लुमताऽङ्गस्य इति निषेधस्य अनित्यत्वम्‌ ।</p>";

}

display(3);

}

if ($gender==="f" && $tri === "f" && ends(array($fo),array("tri","catur"),1) && !ends(array($fo),array("stri"),1) && $svamo===1)

{

$text = one(array("tri+","catur+"),array("tisf+","catasf+"),1);

echo "<p class = sa >By tricaturoH striyAM tisRcatasR (7.2.99) and the optional nature of prohibition of 'na lumatA...' :</p>";

echo "<p class = sa >त्रिचतुरोः स्त्रियां तिसृचतस्रू (७.२.९९) तथा 'न लुमता..' इत्यस्य निषेधस्य अनित्यत्वम्‌ :</p>";

display(3);

}

### /\* caturanaDuhorAmudAttaH (7.1.98) \*/

if ($sarvanamasthana1 === 1 && sub(array("catur","anaquh"),blank(0),blank(0),0))

{

$text = one(array("catur+","anaquh+"),array("catuAr+","anaquAh+"),0);

echo "<p class = sa >By caturanaDuhorAmudAttaH (7.1.98) :</p>";

echo "<p class = sa >चतुरनडुहोरामुदात्तः (७.१.९८) :</p>";

display(3);

}

### /\* sAvanaDuhaH (7.1.82) \*/

if (sub(array("anaquAh+","anaquah+"),blank(0),blank(0),0) && $so==="su!" )

{

$text = one(array("anaquAh+","anaquah+"),array("anaquAnh+","anaquanh+"),0);

echo "<p class = sa >By sAvanaDuhaH (7.1.82) :</p>";

echo "<p class = sa >सावनडुहः (७.१.८२) :</p>";

display(3);

}

### /\* jarAyA jarasanyatarasyAm (7.2.101) \*/

if (arr($text,'/(jar)([aA])[+][a][m]/') && $pada=== "pratyaya" && $so==="su!" && $gender === "n" )

{

echo "<p class = pa >sannipAtaparibhASA prevents application of jarAyA jarasanyatarasyAm.</p>";

echo "<p class = pa >सन्निपातपरिभाषा से जराया जरसन्यतरस्याम्‌ (७.२.१०१) का बाध होता है ।</p>";

display(3);

}

if (arr($text,'/(jar)([aA])[+]/') && in\_array($so,$acsup) )

{

$text = one(array("jara+","jarA+"),array("jaras+","jaras+"),1);

echo "<p class = sa >By jarAyA jarasanyatarasyAm (7.2.101) :</p>";

echo "<p class = hn >By padAGgAdhikAre tasya ca tadantasya ca (pa 30) and anekAltvAtsarvAdeze prApte nirdizyamAnasyAdezA bhavanti (pa 13) :</p>";

echo "<p class = sa >जराया जरसन्यतरस्याम्‌ (७.२.१०१) :</p>";

echo "<p class = hn >पदाङ्गाधिकारे तस्य च तदन्तस्य च (प ३०) तथा अनेकाल्त्वात्सर्वादेशे प्राप्ते निर्दिश्यमानस्यादेशा भवन्ति (प १३) :</p>";

display(3);

}

### /\* paddannomAshRnnizasanyUSandoSanyakaJChakannudannAsaJChasprabhRtiSu (6.1.63) \*/

// The random examples given under vibhASA GizyoH on page 147 are pending. Will do them after understanding fully.

// kakuddoSaNI etc are pending.

$paddanno = array("pAda","danta","nAsikA","mAsa","hfdaya","niSA","asfj","yUza","doz","yakft","Sakft","udaka","Asya");

$paddanno1 = array("pad","dat","nas","mAs","hfd","niS","asan","yUzan","dozan","yakan","Sakan","udan","Asan");

if (sub($paddanno,array("+"),blank(0),0) && in\_array($so,$zasadi) && in\_array($fo,$paddanno))

{

$text = two($paddanno,array("+"),$paddanno1,array("+"),1);

echo "<p class = sa >By paddannomAshRnnizasanyUSandoSanyakaJChakannudannAsaJChasprabhRtiSu (6.1.63) :</p>";

echo "<p class = hn >prabhRtigrahaNaM prakArArtham. tena 'padaGghrIcaraNo'striyAm', 'svAntaM hRnmAnasaM manaH' etc are valid.</p>";

echo "<p class = sa >पद्दन्नोमास्‍हृन्निशसन्यूषन्दोषन्यकञ्छकन्नुदन्नासञ्छस्प्रभृतिषु (६.१.६३) :</p>";

echo "<p class = hn >प्रभृतिग्रहणं प्रकारार्थम्‌ । तेन 'पदङ्घ्रिचरणोऽस्त्रियाम्‌', 'स्वान्तं हृन्मानसं मनः' इत्यादि च सङ्गच्छते ।</p>";

display(0);

}

### /\* mAMsapRtanAsAnUnAM mAMspRtsnavo vAcyAH SasAdau vA (vA 3416) \*/

$mAMsa = array("mAMsa","pftanA","sAnu");

$mAMsa1 = array("mAMs","pft","snu");

if (sub($mAMsa,array("+"),array\_merge($zasadi,array("Si")),0) && in\_array($so,$zasadi) && in\_array($fo,$mAMsa))

{

$text = two($mAMsa,array\_merge($zasadi,array("Si")),$mAMsa1,array\_merge($zasadi,array("Si")),1);

echo "<p class = sa >By mAMsapRtanAsAnUnAM mAMspRtsnavo vAcyAH SasAdau vA (vA 3416) :</p>";

echo "<p class = sa >मांसपृतनासानूनां मांस्पृत्स्नवो वाच्याः शसादौ वा (वा ३४९६) :</p>";

display(0);

}

### /\* asthidadhisakthyakSNAmanaGudAttaH (7.1.75) \*/

$asthi = array("asTi","daDi","sakTi","akzi"); // defining an array of asthi, dadhi etc.

$asthi1 = array("asTan","daDan","sakTan","akzan");

if (sub($asthi,array("+"),$tRtIyAdiSvaci,0) && in\_array($so,$tRtIyAdiSvaci))

{

$text = two($asthi,$tRtIyAdiSvaci,$asthi1,$tRtIyAdiSvaci,0);

echo "<p class = sa >By asthidadhisakthyakSNAmanaGudAttaH (7.1.75) :</p>";

echo "<p class = sa >अस्थिदधिसक्थ्यक्ष्णामनङुदात्तः (७.१.७५) :</p>";

display(0);

}

### /\* saGkhyAvisAyapUrvasyAhnasyAhannanyatarasyAM Gau (6.3.110) \*/

$sankhyahan = array("ekAhna","dvyahna","tryahna","caturahna","paJcAhna","zaDAhna","saptAhna","azwAhna","navAhna","daSAhna","vyahna","sAyAhna"); // defining an array which satisfies needs of this sUtra.

if (sub($sankhyahan,array("+"),array("Ni"),0) && $so==="Ni")

{

$text = two(array("hna"),array("+"),array("han"),array("+"),1);

echo "<p class = sa >By saGkhyAvisAyapUrvasyAhnasyAhannanyatarasyAM Gau (6.3.110) :</p>";

echo "<p class = sa >सङ्ख्याविसायपूर्वस्याह्नस्याहन्नन्यतरस्याम्‌ (६.३.११०) :</p>";

display(0);

}

### /\* sakhyurasambuddhau (7.1.92) \*/

if (sub(array("saKi","saKI"),array("+"),array("O","jas","am","Ow"),0) && $sambuddhi===0 && $gender==="f")

{

echo "<p class = pa >vibhaktau liGgaviziSTasyAgrahaNam (pa) overrules prAtipadikagrahaNe liGgaviziSTasyApi grahaNam. :</p>";

echo "<p class = hn >This bars application of sakhyurasambuddhau. </p>";

echo "<p class = pa >विभक्तौ लिङ्गविशिष्टस्याग्रहणम्‌ (प) से प्रातिपदिकग्रहणे लिङ्गविशिष्टस्यापि ग्रहणम्‌ का निषेध होता है ।</p>";

echo "<p class = hn >इस से सख्युरसम्बुद्धौ का प्रतिषेध होता है ।</p>";

display(0);

}

if (sub(array("saKi","saKI"),$sarvanamasthana,blank(0),0) && $\_GET['cond1\_4\_3']!=="3" && $sambuddhi===0 && $\_GET['cond1\_3\_1'] !== "3" && $gender!=="f")

{

$Nidvat = 1; // 0 - no NidvadbhAva. 1 - NidvadbhAva.

echo "<p class = pa >By sakhyurasambuddhau (7.1.92) :</p>";

echo "<p class = pa >सख्युरसम्बुद्धौ (७.१.९२) :</p>";

if (!in\_array($fo,array("saKi","saKI")))

{

echo "<p class = hn >anaG and NidvadbhAva are done because of aGga. Therefore, they apply also with tadanta words. </p>";

echo "<p class = hn >अनङ्णिद्वद्भावयोराङ्गत्वात्तदन्तेऽपि प्रवृत्तिः । </p>";

}

display(3);

} else {$Nidvat=0; }

### /\* anaG sau (7.1.93) and Gicca (1.1.53) \*/

if (sub(array("saKi","saKI"),array("+"),array("su!"),0) && $so==="su!" && $sambuddhi===0 && $gender==="f")

{

echo "<p class = pa >vibhaktau liGgaviziSTasyAgrahaNam (pa) overrules prAtipadikagrahaNe liGgaviziSTasyApi grahaNam. :</p>";

echo "<p class = hn >This bars application of anaG sau. </p>";

echo "<p class = pa >विभक्तौ लिङ्गविशिष्टस्याग्रहणम्‌ (प) से प्रातिपदिकग्रहणे लिङ्गविशिष्टस्यापि ग्रहणम्‌ का निषेध होता है ।</p>";

echo "<p class = hn >इस से अनङ्‌ सौ का प्रतिषेध होता है ।</p>";

display(0);

}

if (sub(array("saKi","saKI"),array("+"),array("su!"),0) && $\_GET['cond1\_4\_3']!=="3" && $so==="su!" && $\_GET['cond1\_3\_1'] !== "3" && $sambuddhi===0 && $gender!=="f")

{

$text = two(array("saKi","saKI"),array("+"),array("saKan","saKan"),array("+"),0);

echo "<p class = sa >By anaG sau (7.1.93) and Gicca (1.1.53) :</p>";

echo "<p class = sa >अनङ्‌ सौ (७.१.९३) तथा ङिच्च (१.१.५३) :</p>";

if (!in\_array($fo,array("saKi","saKI")))

{

echo "<p class = hn >anaG and NidvadbhAva are done because of aGga. Therefore, they apply also with tadanta words. </p>";

echo "<p class = hn >अनङ्णिद्वद्भावयोराङ्गत्वात्तदन्तेऽपि प्रवृत्तिः । </p>";

}

display(3);

}

### /\* alo'ntyAtpUrva upadhA (1.1.65) \*/

// saJjJA sUtra. Nothing to code here.

### /\* diva aut (7.1.84) \*/

if (sub(array("div+"),array("su!"),blank(0),0))

{

$text = one(array("div+"),array("diO+"),0);

echo "<p class = sa >By diva aut (7.1.84) :</p>";

echo "<p class = sa >दिव औत्‌ (७.१.८४) :</p>";

display(3);

echo "<p class = pa >Because it is alvidhi, sthAnivadbhAva doesn't happen and 'halGyAp..' doesn't apply. </p>";

echo "<p class = pa >अल्विधि होने के कारण स्थानिवद्भाव नहीं है । अतः हल्ङ्‍याप्‌.. की प्रवृत्ति नहीं है ।</p>";

display(0);

}

### /\* diva ut (6.1.131) \*/

// uttarapadatve cApadAdividhau pratiSedhaH (vA) pending.

// dadhisecau example is also pending.

if (sub(array("div+"),$hlsup,blank(0),0) || arr($text,'/[d][i][v][+]$/') )

{

$text = one(array("div+"),array("diu+"),0);

echo "<p class = sa >By diva ut (6.1.131) :</p>";

echo "<p class = sa >दिव उत्‌ (६.१.१३१) :</p>";

display(0);

}

### /\* zeSo ghyasakhi (1.4.7) \*/

if ($nadi!==1 && arr($text,'/[iu][+]/') && $fo!=="saKi" && !arr($text,'/[iu][+]$/'))

{

if (arr(array($fo),'/[p][a][t][i]$/') && $fo==='pati')

{

$ghi = 0; // 0 - no ghi saJjJA. 1 - ghi saJjJA.

echo "<p class = pa >By patiH samAsa eva (1.4.8), the ghi saJjJA is not ascribed. </p>";

echo "<p class = pa >पतिः समास एव (१.४.८) से घिसञ्ज्ञा नहीं है । </p>";

display(0);

}

else

{

$ghi = 1;

echo "<p class = pa >By zeSo ghyasakhi (1.4.7) :</p>";

echo "<p class = pa >शेषो घ्यसखि (१.४.७) :</p>";

display(0);

}

}

else

{

$ghi = 0;

}

### /\* na mu ne (8.2.3) \*/

//adas + wA -> nAbhAva has to happen. Special case.

if ( $fo==="adas" && $so==="wA" && $gender!=="f")

{

$text = one(array("adas"),array("amu"),0);

echo "<p class = sa >By adaso'serdAdu do maH (8.2.80) :</p>";

echo "<p class = sa >अदसोऽसेर्दादु दो मः (८.२.८०) :</p>";

display(0);

echo "<p class = pa >By na mu ne (8.2.3) :</p>";

echo "<p class = pa >न मु ने (८.२.३) :</p>";

$ghi=1;

display(0);

}

### /\* yaci bham (1.4.14) and A kaDArAdekA saJjJA (1.4.1) \*/

// Not coded perfectly. Only for sup pratyayas.

if ($sarvanamasthana1 === 0 && in\_array($so,$acsup) && $luk===0 && $svamo===0 )

{

echo "<p class = pa >By yaci bham (1.4.14) and A kaDArAdekA saJjJA (1.4.1) :</p>";

echo "<p class = pa >यचि भम्‌ (१.४.१४) तथा आ कडारादेका सञ्ज्ञा (१.४.१) :</p>";

display(0);

$bham=1; // 0 - no bha saJjJA. 1 - bha saJjJA.

} else {$bham = 0; }

### /\* svAdiSvasarvanAmasthAne (1.4.17) \*/

if ($sarvanamasthana1 ===0 && in\_array($so,$sup) && $bham!==1 )

{

$pada="pada"; // this sUtra mandates pada saJjJA in these cases.

echo "<p class = pa >By svAdiSvasarvanAmasthAne (1.4.17) :</p>";

echo "<p class = pa >स्वादिष्वसर्वनामस्थाने (१.४.१७) :</p>";

display(0);

}

### /\* SaTcaturbhyazca (7.1.55) \*/

if ( $\_GET['cond1\_6\_1']==="1")

{

echo "<p class = pa >'nuT' Agama doesn't happen in case of gauNatva.</p>";

echo "<p class = pa >गौणत्वे तु नुट्‌ नेष्यते ।</p>";

display(0);

}

elseif ( ($shat===1 || arr(array($fo),'/[c][a][t][u][r]$/')) && $so === "Am" && !in\_array($fo,$sarvanama) && ($samasa===0 || $samasa===1 && $pradhana===1) && $\_GET['cond1\_6\_1']!=="1")

{

$text = one(array("+Am"),array("+nAm"),0);

echo "<p class = sa >By SaTcaturbhyazca (7.1.55) :</p>";

echo "<p class = sa >षट्चतुर्भ्यश्च (७.१.५५) :</p>";

display(3);

$Satcatur=1; // 0 - SaTcaturbhyazca has not applied. 1 - SaTcaturbhyazca has applied.

$pada="pada"; // word gets pada saJjJA.

} else { $Satcatur=0; }

### /\* patch for aSTana A vibhaktau \*/

if ( sub(array("azwan"),array("nAm"),blank(0),1) && $so === "Am" && ($samasa===0 || ($samasa===1 && $pradhana===1)))

{

$text = two(array("azwan"),array("+nAm"),array("azwaA"),array("+nAm"),0);

echo "<p class = sa >By aSTana A vibhaktau (7.2.84) :</p>";

echo "<p class = hn >Atva of aSTana A vibhaktau is optional, as shown by 'aSTano dIrghAt'.</p>";

echo "<p class = sa >अष्टन आ विभक्तौ (७.२.८४)</p>";

echo "<p class = hn >अष्टन आ विभक्तौ से हुआ आत्व वैकल्पिक है । अष्टनो दीर्घात्‌ इस सूत्र में दीर्घग्रहण के ज्ञापक से । </p>";

display(3);

}

### /\* nopadhAyAH (6.4.7) \*/

if ( arr($text,'/[n][+]/') && $so === "Am" && !in\_array($fo,$sarvanama) && ($samasa===0 || ($samasa===1 && $pradhana===1)))

{

$text = three($ac,array("n"),array("+nAm"),$acdir,array("n"),array("+nAm"),0); // for function three, please see function.php.

echo "<p class = sa >By nopadhAyAH (6.4.7) :</p>";

echo "<p class = sa >नोपधायाः (६.४.७) :</p>";

display(3);

$nopadha=1; // 0 - word doesn't have 'n' as upadhA. 1 - word has 'n' as upadhA.

} else { $nopadha=0; }

### /\* pAdaH pat (6.4.130) \*/

if (sub(array("pAd"),array("+"),$sup,0) && $bham===1)

{

$text = two(array("pAd"),$sup,array("pad"),$sup,0);

echo "<p class = sa >By pAdaH pat (6.4.130) :</p>";

echo "<p class = sa >पादः पत्‌ (६.४.१३०):</p>";

display(3);

}

### /\* TeH (6.4.143) \*/

if ($Dit===1 && $bham===1 )

{

$text = Ti(0); // function Ti removes Ti. see function.php for details.

echo "<p class = sa >By TeH (6.4.143) :</p>";

echo "<p class = sa >टेः (६.४.१४३) :</p>";

display(3);

}

### /\* bhasya TerlopaH (7.1.88) \*/

if (sub(array("paTin","maTin","fBukzin"),blank(0),blank(0),0) && $bham===1 && !arr($text,'/[i][n][+]$/'))

{

$text = one(array("paTin","maTin","fBukzin"),array("paT","maT","fBukz"),0);

echo "<p class = sa >By bhasya TerlopaH (7.1.88) :</p>";

echo "<p class = sa >भस्य टेर्लोपः (७.१.८८) :</p>";

display(3);

}

if (sub(array("paTin+I","maTin+I","fBukzin+I",),blank(0),blank(0),1) && $gender==="f" && $bham===1) // patch for supathI.

{

$text = one(array("paTin+I","maTin+I","fBukzin+I",),array("paTI","maTI","fBukzI"),0);

echo "<p class = sa >By bhasya TerlopaH (7.1.88) :</p>";

echo "<p class = sa >भस्य टेर्लोपः (७.१.८८) :</p>";

display(3);

}

### /\* pathimathyRbhukSAmAt (7.1.85) \*/

if (sub(array("paTin","maTin","fBukzin"),array("+"),blank(0),0) && $so==="su!" && $gender==="m")

{

$text = two(array("paTin","maTin","fBukzin"),array("+"),array("paTiA","maTiA","fBukziA"),array("+"),0);

echo "<p class = sa >By pathimathyRbhukSAmAt (7.1.85) :</p>";

echo "<p class = sa >पथिमथ्यृभुक्षामात्‌ (७.१.८५) :</p>";

display(3);

$pathi=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied. Useful for tho'nthaH (7.1.87).

} else {$pathi=0; }

### /\* ito'tsarvanAmasthAne (7.1.85) \*/

if (sub(array("paTi","maTi","fBukzi"),blank(0),blank(0),0) && $sarvanamasthana1===1 )

{

$text = one(array("paTi","maTi","fBukzi"),array("paTa","maTa","fBukza"),0);

echo "<p class = sa >By ito'tsarvanAmasthAne (7.1.85) :</p>";

echo "<p class = sa >इतोऽत्सर्वनामस्थाने (७.१.८५) :</p>";

display(3);

$pathi1=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied. Useful for tho'nthaH (7.1.87).

} else {$pathi=0;}

### /\* tho'nthaH (7.1.87) \*/

if (sub(array("paTa","maTa"),blank(0),blank(0),0) && ($pathi===1 || $pathi1===1))

{

$text = one(array("paTa","maTa",),array("panTa","manTa",),0);

echo "<p class = sa >By tho'nthaH (7.1.87) :</p>";

echo "<p class = sa >थोऽन्थः (७.१.८७) :</p>";

display(3);

}

### /\* sambuddhau napuMsakAnAM nalopo vA vAcyaH (vA) \*/

// Pending. Not clear to me.

### /\* sau ca (6.4.13) \*/

$noin=array("ahan","Ahan","bahuvftrahan","bahupUzan","bahvaryaman"); // words where inhan.. doesn't apply.

$acdir = array("A","A","I","I","U","U","F","F","F","F","e","o","E","O",);

if (sub(array("in","han","pUzan","aryaman"),array("+"),array("su!"),1) && !in\_array($fo,$noin) && in\_array($so,array("su!")) && $sambuddhi===0)

{

$text = two($ac,array("n+"),$acdir,array("n+"),0);

echo "<p class = sa >By sau ca (6.4.13) :</p>";

echo "<p class = sa >सौ च (६.४.१३) :</p>";

if (sub(array("vIn+"),blank(0),blank(0),0))

{

echo "<p class = pa >By 'aninasmangrahaNAni arthavatA cAnarthakena ca tadantavidhiM prayojayanti' (pa) :</p>";

echo "<p class = pa >अनिनस्मन्ग्रहणानि अर्थवता चानर्थकेन च तदन्तविधिं प्रयोजयन्ति (प) :</p>";

}

display(3);

$inhan=1; // 0 - the word is not eligible for inhanpU... 1 - the word is eligible for inhanpU..

} else {$inhan=0; }

### /\* inhanpUSAryamNAM zau (6.4.12) \*/

if (sub(array("in","han","pUzan","aryaman"),array("+"),$sarvanamasthana,0) && !sub(array("ahan","Ahan",),array("+"),array("su!"),0) && $sambuddhi===0 && $inhan===0 && !in\_array($fo,array("ahan","dIrGAhan")))

{

$text = two(array("in","han","pUzan","aryaman"),array("Si"),array("In","hAn","pUzAn","aryamAn"),array("Si"),0);

echo "<p class = sa >By inhanpUSAryamNAM zau (6.4.12) :</p>";

echo "<p class = sa >इन्हन्पूषार्यम्णां शौ (६.४.१२) :</p>";

if (sub(array("vin+"),blank(0),blank(0),0))

{

echo "<p class = pa >By 'aninasmangrahaNAni arthavatA cAnarthakena ca tadantavidhiM prayojayanti' (pa) :</p>";

echo "<p class = pa >अनिनस्मन्ग्रहणानि अर्थवता चानर्थकेन च तदन्तविधिं प्रयोजयन्ति (प) :</p>";

}

display(3); $inhan1=1; // 0 - inhan.. has not applied. 1 - inhan.. has appied.

} else { $inhan1=0; }

### /\* sarvanAmasthAne cAsambuddhau (6.4.8) \*/

//if ( arr($text,'/['.flat($ac).'][n][+]/') && !arr($text,'/['.flat($ac).'][n][+]$/') && !sub(array("Ahan"),blank(0),blank(0),0) && $sarvanamasthana1===1 && $sambuddhi===0 && $inhan===0 && $inhan1===0 && !in\_array($fo,array("maGavan")) ) // To check for dIrghAhan

if ( arr($text,'/['.flat($ac).'][n][+]/') && !arr($text,'/['.flat($ac).'][n][+]$/') && $sarvanamasthana1===1 && $sambuddhi===0 && $inhan===0 && $inhan1===0 && !in\_array($fo,array("maGavan")) )

{

$text = two($ac,array("n+"),$acdir,array("n+"),0);

echo "<p class = sa >By sarvanAmasthAne cAsambuddhau (6.4.8) :</p>";

echo "<p class = sa >सर्वनामस्थाने चासम्बुद्धौ (६.४.८) :</p>";

echo "<p class = pa >alo'ntyAtpUrva upadhA (1.1.65) </p>";

echo "<p class = pa >अलोऽन्त्यात्पूर्व उपधा (१.१.६५) </p>";

display(3);

$casambu=1; // 0 - This sUtra has not applied. 1 - This sUtra has applied. Useful for preventing repeat application.

} else {$casambu=0; }

### /\* tRjvatkroSTuH (7.1.95), vibhASA tRtIyAdiSvaci (7.1.97) and numaciratRjvadbhAvebhyo nuT pUrvavipratiSedhena (vA 4374) \*/

if ((sub(array("krozwu"),array("+"),$sarvanamasthana,0)||sub(array("krozwu"),array("+"),$tRtIyAdiSvaci,0) ) && $sarvanamasthana1===1 && $pada==="pratyaya" && ($sambuddhi===0 || ($sambuddhi===1 && $so!=="su!")))

{

if ( (in\_array($so,$sarvanamasthana)|| in\_array($so,$tRtIyAdiSvaci) || arr($text,'/[+][S][i]$/')) )

{

echo "<p class = pa >tRjvadbhAva of tRjvatkroSTuH is overruled by 'num' by vRddhyauttvatRjvadbhAvaguNebhyo num pUrvavipratiSedhena (vA 4373). </p>";

echo "<p class = pa >तृज्वत्क्रोष्टुः से प्राप्त तृज्वद्भाव का वृद्ध्यौत्त्वतृज्वद्भावगुणेभ्यो नुम्‌ पूर्वविप्रतिषेधेन (वा ४३७३) से बाध होता है ।</p>";

display(0);

}

elseif (sub(array("krozwu"),array("+"),$sarvanamasthana,0))

{

$text = two(array("krozwu"),array("+"),array("krozwf"),array("+"),0);

echo "<p class = sa >By tRjvatkroSTuH (7.1.95) :</p>";

echo "<p class = sa >तृज्वत्क्रोष्टुः (७.१.९५) :</p>";

display(3);

$tRcvat=1;

}

}

if (sub(array("krozwu"),array("+"),$tRtIyAdiSvaci,0) && $so!=="Am")

{

$text = two(array("krozwu"),array("+"),array("krozwf"),array("+"),1);

echo "<p class = sa >vibhASA tRtIyAdiSvaci (7.1.97) :</p>";

echo "<p class = sa >विभाषा तृतीयादिष्वचि (७.१.९७) :</p>";

display(3);

$tRcvat=1;

}

if (sub(array("krozwu",),array("+"),array("Am"),0) && $so==="Am")

{

$text = two(array("krozwu"),array("+Am"),array("krozwu"),array("+nAm"),0);

echo "<p class = sa >numaciratRjvadbhAvebhyo nuT pUrvavipratiSedhena (vA 4374) :</p>";

echo "<p class = hn >This vArttika mandates a 'nuT' Agama. :</p>";

echo "<p class = sa >नुमचिरतृज्वद्भावेभ्यो नुट्‌ पूर्वविप्रतिषेधेन (वा ४३७४) :</p>";

echo "<p class = hn >यह वार्तिक नुट्‌ आगम का विधान करता है ।</p>";

display(0);

$numacira=1; // 0 - numacira.. has not applied. 1 - numacira.. has applied.

} else { $numacira = 0;}

// atisakhi is pending to code. page 158. Understand and then code.

### /\* AGo nA'striyAm (7.3.120) \*/

if ($ghi===1 && $so==="wA" && $gender !== "f" && arr($text,'/[iu][+]/') && !(in\_array($fo,array("suDI","praDI")) && $gender==="n") )

{

if ($bhashitapumska===0 )

{

$text = two(array("i+","u+"),array("wA"),array("i+","u+"),array("nA"),0);

}

else

{

$text = two(array("i+","u+"),array("wA"),array("i+","u+"),array("nA"),1);

$text = two(array("krozwu+",),array("wA"),array("krozwu+",),array("nA"),0);

}

echo "<p class = sa >By AGo nA'striyAm (7.3.120) :</p>";

echo "<p class = sa >आङो नाऽस्त्रियाम्‌ (७.३.१२०) :</p>";

display(3);

$AGo = 1; // 0 - no nAbhAva. 1 - nAbhAva.

} else {$AGo = 0; }

### /\* defining some sets \*/

$ik = array("i","I","u","U","f","F","x","X"); // ik pratyAhAra.

$ikyan = array("y","y","v","v","ar","Ar","ar","Ar"); // yaN replacements of ik pratyAhAra.

$acsup = array\_merge($acsup,array("SI","Si")); // adding Si of jas/Sas as ajAdivibhakti.

### /\* iko'ci vibhaktau (7.1.73) \*/

if ( $gender==="n" && arr($text,'/['.flat($ik).'][+]/') && in\_array($so,$acsup) && sub($ik,array("+"),$acsup,0))

{

if ($ghi===1 && in\_array($so,array("Ne","Nasi!","Nas","Ni")) && $bhashitapumska===0)

{

echo "<p class = pa >guNa of gherGiti is overruled by 'num' by vRddhyauttvatRjvadbhAvaguNebhyo num pUrvavipratiSedhena (vA 4373). </p>";

echo "<p class = pa >घेर्ङिति से प्राप्त गुण का वृद्ध्यौत्त्वतृज्वद्भावगुणेभ्यो नुम्‌ पूर्वविप्रतिषेधेन (वा ४३७३) से बाध होता है ।</p>";

display(0);

$noghe=1; // 0 - gherGiti will not apply. 1 - gherGiti will apply.

} else {$noghe=0; }

if (in\_array($so,array("Am")))

{

echo "<p class = pa >'num' of 'iko'ci vibhaktau' is barred by 'nut' because of 'numaciratRjvadbhAvebhyo nuT pURvavipratiSedhena (vA 4374). </p>";

echo "<p class = pa >इकोऽचि विभक्तौ से प्राप्त नुमागम का 'नुमचिरतृज्वद्भावेभ्यो नुट्‌ पूर्वविप्रतिषेधेन (वा ४३७४) से बाध होता है ।</p>";

display(0);

}

elseif ($ghi===1 && $AGo===1 && $bhashitapumska===0)

{

echo "<p class = pa >AGo nA'striyAm has barred application of iko'ci vibhaktau. </p>";

echo "<p class = pa >आङो नाऽस्त्रियाम्‌ से इकोऽचि विभक्तौ का बाध हुआ है ।</p>";

display(0);

}

elseif ( $bhashitapumska === 1 && in\_array($so,array("wA","Ne","Nasi!","Nas","Ni","os",)))

{

$text = three(array("i","u","f","x"),array("+"),$acsup,array("i","u","f","x"),array("n+"),$acsup,1);

// patch for sudhI

$text = three(array("suDi","praDi","krozwfn","sulUn"),array("+"),$acsup,array("suDI","praDI","krozwf","sulun"),array("+"),$acsup,0);

$text = three(array("sulu",),array("+"),array("wA"),array("sulun",),array("+"),array("wA"),0);

echo "<p class = sa >By iko'ci vibhaktau (7.1.73) :</p>";

echo "<p class = sa >इकोऽचि विभक्तौ (७.१.७३) :</p>";

display(3);

}

else

{

$text = three(array("i","u","f","x"),array("+"),$acsup,array("i","u","f","x"),array("n+"),$acsup,0);

echo "<p class = sa >By iko'ci vibhaktau (7.1.73) :</p>";

echo "<p class = sa >इकोऽचि विभक्तौ (७.१.७३) :</p>";

display(3);

}

$ikoci=1; // 0 - iko'ci vibhaktau has not applied. 1 - iko'ci vibhaktau has applied.

} else { $ikoci = 0; $noghe=0;}

### /\* vA'mzasoH (6.4.80) \*/

if (ends(array($fo),array("strI","stri"),1) && in\_array($so,array("am","Sas")) && sub(array("strI","stri"),array("am","Sas"),blank(0),0))

{

$text = one(array("strI+","stri+"),array("striy+","striy+"),1);

echo "<p class = sa >By vA'mzasoH (6.4.80) :</p>";

echo "<p class = sa >वाऽम्शसोः (६.४.८०) :</p>";

display(0);

}

### /\* lomno'patyeSu bahuSvakAro vaktavyaH (vA 2560) \*/

// Pending, because it is for taddhita derivation. Right now made a patch.

$bahusup = array("jas","Sas","Bis","Byas","Am","sup");

if (sub(array("Oqulomi"),array("+"),$bahusup,0) && in\_array($so,$bahusup))

{

$text = two(array("Oqulomi"),$bahusup,array("uquloma"),$bahusup,0);

echo "<p class = sa >By lomno'patyeSu bahuSvakAro vaktavyaH (vA 2560) :</p>";

echo "<p class = sa >लोम्नोऽपत्येषु बहुष्वकारो वक्तव्यः (वा २५६०) :</p>";

display(0);

}

### /\* aci ra RtaH (7.2.100) \*/

if (sub(array("tisf","catasf"),array("+"),$acsup,0))

{

if ($so==="Am" )

{

$text = two(array("tisf","catasf"),array("+Am"),array("tisf","catasf"),array("+nAm"),0);

echo "<p class = sa >numaciratRjvadbhAvebhyo nuT pUrvavipratiSedhena (vA 4374) :</p>";

echo "<p class = hn >This vArttika mandates a 'nuT' Agama. :</p>";

echo "<p class = sa >नुमचिरतृज्वद्भावेभ्यो नुट्‌ पूर्वविप्रतिषेधेन (वा ४३७४) :</p>";

echo "<p class = hn >यह वार्तिक नुट्‌ आगम का विधान करता है ।</p>";

$numacira=1;

}

else

{

$text = two(array("tisf","catasf"),$acsup,array("tisr","catasr"),$acsup,0);

echo "<p class = sa >By aci ra RtaH (7.2.100) :</p>";

echo "<p class = sa >अचि र ऋतः (७.२.१००) :</p>";

}

display(0);

}

### /\* RduzanaspurudaMso&nehasAM ca (7.1.94) \*/

// displaying the output for dhAtR napuMsaka

if (arr($text,'/[fx][+]$/') && $svamo===1)

{

echo "<p class = pa >By na lumatAGgasya, anaG is prevented.</p>";

echo "<p class = pa >न लुमताङ्गस्य से अनङ्‌ का प्रतिषेध होता है ।</p>";

display(0);

}

if ( (arr($text,'/[fx][+]/')|| $fo==="uSanas"|| $fo==="purudaMsas" || $fo==="anehas" ) && $so==="su!" && $pada==="pratyaya" && $sambuddhi===0 && $svamo===0)

{

$text = two(array("f","x","uSanas","purudaMsas","anehas"),array("+"),array("an","an","uSanan","purudaMsan","anehan"),array("+"),0);

echo "<p class = sa >By RduzanaspurudaMso'nehasAM ca (7.1.94) :</p>";

echo "<p class = sa >ऋदुशनस्पुरुदंसोऽनेहसां च (७.१.९४) :</p>";

display(3);

}

if ( $fo==="uSanas" && $so==="su!" && $sambuddhi===1)

{

$text = two(array("uSanas"),array("+"),array("uSanan"),array("+"),1);

$text = two(array("uSanas"),array("+"),array("uSana"),array("+"),1);

echo "<p class = sa >By asa sambuddhau vA'naG nalopazca vA vAcyaH (vA 5037) :</p>";

echo "<p class = sa >अस्य सम्बुद्धौ वाऽनङ्‌ नलोपश्च वा वाच्यः (वा ५०३७) :</p>";

display(3);

}

### /\* Rto GitsarvanAmasthAnayoH (7.3.110) \*/

if (arr($text,'/[fx][+]/') && ( $sarvanamasthana1===1 || $so==="Ni") && $pada==="pratyaya" && $svamo===0)

{

$text = two(array("f","x"),array("+"),array("ar","al"),array("+"),0);

echo "<p class = sa >By Rto GisarvanAmasthAnayoH (7.3.110) and uraNraparaH (1.1.51) :</p>";

echo "<p class = sa >ऋतो ङिसर्वनामस्थानयोः (७.३.११०) तथा उरण्रपरः (१.१.५१) :</p>";

display(3);

}

### /\* aptRntRcsvasRnaptRneSTRtvaSTRkSattRhotRpotRprazAstRRNAm (6.4.11) \*/

// Not coded perfectly for tRn and tRc. naptrAdigrahaNaM vyutpattipakSe niyamArtham is pending.

$svasR = array("svasf","naptf","nezwf","tvazwf","kzattf","hotf","potf","praSAstf");

$excludesvasR = array("pitf","BrAtf","jAmAtf","mantf","hantf","nanAndf","duhitf","yAtf","mAtf","katara","itara","anyatara");

if (((sub(array("tar+","war+"),blank(0),blank(0),0) )|| ($tRcvat===1&& $gender!=="n" )||in\_array($fo,$svasR) || in\_array($fo,$svasrAdi)) && $sarvanamasthana1===1 && $pada==="pratyaya" && !in\_array($fo,$excludesvasR) && (($sambuddhi===1 && $so!=="su!" ) || $sambuddhi===0) )

{

$text = three($ac,array("r"),array("+"),dirgha($ac),array("r"),array("+"),0);

$text = one(array("svasan","naptan","nezwan","tvazwan","kzattan","hotan","potan","praSAstan"),array("svasAn","naptAn","nezwAn","tvazwAn","kzattAn","hotAn","potAn","praSAstAn"),0);

echo "<p class = sa >By aptRntRcsvasRnaptRneSTRtvaSTRkSattRhotRpotRprazAstRRNAm (6.4.11) :</p>";

echo "<p class = sa >अप्तृन्तृच्स्वसृनप्तृनेष्टृत्वष्टृक्षत्तृहोतृपोतृप्रशास्तॄणाम्‌ (६.४.११) :</p>";

display(3);

$aptRn=1; // 0 - aptRn.. has not appied. 1 - aptRn.. has applied.

} else { $aptRn=0; }

if (arr(array($fo),'/[a][p]$/') && $sarvanamasthana1===1 && $pada==="pratyaya" && (($sambuddhi===1 && $so!=="su!" ) || $sambuddhi===0))

{

$text = two(array("ap"),array("+"),array("Ap"),array("+"),0);

echo "<p class = sa >By aptRntRcsvasRnaptRneSTRtvaSTRkSattRhotRpotRprazAstRRNAm (6.4.11) :</p>";

echo "<p class = sa >अप्तृन्तृच्स्वसृनप्तृनेष्टृत्वष्टृक्षत्तृहोतृपोतृप्रशास्तॄणाम्‌ (६.४.११) :</p>";

display(3);

}

### /\* apo bhi (7.4.48) \*/

if (arr(array($fo),'/[a][p]$/') && arr($text,'/[a][p][+][B]/') && in\_array($so,array("Bis","Byas","ByAm")))

{

$text = two(array("ap"),array("+B"),array("at"),array("+B"),0);

echo "<p class = sa >By apo bhi (7.4.48) :</p>";

echo "<p class = sa >अपो भि (७.४.४८) :</p>";

display(3);

}

### /\* Ato dhAtoH (6.4.140) \*/

$haha = array("hAhA");

if ($bham === 1 && arr($text,'/[A][+]/') && !in\_array($fo,$haha) && ($\_GET['cond1\_2']==="1" || $\_GET['cond2\_1']==="1" ) && $Ap===0)

{

$text = two(array("A"),array("+"),array(""),array("+"),0);

echo "<p class = sa >By Ato dhAtoH (6.4.140) :</p>";

echo "<p class = sa >आतो धातोः (६.४.१४०) :</p>";

display(6);

}

if ($bham === 1 && arr($text,'/[A][+]/') && in\_array($fo,array('ktvA','SnA')) && $Ap===0)

{

$text = two(array("A"),array("+"),array(""),array("+"),1);

echo "<p class = sa >By Ato dhAtoH (6.4.140) :</p>";

echo "<p class = hn >AtaH iti yogavibhAgAdadhAtorapi AkAralopaH kvacit |</p>";

echo "<p class = sa >आतो धातोः (६.४.१४०) :</p>";

echo "<p class = hn >आतः इति योगविभागादधातोरपि आकारलोपः क्वचित्‌ ।</p>";

display(6);

}

### /\* na saMyogAdvamantAt (6.4.137) \*/

if ($bham === 1 && arr($text,'/['.pc('hl').'][vm][a][n][+]/'))

{

echo "<p class = sa >By na saMyogAdvamantAt (6.4.137) :</p>";

echo "<p class = sa >न संयोगाद्वमन्तात्‌ (६.४.१३७) :</p>";

display(3); $vamanta=1; // 0 - the word is not vamanta. 1 - the word is vamanta.

} else {$vamanta=0; }

### /\* zvayuvamaghonAmataddhite (6.4.133) \*/

if ($bham === 1 && $taddhita===0 && sub(array("Svan","yuvan","maGavan"),array("+"),blank(0),0))

{

$text= two(array("Svan","yuvan","maGavan"),array("+"),array("S+u+an","yu+u+an","maGa+u+an"),array("+"),0);

echo "<p class = sa >By zvayuvamaghonAmataddhite (6.4.133) :</p>";

echo "<p class = sa >श्वयुवमघोनामतद्धिते (६.४.१३३) :</p>";

display(3);

$samp=array\_merge($samp,array(1)); // add 1 to array if samprasAraNa happens.

}

### /\* vAha UTh (6.4.132) \*/

if ($bham===1 && sub(array("vAh"),array("+"),blank(0),0) )

{

$text = one(array("vAh+"),array("U+Ah+"),0);

echo "<p class = sa >By vAha UTh (6.4.132) :</p>";

echo "<p class = sa>वाह ऊठ्‌ (६.४.१३२) :</p>";

display(3); $samp=array\_merge($samp,array(1));

}

### /\* vasoH samprasAraNam (6.4.131) \*/

if ($bham===1 && sub(array("sedivasu!","vasu!"),array("+"),blank(0),0) && $vasu===1)

{

$text = one(array("sedivasu!+","vasu!+"),array("sed+u+asu!+","+u+asu!+"),0);

echo "<p class = sa >By vasoH samprasAraNam (6.4.131) :</p>";

if (sub(array("sed+u+asu!+"),blank(0),blank(0),0))

{

echo "<p class = hn >Even though the iDAgama is antaraGga, it doesnt hold in samprasAraNa by 'akRtavyUhAH pANinIyAH'.</p>";

}

echo "<p class = sa>वसोः सम्प्रसारणम्‌ (६.४.१३१) :</p>";

if (sub(array("sed+u+asu!+"),blank(0),blank(0),0))

{

echo "<p class = hn >अन्तरङ्गोऽपि इडागमः सम्प्रसारणविषये न प्रवर्तते, 'अकृतव्यूहाः' इति परिभाषया ।</p>";

}

display(3); $samp=array\_merge($samp,array(1));

}

### /\* samprasAraNAcca (6.1.108) \*/

if ( in\_array(1,$samp) && sub(array("+i+","+u+","U+"),$ac,blank(0),0) )

{

$text = two(array("+i+","+u+","U+"),$ac,array("+i+","+u+","U"),blank(count($ac)),0);

echo "<p class = sa >By samprasAraNAcca (6.1.108) :</p>";

echo "<p class = sa>सम्प्रसारणाच्च (६.१.१०८) :</p>";

display(0);

}

### /\* na samprasAraNe samprasAraNam (6.2.37) \*/

if ( in\_array(1,$samp) && arr($text,'/[yv][uU][+]/') )

{

echo "<p class = sa >By na samprasAraNe samprasAraNam (6.2.37) :</p>";

echo "<p class = sa>न सम्प्रसारणे सम्प्रसारणम्‌ :</p>";

display(0);

}

### /\* napuMsakAcca (7.1.19) \*/

if ( $gender==="n" && in\_array($so,array("O","Ow")))

{

$text = two(array("+"),array("Ow","O",),array("+"),array("SI","SI"),0);

echo "<p class = sa >By napuMsakAcca (7.1.19) :</p>";

echo "<p class = sa >नपुंसकाच्च (७.१.१९) :</p>";

display(3);

$napuMsakAcca=1; // 0 - napuMsakAcca has not applied. 1 - napuMsakAcca has applied.

} else { $napuMsakAcca=0; }

### /\* auGaH zyAM pratiSedho vAcyaH (vA) \*/

if (arr($text,'/[+][SI]/') && in\_array($so,array("O","Ow")))

{

echo "<p class = pa >By auGaH zyAM pratiSedho vAcyaH (vA) :</p>";

echo "<p class = pa >औङः श्यां प्रतिषेधो वाच्यः (वा) :</p>";

display(0);

$auGazyA = 1; // 0 - doesn't prevent application of yasyeti ca. 1 - prevents application of yasyeti ca.

} else { $auGazyA = 0; }

### /\* paddannomAshRnnizasanyUSandoSanyakaJChakannudannAsaJChasprabhRtiSu (6.1.63) special case kakuddoSaNI \*/

if (sub(array("doz"),array("+"),array("SI"),0) && in\_array($so,array("O","Ow")))

{

$text = two(array("doz"),array("+"),array("dozan"),array("+"),0);

echo "<p class = sa >By paddannomAshRnnizasanyUSandoSanyakaJChakannudannAsaJChasprabhRtiSu (6.1.63) :</p>";

echo "<p class = hn >prabhRtigrahaNaM prakArArtham. tathA ca auGaH zyAmapi dozannAdezaH. </p>";

echo "<p class = sa >पद्दन्नोमास्‍हृन्निशसन्यूषन्दोषन्यकञ्छकन्नुदन्नासञ्छस्प्रभृतिषु (६.१.६३) :</p>";

echo "<p class = hn >प्रभृतिग्रहणं प्रकारार्थम्‌ । तथा च औङः श्यामपि दोषन्नादेशः ।</p>";

display(0);

}

### /\* bhasya (6.4.129) and allopo'naH (6.4.134) and vibhASA GizyoH (6.4.236) and na saMyogAdvamantAt (6.4.137) \*/

if ($bham === 1 && $vamanta===0 && arr($text,'/[a][n][+]/') && !(sub(array("udan+","Asan"),blank(0),blank(0),0) && !in\_array($so,$yacibham)) )

{

if ( $so==="Ni" || $napuMsakAcca===1 )

{

$text = one(array("an+"),array("n+"),1);

echo "<p class = sa >By allopo'naH (6.4.134) and vibhASA GizyoH (6.4.236) :</p>";

echo "<p class = sa >अल्लोपोऽनः (६.४.१३४) तथा विभाषा ङिश्योः (६.४.२३६) :</p>";

display(6);

}

else

{

$text = one(array("an+"),array("n+"),0);

echo "<p class = sa >By allopo'naH (6.4.134) :</p>";

echo "<p class = sa >अल्लोपोऽनः (६.४.१३४) :</p>";

display(6);

}

$allopo=1; // 0 - allopa has not happened. 1 - allopa has happened.

} else {$allopo=0; }

### /\* ho hanterJNinneSu (7.3.54) \*/

// Jit and Nit pending. nakAra done.

if ( sub(array("h"),array("n"),blank(0),0) && arr(array($fo),'/[h][a][n]/') && !in\_array($fo,array("ahan","dIrGAhan")))

{

$text = two(array("h"),array("n"),array("G"),array("n"),0);

echo "<p class = sa >By ho hanterJNinneSu (7.3.54) :</p>";

echo "<p class = sa >हो हन्तेर्ञ्णिन्नेषु (७.३.५४) :</p>";

display(3);

$hohante=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $hohante=0; }

### /\* jasi ca (7.3.109) \*/

### /\* trestrayaH (7.1.53) \*/

if (arr($text,'/[t][r][i][+][A][m]$/') && !sub(array("stri"),array("+"),blank(0),0) && $so==="Am" )

{

$text = two(array("tri"),array("Am"),array("traya"),array("Am"),0);

echo "<p class = sa >By trestrayaH (7.1.53) :</p>";

echo "<p class = sa >त्रेस्त्रयः (७.१.५३) :</p>";

display(3);

}

if (arr($text,'/[t][r][i][+][A][m]$/') && !sub(array("stri"),array("+"),blank(0),0) && $so==="Am" && $fo!=="tri")

{

$text = two(array("tri"),array("Am"),array("traya"),array("Am"),1);

echo "<p class = sa >By trestrayaH (7.1.53) and 'gauNatve tu neti kecit' :</p>";

echo "<p class = sa >त्रेस्त्रयः (७.१.५३) तथा 'गौणत्वे तु नेति केचित्‌' :</p>";

display(3);

}

### /\* viSvagdevayozca TeradryaJcatAvapratyaye (6.3.92) \*/

$sarvanamaadri = array("sarvadri","viSvadri","uBadri","uBayadri","ataradri","atamadri","anyadri","anyataradri","itaradri","tvadri","tvadri","nemadri","simadri","tyadri","tadri","yadri","etadri","idadri","adadri","ekadi","dvadri","yuzmadri","asmadri","Bavadri","kadri","idakadri");

if ($ancu===1 && ( sub(array("vizvak","deva"),array("anc"),blank(0),0)||sub($sarvanama,array("anc"),blank(0),0) ) )

{

$text = three($sarvanama,array("anc"),array("+"),$sarvanamaadri,array("anc"),array("+"),0);

$text = three(array("vizvak","deva"),array("anc"),array("+"),array("vizvadri","devadri"),array("anc"),array("+"),0);

echo "<p class = sa >viSvagdevayozca TeradryaJcatAvapratyaye (6.3.92) :</p>";

echo "<p class = sa >विश्वग्देवयोश्च टेरद्र्याञ्चतावप्रत्यये (६.३.९२) :</p>";

display(0);

### /\* patch for iko yaNaci \*/

if(sub(array('i','I','u','U'),prat('ac'),blank(0),0) && $bham===0) // for function prat, see function.php.

{

$text = two(array('i','I','u','U'),prat('ac'),array('y','y','v','v'),prat('ac'),0);

echo "<p class = sa >By iko yaNaci (6.1.77) :</p>";

echo "<p class = sa >इको यणचि (६.१.७७) :</p>";

display(4);

}

if(sub(array("f","F","x","X"),prat('ac'),blank(0),0) && $bham===0)

{

$text = two(array("f","F","x","X"),prat('ac'),array("r","r","l","l"),prat('ac'),0);

echo "<p class = sa >By iko yaNaci (6.1.77) :</p>";

echo "<p class = sa >इको यणचि (५.१.७७) :</p>";

display(4);

}

}

### /\* adasa au sulopazca (7.2.107) \*/

if (sub(array("adas"),array("+"),array("su!"),0) && $so==="su!" && $fo==="adas")

{

$text = two(array("adas"),array("su!"),array("adaO"),array(""),0);

echo "<p class = sa >By adasa au sulopazca (7.2.107) :</p>";

echo "<p class = sa >अदस औ सुलोपश्च (७.२.१०७) :</p>";

display(3);

}

### /\* tyadAdInAmaH (7.2.102) \*/

if (sub(array("dvi"),array("+"),blank(0),0) && in\_array($so,$sup) && ends(array($fo),array("dvi"),1) && $\_GET['cond1\_3\_2']==="2" && $idamoma===0 && $svamo===0 && $noatvasatva===0)

{

$text = one(array("dvi"),array("dv+a"),0);

echo "<p class = sa >By tyadAdInAmaH (7.2.102) :</p>";

echo "<p class = hn >Only the words till 'dvi' are included. (vA 4468) :</p>";

echo "<p class = sa >त्यदादीनामः (७.२.१०२) :</p>";

echo "<p class = hn >द्विपर्यन्तानामेवेष्टिः (वा ४४६८) :</p>";

display(3);

}

$tyadadinamah = array("dv+a","tya+a","ta+a","ya+a","eta+a","ida+a","ada+a","eka+a","idaka+a"); // creating a set of words eligible for tyadAdInAmaH

$tyadadinamah1 = array("dva","tya","ta","ya","eta","ida","ada","eka","idaka"); // creating a list of converted words after tyadAdInAmaH.

if (sub($tyadadi,array("+"),blank(0),1) && !sub(array("dvi"),array("+"),blank(0),1) && in\_array($so,$sup) && $idamoma===0 && $svamo===0 && $noatvasatva===0 )

{

$text = one($tyadadi,$tyadadinamah,0);

echo "<p class = sa >By tyadAdInAmaH (7.2.102) :</p>";

echo "<p class = hn >Only the words till 'dvi' are included. (vA 4468) :</p>";

echo "<p class = sa >त्यदादीनामः (७.२.१०२) :</p>";

echo "<p class = hn >द्विपर्यन्तानामेवेष्टिः (वा ४४६८) :</p>";

display(3);

$text = one($tyadadinamah,$tyadadinamah1,0);

$text=one(array("etam"),array("eta+am"),0);

echo "<p class = sa >By ato guNe (6.1.96) :</p>";

echo "<p class = sa >अतो गुणे (६.१.९६) :</p>";

display(0);

}

### /\* ajAdyataSTAp (4.1.4) \*/

if ($gender==="f" && sub(array("a+"),$sup,blank(0),0) && in\_array($so,$sup) && ($SaTsvasrAdi=0 || sub(array("a+"),$sup,blank(0),0)))

{

$text = two(array("a+"),$sup,array("A+"),$sup,0);

echo "<p class = sa >By ajAdyataSTAp (4.1.4) :</p>";

echo "<p class = sa >अजाद्यतष्टाप्‌ (४.१.४) :</p>";

display(3);

$Ap=1;

}

/\*if ($gender==="f" && $Ap===1 && sub(array("ida","tya","ta","ya","eta","ada"),array("+"),blank(0),0) && in\_array($fo,array("idam","idakam","tyad","tad","yad","etad","adas")) && in\_array($so,$sup))

{

$text = one(array("ida+","tya+","ta+","ya+","eta+","ada+"),array("idaA+","tyaA+","taA+","yaA+","etaA+","adaA+"),0);

echo "<p class = sa >By ajAdyataSTAp (4.1.4) :</p>";

echo "<p class = sa >अजाद्यतष्टाप्‌ (४.१.४) :</p>";

display(3);

}\*/ // removed it and placed the code abovementioned. keeping it here in case it is needed in future.

### /\* akaH savarNe dIrghaH (6.1.101) \*/

if ($gender==="f" && $Ap===1 && sub(array("idaA","tyaA","taA","yaA","etaA","adaA"),array("+"),blank(0),0) && in\_array($fo,array("idam","idakam","tyad","tad","yad","etad","adas")) && in\_array($so,$sup))

{

$text = one(array("idaA","tyaA","taA","yaA","etaA","adaA"),array("idA+","tyA","tA","yA","etA","adA"),0);

echo "<p class = sa >By akaH savarNe dIrghaH (6.1.101) :</p>";

echo "<p class = sa >अकः सवर्णे दीर्घः (६.१.१०१) :</p>";

display(3);

}

### /\* adasa au sulopazca (7.2.107) \*/

if (sub(array("adakas"),array("+"),array("su!"),0) && $so==="su!" && $fo==="adakas")

{

$text = two(array("adakas"),array("su!"),array("adakaO"),array(""),1);

$text = two(array("adakas"),array("su!"),array("asukas"),array(""),0);

echo "<p class = sa >By adasa au sulopazca (7.2.107) and autvapratiSedhaH sAkackasya vA vaktavyaH[[1]](#footnote-0) sAdutvaM ca (vA 4482) :</p>";

echo "<p class = sa >अदस औ सुलोपश्च (७.२.१०७) तथा औत्वप्रतिषेधः साकच्कस्य वा वक्तव्यः सादुत्वं च (वा ४४८२) :</p>";

display(3);

}

### /\* tadoH saH sAvanantyayoH (7.2.106) \*/

$tyadadinamah3 = array("dva","tya","eta","ta","ida","ada","eka","idaka","tyA","tA","etA"); // eligible for application of this sUtra.

$tyadadinamah2 = array("dva","sya","eza","sa","isa","asa","eka","isaka","syA","sA","esA"); // conversion after application of this sUtra.

if (sub($tyadadinamah3,array("su!"),blank(0),0) && $sarvafinal!==0 && $noatvasatva===0)

{

$text = two($tyadadinamah3,array("su!"),$tyadadinamah2,array("su!"),0);

echo "<p class = sa >By tadoH saH sAvanantyayoH (7.2.106) :</p>";

echo "<p class = sa >तदोः सः सावनन्त्ययोः (७.२.१०६) :</p>";

display(3);

}

if (sub(array("adaO","adakaO"),blank(0),blank(0),0))

{

$text = one(array("adaO","adakaO"),array("asaO","asakaO"),0);

$text = one(array("adakas"),array("asukas"),0);

echo "<p class = sa >By tadoH saH sAvanantyayoH (7.2.106) :</p>";

echo "<p class = sa >तदोः सः सावनन्त्ययोः (७.२.१०६) :</p>";

display(3);

}

### /\* sarvanAmnaH smai (7.1.14) \*/

if (arr($text,'/[a][+][N][e]/') && $so === "Ne" && $sarvafinal!==0)

{

if ($sarvafinal===2)

{

$text = last(array("Ne"),array("smE"),1);

}

else

{

$text = last(array("Ne"),array("smE"),0);

}

echo "<p class = sa >By sarvanAmnaH smai (7.1.14) :</p>";

echo "<p class = sa >सर्वनाम्नः स्मै (७.१.१४) :</p>";

display(3);

$sarva =1; // 0 - sarvanAmnaH smai has not applied. 1 - sarvanAmnaH smai has applied.

} else { $sarva = 0; }

### /\* GasiGyoH smAtsminau (7.1.15) \*/

if (arr($text,'/[a][+][N]/') && $pada=== "pratyaya" && in\_array($so,array("Nasi!","Ni")) && $sarvafinal!==0)

{

if ($sarvafinal===2)

{

$text = last(array("Ni","Nasi!"),array("smin","smAt"),1);

}

else

{

$text = last(array("Ni","Nasi!"),array("smin","smAt"),0);

}

echo "<p class = sa >By GasiGyoH smAtsminau (7.1.15) :</p>";

echo "<p class = sa >ङसिङ्योः स्मात्स्मिनौ (७.१.१५) :</p>";

display(3);

$sarva1 =1; // 0 - GasiGyoH smAtsminau has not applied. 1 - GasiGyoH smAtsminau has applied.

} else { $sarva1 = 0; }

### /\* pUrvAdibhyo navabhyo vA (7.1.16) \*/

if (ends(array($fo),array("pUrva","para","avara","dakziRa","uttara","apara","aDara","sva","antara"),1) && $pada=== "pratyaya" && in\_array($so,array("Nasi!","Ni")) && $sarvafinal!==0)

{

$text = last(array("smin","smAt"),array("Ni","Nasi!"),1);

echo "<p class = sa >By pUrvAdibhyo navabhyo vA (7.1.16) :</p>";

echo "<p class = sa >पूर्वादिभ्यो नवभ्यो वा (७.१.१६) :</p>";

display(3);

$sarva1 =1;

} elseif ($sarva1 ===1)

{ $sarva1 = 1; }

else

{

$sarva1 = 0;

}

### /\* TAGasiGasAminAtsyAH (7.1.12) \*/

if ( in\_array($so,array("wA","Nas")) && arr($text,'/[a][+][wN]/'))

{

$text = one(array("a+wA","a+Nas"),array("a+ina","a+sya"),0);

$text = two(array("jaras"),array("ina","sya"),array("jaras"),array("wA","Nas"),0);

echo "<p class = sa >By TAGasiGasAminAtsyAH (7.1.12) :</p>";

echo "<p class = sa >टाङसिङसामिनात्स्याः (७.१.१२) :</p>";

display(3);

$wa = 1; // 0 - wA.. has not applied. 1 - wA.. has applied.

} else { $wa =0; }

if (arr($text,'/[a][+][N]/') && in\_array($so,array("Nasi!")))

{

$text = one(array("a+Nasi!"),array("a+At"),0);

$text = two(array("jaras"),array("At"),array("jaras"),array("Nasi!"),0);

echo "<p class = sa >By TAGasiGasAminAtsyAH (7.1.12) :</p>";

echo "<p class = sa >टाङसिङसामिनात्स्याः (७.१.१२) :</p>";

display(3);

$wa1 = 1; // 0 - wA.. has not applied. 1 - wA.. has applied.

} else { $wa1 =0; }

### /\* Ami sarvanAmnaH suT (7.1.15) \*/

if ( $so === "Am" && $sarvafinal !== 0)

{

if ( $sarvafinal === 2)

{

$text = last(array("Am"),array("sAm"),1);

}

else

{

$text = last(array("Am"),array("sAm"),0);

}

echo "<p class = sa >By Ami sarvanAmnaH suT (7.1.15) :</p>";

echo "<p class = sa >आमि सर्वनाम्नः सुट्‌ (७.१.१५) :</p>";

display(3);

$sut=1; // 0 - Ami sarvanAmnaH sUT has not applied. 1 - Ami sarvanAmnaH suT has applied.

} else { $sut=0;}

### /\* sAma Akam (7.1.33) \*/

if (sub(array("asma","yuzma","ima","itva","Ava","yuva"),array("+"),array("sAm"),0) && in\_array($so,array("Am")))

{

$text = one(array("asma+sAm","yuzma+sAm","ima+sAm","itva+sAm","Ava+sAm","yuva+sAm"),array("asma+Akam","yuzma+Akam","ima+Akam","itva+Akam","Ava+Akam","yuva+Akam"),0);

echo "<p class = sa >By sAma Akam (7.1.33) :</p>";

echo "<p class = sa >साम आकम्‌ (७.१.३३) :</p>";

display(3);

$sAmaAkam=1; // 0 - sAma Akam has not applied. 1 - sAma Akam has applied.

} else { $sAmaAkam=0; }

### /\* dvitIyATaussvenaH (2.4.34) \*/

if (sub(array("ida+","eta+","idaka+","idA+",),blank(0),blank(0),0) && in\_array($fo,array("idam","etad","idakam")) && $anvadesha===1 && in\_array($so,array("am","Ow","Sas","wA","os")))

{

$text = one(array("ida+","eta+","idaka+","idA+",),array("ena+","ena+","ena+","enA+"),0);

echo "<p class = sa >By dvitIyATaussvenaH (2.4.34) :</p>";

echo "<p class = sa >द्वितीयाटौस्स्वेनः (२.४.३४) :</p>";

display(0);

}

### /\* idamo'nvAdeze'zanudAttastRtIyAdau (2.4.32) \*/

if (sub(array("idaka+"),blank(0),blank(0),0) && $fo==="idakam" && $anvadesha===1 && in\_array($so,$tRtIyAdi))

{

$text = one(array("idaka+"),array("a+"),0);

echo "<p class = sa >By idamo'nvAdeze'zanudAttastRtIyAdau (2.4.32) :</p>";

echo "<p class = sa >इदमोऽन्वादेशेऽशनुदात्तस्तृतीयादौ (२.४.३२) :</p>";

display(0);

}

### /\* nAnarthake'lo'ntyavidhiranabhyAsavikAre (vA 490) \*/

// Pending. Not clear to me.

### /\* Adyantavadekasmin \*/

// paribhASA. Difficult to code.

### /\* goto Nit (7.1.90) and oto Niditi vAcyam (vA 5035) \*/

if (sub(array("o+"),$sarvanamasthana,blank(0),0) && !in\_array($so,array("am","Sas")))

{

if (sub(array("go+"),$sarvanamasthana,blank(0),0))

{$Nidvat1 = 1;

echo "<p class = pa >By goto Nit (7.1.90) :</p>";

echo "<p class = pa >गोतो णित्‌ (७.१.९०) :</p>";

display(3);

}

elseif (!preg\_match('/[o]$/',$fo))

{$Nidvat1 = 0;

echo "<p class = pa >oto Niditi vAcyam (vA 5035) is overruled by 'okArAntAdvihitaM sarvanAmasthAnam'.</p>";

echo "<p class = pa >ओकारान्ताद्विहितं सर्वनामस्थानम्‌ इति व्याख्यानात्‌ ओतो णिदिति वाच्यमित्यस्य प्रवृत्तिर्नास्ति ।</p>";

display(0);

}

else

{$Nidvat1 = 1;

echo "<p class = pa >By oto Niditi vAcyam (vA 5035) :</p>";

echo "<p class = pa >ओतो णिदिति वाच्यम्‌ (वा ५०३५) :</p>";

display(3);

}

} else {$Nidvat1 =0; }

### /\* aco JNiti (7.2.115) \*/

if ((arr($text,'/['.flat($ac).'][+][JR]/')||arr($text,'/[a][+][\*][JR]$/')||$Nidvat===1||$Nidvat1===1) && arr($text,'/['.pc('ac').'][+]/') )

{

$text = two($ac,array("+"),vriddhi($ac),array("+"),0);

echo "<p class = sa >By aco JNiti (7.2.115) :</p>";

echo "<p class = sa >अचो ञ्णिति (७.२.११५) :</p>";

display(3);

}

if (arr($text,'/[iufx][+][j]/') && $so==="jas")

{

$text = two(array("i","u","f","x"),array("+"),array("e","o","ar","al"),array("+"),0);

echo "<p class = sa >By jasi ca (7.3.109) :</p>";

echo "<p class = sa >जसि च (७.३.१०९) :</p>";

display(3);

}

### /\* RRta iddhAtoH (7.1.100) \*/

$kRR = array("kF","tF","gF"); // Creating a list of RR-anta words. They are anukaraNas of dhAtus mostly. If additional forms are found, list them here.

if (arr($text,'/[ktg][F][+]/'))

{

$dhatu = 1;

}

if (arr($text,'/[ktg][F][+]/'))

{

$text = two(array("F"),array("+"),array("ir"),array("+"),1);

echo "<p class = sa >By RRta iddhAtoH (7.1.100) :</p>";

echo "<p class = hn >'prakRtivadanukaraNam' (pa 37) has vaikalpika atideza, which mandates itva here. As this is optional, itvAbhAva is also seen. </p>";

echo "<p class = sa >ॠत इद्धातोः (७.१.१००) :</p>";

echo "<p class = hn >'कॄ' 'तॄ' अनयोरनुकरणे 'प्रकृतिवदनुकरणम्‌-' (प ३७) इति वैकल्पिकातिदेशादित्त्वं रपरत्वं च । इत्त्वाभावपक्षे 'ऋदुशनस्‌-' इति 'ऋतो ङि-' इति च तपरकरणादनङ्गुणौ न ।</p>";

display(3);

}

if ( $dhatu === 1 && sub(array("F"),array("+"),blank(0),0) && !arr($text,'/[ktg][F][+]/') )

{

$text = two(array("F"),array("+"),array("ir"),array("+"),0);

echo "<p class = sa >By RRta iddhAtoH (7.1.100) :</p>";

echo "<p class = sa >ॠत इद्धातोः (७.१.१००) :</p>";

display(3);

}

### /\* ambArthanadyorhrasvaH (7.3.103) \*/

if (sub(array("ambAqA","ambAlA","ambikA"),array("+"),blank(0),0) && $sambuddhi===1 && $so==="su!")

{

echo "<p class = pa >asaMyuktA ye DalakAstadvatAM hrasvo na (vA 4592) :</p>";

echo "<p class = pa >असंयुक्ता ये डलकास्तद्वतां ह्रस्वो न (वा ४५९२) :</p>";

display(0);

}

if ($sambuddhi===1 && ($nadi!==0 || ends(array($fo),array("ambA","akkA","allA"),1)) && $so==="su!")

{

if ($nadi===1 || ends(array($fo),array("ambA","akkA","allA"),1))

{

$text = two(array("A","I","U"),array("+"),array("a","i","u"),array("+"),0);

}

else

{

$text = two(array("A","I","U"),array("+"),array("a","i","u"),array("+"),1);

}

echo "<p class = sa >By ambArthanadyorhrasvaH (7.3.103) :</p>";

echo "<p class = sa >अम्बार्थनद्योर्ह्रस्वः :</p>";

display(3);

$amba = 1; // 0 - ambArthanadyorhrasva has not applied. 1 - ambArthanadyorhrasva has applied.

} else { $amba = 0; }

### /\* idudbhAym (7.3.117) \*/

if ($nadi!==0 && arr($text,'/[iu][+][N][i]$/'))

{

$text = two(array("i","u"),array("+Ni"),array("i","u"),array("+Am"),1);

echo "<p class = sa >By idudbhAym (7.3.117) :</p>";

echo "<p class = sa >इदुद्भ्याम्‌ (७.३.११७) :</p>";

display(3);

}

### /\* GerAmnadyAmnIbhyaH (7.3.116) \*/

if (arr($text,'/[n][I][+]/') && $pada=== "pratyaya" && $so==="Ni")

{

$text = two(array("+"),array("Ni"),array("+"),array("Am"),0);

echo "<p class = sa >By GerAmnadyAmnIbhyaH (7.3.116) :</p>";

echo "<p class = sa >ङेराम्नद्याम्नीभ्यः (७.३.११६) :</p>";

display(3);

}

if ($nadi!==0 && $pada=== "pratyaya" && $so==="Ni")

{

if ($nadi===1)

{

$text = two(array("I","U"),array("+Ni"),array("I","U"),array("+Am"),0);

}

else

{

$text = two(array("I","U"),array("+Ni"),array("I","U"),array("+Am"),1);

}

echo "<p class = sa >By GerAmnadyAmnIbhyaH (7.3.116) :</p>";

echo "<p class = sa >ङेराम्नद्याम्नीभ्यः (७.३.११६) :</p>";

display(3);

}

if ($Ap===1 && $pada=== "pratyaya" && $so==="Ni")

{

$text = two(array("A"),array("+Ni"),array("A"),array("+Am"),0);

echo "<p class = sa >By GerAmnadyAmnIbhyaH (7.3.116) :</p>";

echo "<p class = sa >ङेराम्नद्याम्नीभ्यः (७.३.११६) :</p>";

display(3);

}

### /\* aut (7.3.118) \*/

if (arr($text,'/[iu][+][N][i]$/') && $so==="Ni" && $pada==="pratyaya")

{

$text = two(array("i","u",),array("Ni"),array("i","u"),array("O"),0);

echo "<p class = sa >By aut (7.3.118) :</p>";

echo "<p class = sa >औत्‌ (७.३.११८) :</p>";

display(3);

}

### /\* ANnadyAH (7.3.112) \*/

// The method for finding Git is coarse. needs fine tuning.

if ($nadi!==0 && arr($text,'/[+][N]/') && in\_array($so,array("Ne","Nasi!","Nas",)))

{

if ($nadi===1)

{

$text = two(array("+"),array("N"),array("+"),array("A+N"),0);

}

else

{

$text = two(array("+"),array("N"),array("+"),array("A+N"),1);

}

echo "<p class = sa >By ANnadyAH (7.3.112) :</p>";

echo "<p class = sa >आण्नद्याः (७.३.११२) :</p>";

display(3);

$ANnadyAH =1; // 0 - ANnadyAH has not applied. 1 - ANnadyAH has applied.

} else {$ANnadyAH = 0; }

### /\* acca gheH (7.3.119) \*/

if ($ghi===1 && in\_array($so,array("Ni")))

{

$text = two(array("i","u"),array("O"),array("a","a"),array("O"),0);

echo "<p class = sa >By acca gheH (7.3.119) :</p>";

echo "<p class = sa >अच्च घेः (७.३.११९) :</p>";

display(3);

}

### /\* gherGiti (7.3.111) \*/

if ($ghi===1 && $noghe===0 && arr($text,'/[iu][+]/') && in\_array($so,array("Ne","Nasi!","Nas")))

{

$text = two(array("i","u"),array("Ne","Nasi!","Nas"),array("e","o"),array("Ne","Nasi!","Nas"),0);

echo "<p class = sa >By gherGiti (7.3.111) :</p>";

echo "<p class = sa >घेर्ङिति (७.३.१११) :</p>";

display(3);

}

### /\* auGaH ApaH (7.1.18) \*/

if ($Ap===1 && arr($text,'/[A][+][O]$/') && in\_array($so,array("O")))

{

$text = two(array("A+"),array("O"),array("A+"),array("SI",),0);

echo "<p class = sa >By auGa ApaH (7.1.18) :</p>";

echo "<p class = sa >औङ आपः (७.१.१८) :</p>";

display(3);

}

if ($Ap===1 && arr($text,'/[A][+][O][w]$/') && in\_array($so,array("Ow")))

{

$text = two(array("A+"),array("Ow"),array("A+"),array("SI"),0);

echo "<p class = sa >By auGaH ApaH (7.1.18) :</p>";

echo "<p class = sa >औङ आपः (७.१.१८) :</p>";

display(3);

}

### /\* sambuddhau ca (7.3.106) \*/

if ($Ap===1 && $sambuddhi===1 && in\_array($so,array("su!")) && arr($text,'/[A][+]/'))

{

$text = two(array("A+"),array("su!"),array("e+"),array("su!"),0);

echo "<p class = sa >By sambuddhau ca (7.3.106) :</p>";

echo "<p class = sa >सम्बुद्धौ च (७.३.१०६) :</p>";

display(3);

}

### /\* sarvanAmnaH syADDhrasvazca (7.3.114) \*/

$sarvanamastri = array("sarvA","viSvA","uBA","uBayA","atarA","atamA","anyA","anyatarA","itarA","tvA","nemA","simA","pUrvA","parA","avarA","dakziRA","uttarA","aparA","aDarA","svA","antarA","tyA","tA","yA","etA","idA","adA","ekA","dvA","kA","idakA");

$sarvanamastri1 = array("sarva","viSva","uBa","uBaya","atara","atama","anya","anyatara","itara","tva","nema","sima","pUrva","para","avara","dakziRa","uttara","apara","aDara","sva","antara","tya","ta","ya","eta","ida","ada","eka","dva","ka","idaka");

if ($Ap===1 && $sarvafinal !==0 && in\_array($so,array("Ne","Nas","Ni")) && ( sub($sarvanamastri,blank(0),blank(0),0) ||in\_array($fo,array("idam")) ))

{

if ( $sarvafinal === 2)

{

$text = one(array("A+Ne","A+Nas","A+Am"),array("a+syA+Ne","a+syA+Nas","a+syA+Am"),1);

}

else

{

$text = one(array("A+Ne","A+Nas","A+Am"),array("a+syA+Ne","a+syA+Nas","a+syA+Am"),0);

}

echo "<p class = sa >By sarvanAmnaH syADDhrasvazca (7.3.114) :</p>";

echo "<p class = sa >सर्वनाम्नः स्याड्ढ्रस्वश्च (७.३.११४) :</p>";

display(3);

$syaddhrasva = 1; // 0 - sarvanAmnaH syADDhrasvazca has not applied. 1 - sarvanAmnaH syADDhrasvazca has applied.

} else { $syaddhrasva = 0; }

if ($Ap===1 && $sarvafinal !==0 && in\_array($so,array("Nasi!")) && ( sub($sarvanamastri,blank(0),blank(0),0) ||in\_array($fo,array("idam"))) )

{

if ( $sarvafinal === 2)

{

$text = one(array("A+Nasi!"),array("a+syA+Nasi!"),1);

}

else

{

$text = one(array("A+Nasi!"),array("a+syA+Nasi!"),0);

}

echo "<p class = sa >By sarvanAmnaH syADDhrasvazca (7.3.114) :</p>";

echo "<p class = sa >सर्वनाम्नः स्याड्ढ्रस्वश्च (७.३.११४) :</p>";

display(3);

$syaddhrasva1 = 1; // 0 - sarvanAmnaH syADDhrasvazca has not applied. 1 - sarvanAmnaH syADDhrasvazca has applied.

} else {$syaddhrasva1=0; }

### /\* hali lopaH (7.2.113) \*/

if (sub(array("ida+","idA+"),$hl,blank(0),0) && $fo==="idam" && !in\_array($so,array("jas","Ow","O","Sas","wA","os")))

{

$text = one(array("ida+","idA+",),array("a+","A+",),0);

echo "<p class = sa >By hali lopaH (7.2.113) :</p>";

echo "<p class = sa >हलि लोपः (७.२.११३) :</p>";

display(3);

}

### /\* anApyakaH (7.2.112) \*/

if (sub(array("ida+","idA+"),blank(0),blank(0),0) && $fo==="idam" && in\_array($so,$tRtIyAdiSvaci))

{

$text = one(array("ida+","idA+"),array("ana+","anA+"),0);

echo "<p class = sa >By anApyakaH (7.2.112) :</p>";

echo "<p class = sa >अनाप्यकः (७.२.११२) :</p>";

display(3);

}

### /\* AGi cApaH (7.3.105) \*/

if ($Ap===1 && in\_array($so,array("wA","os")))

{

$text = two(array("A+"),array("wA","os"),array("e+"),array("wA","os"),0);

echo "<p class = sa >By AGi cApaH (7.3.105) :</p>";

echo "<p class = sa >आङि चापः (७.३.१०५) :</p>";

display(3);

}

### /\* vibhASA diksamAse bahuvrIhau (1.1.28) \*/

if ($Ap===1 && in\_array($so,array("Ne","Nas","Nasi!","Ni")) && in\_array(array($fo),$diksamAsa))

{

echo "<p class = pa >vibhASA diksamAse bahuvrIhau (1.1.28) </p>";

echo "<p class = pa >विभाषा दिक्समासे बहुव्रीहौ (१.१.२८) </p>";

display(0);

$sarvafinal = 2;

}

### /\* yADApaH (7.3.113) \*/

if ($Ap===1 && (( in\_array($so,array("Ne","Nas","Ni")) && $syaddhrasva===0 ) || ( in\_array($so,array("Ne","Nas","Ni")) && $sarvafinal===2 ) ))

{

$text = three(array("A+"),array(""),array("Ne","Nas","Ni","Am"),array("A+"),array("yA+"),array("Ne","Nas","Ni","Am"),0);

$text = one(array("syA+yA+"),array("syA+"),0);

echo "<p class = sa >By yADApaH (7.3.113) :</p>";

echo "<p class = sa >याडापः (७.३.११३) :</p>";

display(3);

}

if ($Ap===1 && (( in\_array($so,array("Nasi!")) && $syaddhrasva===0 ) || ( in\_array($so,array("Nasi!")) && $sarvafinal===2 ) ))

{

$text = three(array("A+"),array(""),array("Nasi!"),array("A+"),array("yA+"),array("Nasi!"),0);

$text = one(array("syA+yA+"),array("syA+"),0);

echo "<p class = sa >By yADApaH (7.3.113) :</p>";

echo "<p class = sa >याडापः (७.३.११३) :</p>";

display(3);

}

### /\* nedamadasorakoH (7.1.11) \*/

if (arr($text,'/^[a][+]/') && $so==="Bis" && in\_array($fo,array("idam","idakam","adas")))

{

echo "<p class = sa >By nedamadasorakoH (7.1.11) :</p>";

echo "<p class = sa >नेदमदसोरकोः (७.२.११) :</p>";

display(3);

$nedamadas=1; // 0 - nedamadasorakoH doesn't prevent application of ato bhisa ais. 1 - nedamadasorakoH prevents application of ato bhisa ais.

} else { $nedamadas=0; }

### /\* jasaH zI (7.1.17) \*/

if (arr($text,'/[a][+]/') && $pada=== "pratyaya" && $so === "jas" && ends(array($fo),$sarvanama,1) && $nojas===0 && $sarvafinal!==0 )

{

if ($sarvafinal===2)

{

$text = last(array("jas"),array("SI"),1);

}

else

{

$text = last(array("jas"),array("SI"),0);

}

echo "<p class = sa >By jasaH zI (7.1.17) :</p>";

echo "<p class = sa >जसः शी (७.१.१७) :</p>";

echo "<p class = hn >N.B. anekAlzitsarvasya mandates sarvAdeza :</p>";

echo "<p class = hn >अनेकाल्शित्सर्वस्य से सर्वादेश होता है । :</p>";

display(3);

$sarva2 =1; // 0 - jasaH zI has not happened. 1 - jasaH zI has happened.

} else { $sarva2 = 0; }

### /\* pUrvaparAvaradakSiNottarAparAdharANi vyavasthAyAmasaJjJAyAm (1.1.34) \*/

if ($so === "jas" && $purvapara===1 && in\_array($fo,array("pUrva","para","avara","dakziRa","uttara","apara","aDara",)))

{

$text = last(array("SI"),array("jas"),1);

echo "<p class = sa >By pUrvaparAvaradakSiNottarAparAdharANi vyavasthAyAmasaJjJAyAm (1.1.34) :</p>";

echo "<p class = sa >पूर्वपरावरदक्षिणोत्तरापराधराणि व्यवस्थायामसंज्ञायाम्‌ (१.१.३४) :</p>";

display(0);

$purva=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else {$purva=0;}

### /\* svamajJAtidhanAkhyAyAm (1.1.35) \*/

if ($so === "jas" && $sva===1 && in\_array($fo,array("sva",)))

{

$text = last(array("SI"),array("jas"),1);

echo "<p class = sa >By svamajJAtidhanAkhyAyAm (1.1.35) :</p>";

echo "<p class = sa >स्वमज्ञातिधनाख्यायाम्‌ (१.१.३५) :</p>";

display(0);

}

### /\* antaraM bahiryogopasaMvyAnayoH (1.1.36) \*/

if ($so === "jas" && $antara===1 && in\_array($fo,array("antara",)))

{

$text = last(array("SI"),array("jas"),1);

echo "<p class = sa >By antaraM bahiryogopasaMvyAnayoH (1.1.36) :</p>";

echo "<p class = sa >अन्तरं बहिर्योगोपसंव्यानयोः (१.१.३६) :</p>";

display(0);

}

### /\* vibhASA jasi \*/

### /\* na bahuvrIhau \*/

### /\* tRtIyAsamAse \*/

### /\* dvandve ca \*/

### /\* prathamacaramatayAlpArdhakatipayanemAzca (1.1.33) \*/

if ($so === "jas" && in\_array($fo,array("praTama","carama","alpa","arDa","katipaya")))

{

$text = last(array("jas"),array("SI"),1); // function last removes and replaces a specified string from the end of word. for details, see function.php.

echo "<p class = sa >By prathamacaramatayAlpArdhakatipayanemAzca (1.1.33) :</p>";

echo "<p class = sa >प्रथमचरमतयाल्पार्धकतिपयनेमाश्च (१.१.३३) :</p>";

display(0);

}

if ($so === "jas" && in\_array($fo,array("nema")))

{

$text = last(array("SI"),array("jas"),1);

echo "<p class = sa >By prathamacaramatayAlpArdhakatipayanemAzca (1.1.33) :</p>";

echo "<p class = sa >प्रथमचरमतयाल्पार्धकतिपयनेमाश्च (१.१.३३) :</p>";

display(0);

}

### /\* dvandve ca (1.1.31) \*/

if ($so === "jas" && $dvandveca===1 )

{

$text = last(array("jas"),array("SI"),1);

echo "<p class = sa >By vibhASA jasi (1.1.32) :</p>";

echo "<p class = sa >विभाषा जसि (१.१.३२) :</p>";

display(0);

}

if ($so === "jas" && arr(array($fo),'/(taya)$/'))

{

$text = last(array("jas"),array("SI"),1);

echo "<p class = sa >By prathamacaramatayAlpArdhakatipayanemAzca (1.1.33) :</p>";

echo "<p class = sa >प्रथमचरमतयाल्पार्धकतिपयनेमाश्च (१.१.३३) :</p>";

display(0);

}

### /\* upadeze'janunAsika it (1.3.2)\*/ // Temporary patch. Not coded perfectly.

if (arr($text,'/['.flat($ac).'][!]/') )

{

it('/(['.flat($ac).'][!])/');

echo "<p class = pa >By upadeze'janunAsika it (1.3.2) :</p>";

echo "<p class = pa >उपदेशेऽजनुनासिक इत्‌ (१.३.२) :</p>";

display(0);

$text = two($ac,array("!"),blank(count($ac)),array(""),0);

echo "<p class = sa >By tasya lopaH (1.3.9) :</p>";

echo "<p class = sa >तस्य लोपः (१.३.९) :</p>";

display(0);

}

### /\* patch for saMyogAntalopa of maghavan \*/

if (sub(array("maGavant",),blank(0),blank(0),0) && in\_array($so,array("su!")) && $sambuddhi===0)

{

$text = two(array("maGavant",),array("+"),array("maGavan",),array("+"),0);

echo "<p class = sa >By saMyogAntasya lopaH (8.2.23) :</p>";

echo "<p class = hn >Because of bahulagrahaNa in maghavan, saMyogAntasya lopaH is not asiddha here. :</p>";

echo "<p class = sa >संयोगान्तस्य लोपः (८.२.२३) :</p>";

echo "<p class = hn >मघवा बहुलं के बहुलग्रहण के कारण, संयोगान्तलोप असिद्ध नहीं है । </p>";

display(0);

}

### /\* AdirGiTuDavaH (1.3.5) \*/

if ((substr($first,0,2) === "Yi" || substr($first,0,2) === "wu" || substr($first,0,2) === "qu") && $pada=== "pratyaya")

{

if(substr($first,0,2) === "Yi") { $itprakriti = array\_merge($itprakriti,array("Yi")); }

if(substr($first,0,2) === "wu") { $itprakriti = array\_merge($itprakriti,array("wu")); }

if(substr($first,0,2) === "qu") { $itprakriti = array\_merge($itprakriti,array("qu")); }

echo "<p class = pa >By AdirGiTuDavaH (1.3.5) :</p>";

echo "<p class = pa >आदिर्ञिटुडवः (१.३.५) :</p>";

display(0);

$text = first(array("Yi","wu","qu"),array("","",""),0); // function first removes and replaces specific strings from the words. For details see function.php.

echo "<p class = sa >tasya lopaH (1.3.9) :</p>";

echo "<p class = sa >तस्य लोपः (१.३.९) :</p>";

display(0);

}

### /\* cuTU (1.3.7) \*/

if (arr($text,'/[+][cCjJYwWqQR]/') && $wa1 === 0 && ($sarva2 ===0 || $purva=1))

{

it('/([+][cCjJYwWqQR])/');

echo "<p class = pa >By cuTU (1.3.7) :</p>";

echo "<p class = pa >चुटू (१.३.७) :</p>";

display(0);

$text = last(array("jas","wA"),array("as","A"),0);

echo "<p class = sa >tasya lopaH (1.3.9) :</p>";

echo "<p class = sa >तस्य लोपः (१.३.९) :</p>";

display(0);

}

### /\* SaH pratyayasya (1.3.6) \*/

if (arr($text,'/[+][z]/') && $pada=== "pratyaya")

{

it('/([+][z])/');

echo "<p class = pa >By SaH pratyayasya (1.3.6) :</p>";

echo "<p class = pa >षः प्रत्ययस्य (१.३.६) :</p>";

display(0);

$text = two(array("+"),array("z"),array("+"),array(""),0);

echo "<p class = sa >By tasya lopaH (1.3.9) :</p>";

echo "<p class = sa >तस्य लोपः (१.३.९) :</p>";

display(0);

}

### /\* GeryaH (7.1.13) \*/

if (arr($text,'/[a][+][N][e]/') && $pada=== "pratyaya" && $so === "Ne" )

{

$text = one(array("a+Ne"),array("a+ya"),0);

$text = two(array("jaras"),array("ya"),array("jaras"),array("Ne"),0);

echo "<p class = sa >By GeryaH (7.1.13) :</p>";

echo "<p class = sa >ङेर्यः (७.१.१३) :</p>";

display(3);

$Ne=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $Ne = 0; }

if (in\_array($fo,array("SrIpA")) && $gender==="n" && $so==="Ne")

{

echo "<p class = pa >By sannipAtaparibhASA prevents application of 'Ato dhAtoH'.</p>";

echo "<p class = pa >सन्निपातपरिभाषा आतो धातोः का प्रयोग निषेध करती है ।</p>";

display(0);

}

### /\* lazakvataddhite (1.3.8) \*/

if (((arr($text,'/[+][lSkKgGN]/'))||$sarva2===1||$purva===1) && $taddhita === 0 && $sarva === 0 )

{

it('/([+][lSkKgGN])/');

echo "<p class = pa >By lazakvataddhite (1.3.8) :</p>";

echo "<p class = pa >लशक्वतद्धिते (१.३.८) :</p>";

display(0);

$text = last(array("Sas","Ni","SI","Nas","Ne","Si","kvin"),array("as","i","I","as","e","i","vin"),0);

echo "<p class = sa >By tasya lopaH (1.3.9) :</p>";

echo "<p class = sa >तस्य लोपः (१.३.९) :</p>";

display(0);

}

### /\* na vibhaktau tusmAH (1.3.4) \*/

if (arr($text,'/[tTdDnsm]$/') && $pada=== "pratyaya" && in\_array($so,array("jas","am","Sas","ByAm","Bis","Byas","os","Am","Ni")) && $wa === 0 && $wa1 === 0)

{

echo "<p class = pa >By na vibhaktau tusmAH (1.3.4) :</p>";

echo "<p class = pa >न विभक्तौ तुस्माः (१.३.४) :</p>";

display(0);

}

### /\* halantyam (1.3.3) and tasya lopaH \*/

if ((arr($text,'/['.flat($hl).']$/') && $pada=== "pratyaya" && !in\_array($so,$sup)) || (in\_array($so,$sup) && !arr($text,'/[tTdDnsm]$/') && arr($text,'/['.pc('hl').']$/')))

{

itprat('/(['.flat($hl).']$)/');

echo "<p class = pa >By halantyam (1.3.3) :</p>";

echo "<p class = pa >हलन्त्यम्‌ (१.३.३) :</p>";

display(0);

$text = last(prat('hl'),blank(count(prat('hl'))),0);

echo "<p class = sa >By tasya lopaH (1.3.9) :</p>";

echo "<p class = sa >तस्य लोपः (१.३.९) :</p>";

display(0);

}

### /\* it additions in case of kvin and kvip \*/

if ($kvin===1)

{

$itpratyaya=array\_merge($itpratyaya,array("k","n","i")); // still ambiguous whether these should be added to pratyayas only or sometimes to prakRtis also.

$it=array\_merge($it,array("k","n","i"));

}

if ($kvip===1)

{

$itpratyaya=array\_merge($itpratyaya,array("k","p","i"));

$it=array\_merge($it,array("k","p","i"));

}

### /\* nAJceH pUjAyAm (6.4.30) \*/

//if ( !itcheck(array("i"),1) && arr($text,'/[n]['.pc('hl').'][+]/') && ( itcheck(array("k","G"),2)|| $ancu===1 )) // blocked because gives wrong result in 'yuj'->'yunj'.

if ( !itcheck(array("i"),1) && arr($text,'/[n][c][+]/') && $nance===1 ) // for function itcheck please see function.php.

{

echo "<p class = sa >nAJceH pUjAyAm (6.4.30) :</p>";

echo "<p class = sa >नाञ्चेः पूजायाम्‌ (६.४.३०) :</p>";

display(0);

}

### /\* kruJca \*/

if ( sub(array("kruYc","krunc"),array("+"),blank(0),0) )

{

echo "<p class = pa >'RtvigdadhRksragdigaJcuyujikruJcAM ca' prohibits application of nalopa.</p>";

echo "<p class = pa >'ऋत्विग्दधृक्स्रग्दिगञ्चुयुजिक्रुञ्चां च' इत्यनेन नलोपाभावोऽपि निपात्यते ।</p>";

display(0); $kruJca=1;

} else { $kruJca = 0; }

### /\* defining dhatus with 'i' as it \*/

if (sub(array("hiMs"),array("+"),blank(0),0))

{

$it=array\_merge($it,array("i"));

$itprakriti=array\_merge($it,array("i"));

}

### /\* aniditAM hala upadhAyAH kGiti (6.4.24) \*/

if ( !itcheck(array("i"),1) && arr($text,'/[nM]['.pc('hl').'][+]/') && ( itcheck(array("k","N"),1)|| $ancu===1 ) && $kruJca===0 && $nance===0 )

{

$text = three(array("n","M"),$hl,array("+"),array("",""),$hl,array("+"),0);

echo "<p class = sa >aniditAM hala upadhAyAH kGiti (6.4.24) :</p>";

echo "<p class = sa >अनिदितां हल उपधायाः क्ङिति (६.४.२४) :</p>";

display(0);

$aniditAm = 1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $aniditAm = 0; }

### /\* samaH sami (6.3.93) \*/

if (preg\_match('/[s][a][m][a][n][c]/',$fo) && sub(array("anc","Anc"),array("+"),blank(0),0) && $ancu===1)

{

$text = two(array("samac"),array("+"),array("samiac"),array("+"),0);

echo "<p class = sa >samaH sami (6.3.93) :</p>";

echo "<p class = sa >समः समि (६.३.९३) :</p>";

display(0);

}

### /\* sahasya sadhriH (6.3.95) \*/

if (preg\_match('/[s][a][h][a][a][n][c]/',$fo) && sub(array("anc","Anc"),array("+"),blank(0),0) && $ancu===1)

{

$text = two(array("sahaac"),array("+"),array("saDriac"),array("+"),0);

echo "<p class = sa >sahasya sadhriH (6.3.95) :</p>";

echo "<p class = sa >सहस्य सध्रिः (६.३.९५) :</p>";

display(0);

}

### /\* tirasastiryalope (6.3.94) \*/

if (preg\_match('/[t][i][r][a][s][a][n][c]/',$fo) && sub(array("anc","Anc","ac","Ac"),array("+"),blank(0),0) && $bham===0 && $ancu===1)

{

$text = two(array("tirasac","tirasanc"),array("+"),array("tiriac","tirianc"),array("+"),0);

echo "<p class = sa >tirasastiryalope (6.3.94) :</p>";

echo "<p class = sa >तिरसस्तिर्यलोपे (६.३.९४) :</p>";

display(0);

}

### /\* atvasantasya cAdhAtoH (6.4.14) \*/

if ($atu==1 && $dhatu===1 )

{

echo "<p class = pa >cAdhAtoH means that atvasantasya cAdhAtoH (6.4.14) will not apply. :</p>";

echo "<p class = pa >चाधातोः से अत्वसन्तस्य चाधातोः (६.४.१४) का प्रयोग बाधित होता है । :</p>";

display(0);

}

elseif ( ( $atu===1 ) && $so==="su!" && $sambuddhi===0 )

{

$text = two($ac,array("t+"),$acdir,array("t+"),0);

$text = one(array("as+"),array("As+"),0);

echo "<p class = sa >atvasantasya cAdhAtoH (6.4.14) :</p>";

echo "<p class = sa >अत्वसन्तस्य चाधातोः (६.४.१४) :</p>";

display(3);

}

### /\* ugidacAM sarvanAmasthAne'dhAtoH (7.1.70) \*/

if ($atu===1 && !itcheck(array("u"),1) && $sarvanamasthana1===1 )

{

echo "<p class = pa >Because the word is not ugit; ugidacAM doesn't apply. </p>";

echo "<p class = pa >उगितत्त्वाभावान्न नुम्‌ ।</p>";

display(0);

}

if ($sarvanamasthana1===1 && ( ($ancu===0 && $dhatu===1) ) && $kruJca===0 && arr(array($fo),'/[a][t]$/'))

{

echo "<p class = pa >'ac' is for restricting the application of 'ugidacAm...' to aYcu if the word is a dhAtu. </p>";

echo "<p class = pa >धातोश्चेदुगित्कार्यं तर्ह्यञ्चतेरेव ।</p>";

display(0);

}

//if ($sarvanamasthana1===1 && $nAbhyasta===0 && (( $dhatu===0 && arr($text,'/[uUfFxX][!][+]/') ) || ($ancu===1 && $dhatu===1) || ( ($kvip===1 || $kvin===1)&& $dhatu===1 && arr(array($fo),'/[a][t]$/')) || $bhavat===1 ) && $kruJca===0 && $vAnapuMsaka===0)

if ($sarvanamasthana1===1 && (( $dhatu===0 && itcheck(array("u","U","f","F","x","X"),1)) || ($ancu===1 && $kvin===1) || ( ($kvip===1 || $kvin===1)&& $dhatu===1 && arr(array($fo),'/[a][t]$/')) || $bhavat===1 ) )

{

$text = one(array("ac+","Ac+"),array("anc+","Anc+"),0);

$text = three(blank(1),$hl,array("u!+","U!+","f!+","F!+","x!+"),array("n"),$hl,array("u!+","U!+","f!+","F!+","x!+"),0); $num=array\_merge($num,array(1));

$text = mit('/['.pc('hl').']/',"n",0);

$text = one(array("annc"),array("anc"),0);

echo "<p class = sa >By ugidacAM sarvanAmasthAne'dhAtoH (7.1.70) :</p>";

echo "<p class = sa >उगिदचां सर्वनामस्थानेऽधातोः (७.१.७०) :</p>";

if ( ($kvip===1 || $kvin===1)&& $dhatu===1)

{

echo "<p class = hn >'adhAtoH' extends the application of this rule to the words which had adhAtu in the first place.</p>";

echo "<p class = hn >'अधातोः' इति त्वधातुभूतपूर्वस्यापि नुमर्थम्‌ ।</p>";

}

if ($nance===1)

{

$text = one(array("goannc"),array("goanc"),0); // patch to remove duplication of nakAra.

echo "<p class = hn >As there is no lopa of nakAra, 'ugidacAm..' doesn't apply.</p>";

echo "<p class = hn >अलुप्तनकारत्वात्‌ न नुम्‌ ।</p>";

}

display(3);

$ugidacAm=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else {$ugidacAm=0; }

### /\* uda It (6.4.139) \*/

if (preg\_match('/[u][d][a][n][c]/',$fo) && $aniditAm === 1 && sub(array("ac","Ac"),array("+"),blank(0),0) && $bham===1 && $ancu===1)

{

$text = two(array("ac","Ac"),array("+"),array("Ic","Ic"),array("+"),0);

echo "<p class = sa >uda It (6.4.139) :</p>";

echo "<p class = sa >उद इत्‌ (६.४.१३९) :</p>";

display(0);

}

### /\* acaH (6.4.138) \*/

if ( preg\_match('/[aA][n][c]/',$fo) && $aniditAm === 1 && sub(array("ac","Ac"),array("+"),blank(0),0) && $bham===1 && $ancu===1)

{

if (sub(array("i","I","u","U","f","F","x","X","y","v"),prat('ac'),array("c"),0))

{

echo "<p class = pa >Though iko yaNaci is antaraGga than lopa by acaH, its application is barred by 'akRtavyUhAH pANinIyAH (pa 57).</p>";

echo "<p class = pa >इको यणचि से प्राप्त यण्‌ अन्तरङ्ग होने पर भी अकृतव्यूहाः पाणिनीयाः (प ५७) से वह अचः का बाध नहीं करता ।</p>";

display(0);

}

$text = two(array("yac","ac","Ac"),array("+"),array("ic","c","ac"),array("+"),0);

echo "<p class = sa >acaH (6.4.138) :</p>";

echo "<p class = sa >अचः (६.४.१३८) :</p>";

if ($nance===1)

{

echo "<p class = hn >As there is no lopa of nakAra in nAJceH pUjAyAm, there is not akAralopa.</p>";

echo "<p class = hn >नलोपाभावादकारलोपो न ।</p>";

}

display(3);

$acaH=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $acaH=0; }

### /\* cau (6.3.138) \*/

if ( $acaH===1)

{

$text = three($ac,array("c","c"),array("+"),$acdir,array("c","c"),array("+"),0);

echo "<p class = sa >cau (6.3.138) :</p>";

echo "<p class = sa >चौ (६.३.१३८) :</p>";

display(3);

}

### /\* shatR \*/

if ($shatR===1)

{

$it=array\_merge($it,array("f"));

$itprakriti=array\_merge($itprakriti,array("f"));

}

### /\* nAbhyAsAcChatuH (7.1.78) \*/

if ($abhyasta===1 && $shatR===1 && itcheck(array("f"),1) && $gender!=="n")

{

echo "<p class = sa >By nAbhyastAcChatuH (7.1.78) :</p>";

echo "<p class = sa >नाभ्यास्ताच्छतुः (7.1.78) :</p>";

display(0);

$nAbhyasta=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $nAbhyasta=0; }

### /\* vA napuMsakasya (7.1.79) \*/

if ($abhyasta===1 && $shatR===1 && itcheck(array("f"),1) && $gender==="n")

{

$text = mit('/['.pc('hl').'][+]/','n',1); $num=array\_merge($num,array(1));

$text = one(array("annc"),array("anc"),0);

echo "<p class = sa >By vA napuMsakasya (7.1.79) :</p>";

echo "<p class = sa >वा नपुंसकस्य (७.१.७९) :</p>";

display(0);

$vAnapuMsaka=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $vAnapuMsaka=0; }

### /\* zapzyanornityam (7.1.81) \*/

if (arr($text,'/[aA][t][+[S][I]/') && ($shap===1|| $shyan===1 ) )

{

$text = mit('/['.pc('hl').'][+]/','n',0); $num=array\_merge($num,array(1)); // function mit adds a mit Agama according to midaco'ntyAtparaH. See function.php

$text = one(array("annc"),array("anc"),0);

echo "<p class = sa >By zapzyanornityam (7.1.81) :</p>";

echo "<p class = sa >शप्श्यनोर्नित्यम्‌ (७.१.८१) :</p>";

display(0);

}

### /\* AcChInadyornum (7.1.80) \*/

if (arr($text,'/[aA][t][+[S][I]/') && $shatR===1 && itcheck(array("f"),1) && $AcCInadyo===1 )

{

$text = mit('/['.pc('hl').'][+]/','n',1); $num=array\_merge($num,array(1));

$text = one(array("annc"),array("anc"),0);

echo "<p class = sa >By AcChInadyornum (7.1.80) :</p>";

echo "<p class = sa >आच्छीनद्योर्नुम्‌ (७.१.८०) :</p>";

display(0);

}

### /\* special message for ikAra of kvip, kvin etc. \*/

if (arr($text,'/[+][v][i]$/')&& in\_array($so,array("kvip","kvin")) && $taddhita === 0 && $sarva === 0 )

{

$text = last(array("vi"),array("v"),0);

echo "<p class = sa >ikAra after 'v' in kvip, kvin etc are for fecility of pronounciation only.</p>";

echo "<p class = sa >क्विप्‌, क्विन्‌ इत्यादि में वकार के बाद का इकार उच्चारणार्थ ही होता है ।</p>";

display(0);

}

### /\* verapRktasya (3.1.67) \*/

if (arr($text,'/[+][v]$/')&& in\_array($so,array("kvip","kvin")) && $taddhita === 0 && $sarva === 0 )

{

$text = last(array("v"),array(""),0);

echo "<p class = sa >verapRktasya (3.1.67) :</p>";

echo "<p class = sa >वेरपृक्तस्य (३.१.६७) :</p>";

display(0);

}

### /\* GasiGasozca (6.1.110) \*/

if (arr($text,'/[eo][+]/') && in\_array($so,array("Nasi!","Nas")))

{

$text = two(array("e+","o+"),array("a"),array("e+","o+"),array(""),0);

echo "<p class = sa >By GasiGasozca (6.1.110) :</p>";

echo "<p class = sa >ङसिङसोश्च (६.१.११०) :</p>";

display(0);

}

### /\* hrasvanadyApo nuT (7.1.54) \*/

if ( $so === "Am" && $numacira===0 && arr($text,'/[aiufx][+][A][m]/') )

{

$text = two($hrasva,array("+Am"),$hrasva,array("+nAm"),0);

echo "<p class = sa >By hrasvanadyApo nuT (7.1.54) :</p>";

echo "<p class = sa >ह्रस्वनद्यापो नुट्‌ (७.१.५४) :</p>";

display(3);

}

if ( $so === "Am" && $numacira===0 && $nadi!==0 && arr($text,'/[IU][+][A][m]/') )

{

if ($nadi===2)

{

$text = two(array("I","U"),array("+Am"),array("I","U"),array("+nAm"),1);

}

else

{

$text = two(array("I","U"),array("+Am"),array("I","U"),array("+nAm"),0);

}

echo "<p class = sa >By hrasvanadyApo nuT (7.1.54) :</p>";

echo "<p class = sa >ह्रस्वनद्यापो नुट्‌ (७.१.५४) :</p>";

display(3);

}

if ( $so === "Am" && $numacira===0 && $Ap===1 && arr($text,'/[A][+][A][m]/'))

{

$text = two(array("A"),array("+Am"),array("A"),array("+nAm"),0);

echo "<p class = sa >By hrasvanadyApo nuT (7.1.54) :</p>";

echo "<p class = sa >ह्रस्वनद्यापो नुट्‌ (७.१.५४) :</p>";

display(3);

}

### /\* bahUrji exception to napuMsakasya jhalacaH (7.1.72) \*/

if ( $gender==="n" && $sarvanamasthana1===1 && sub(array("bahUrj"),blank(0),blank(0),0) && arr($text,'/['.pc('Jl').'][+]/') && $amipUrva === 0)

{

$text = two(array("bahUrj"),array("+"),array("bahUrnj"),array("+"),1);

echo "<p class = sa >By bahUrji numpratiSedhaH (vA 4331) and antyAtpUrvo vA num (vA 4332) :</p>";

echo "<p class = sa >बहूर्जि नुम्प्रतिषेधः (वा ४३३१) तथा अन्त्यात्पूर्वो वा नुम् (वा ४३३२) :</p>";

display(3);

$bahurj=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $bahurj=0; }

### /\* beBid, cecCid exception to napuMsakasya jhalacaH (7.1.72) \*/

if ( $gender==="n" && $sarvanamasthana1===1 && $kvip===1 && sub(prat('Jl'),array("+"),blank(0),0) && in\_array($so,array("jas","Sas")) && arr($text,'/['.pc('Jl').'][+]/') && $amipUrva === 0)

{

echo "<p class = pa >allopa behaves like sthAnivad, therefore num (because of jhalantatva) doesn't happen. sthAnivadbhAva doesn't happen in svavidhi. Therefore num (because of ajantatva) doesn't happen. </p>";

echo "<p class = pa >शावल्लोपस्य स्थानिवत्त्वादझलन्तत्वान्न नुम्‌ । अजन्तलक्षणस्तु नुम्‌ न । स्वविधौ स्थानिवत्त्वाभावात्‌ । </p>";

display(0); $bebhid=1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $bebhid=0; }

### /\* svap patch \*/

if ( $gender==="n" && $sarvanamasthana1===1 && arr($text,'/[A][n][p][+]/') && arr(array($fo),'/[a][p]$/') && $amipUrva === 0 && $nAbhyasta===0)

{

$text = two(array("Anp"),array("+"),array("anp"),array("+"),1);

echo "<p class = sa >'niravakAzatvaM pratipadoktatvam' mandates non application of dIrghatva here. :</p>";

echo "<p class = sa >'निरवकाशं प्रतिपदोक्तत्वम्‌' इति पक्षे तु प्रकृते तद्विरहात्‌ नुमेव । :</p>";

display(3);

}

### /\* napuMsakasya jhalacaH (7.1.72) \*/

// right now supuMs is giving ugidacAM and not napuMsakasya jhalacaH. Pending to correct.

if ( $gender==="n" && $sarvanamasthana1===1 && arr($text,'/['.pc('ac').'][+]/') && $amipUrva === 0 && $nAbhyasta===0)

{

$text = mit('/['.pc('ac').'][+]/','n',0);

echo "<p class = sa >By napuMsakasya jhalacaH (7.1.72) :</p>";

echo "<p class = sa >नपुंसकस्य झलचः (७.१.७२) :</p>";

display(3);

}

if ( $gender==="n" && $sarvanamasthana1===1 && arr($text,'/['.pc('Jl').'][+]/') && $amipUrva === 0 && $bahurj===0 && $bebhid===0 && $ugidacAm===0 && $nAbhyasta===0 && $vAnapuMsaka===0)

{

$text = mit('/['.pc('Jl').'][+]/','n',0);

echo "<p class = sa >By napuMsakasya jhalacaH (7.1.72) :</p>";

echo "<p class = sa >नपुंसकस्य झलचः (७.१.७२) :</p>";

display(3);

}

### /\* sarvanAmasthAne cAsambuddhau (6.4.8) \*/

$acdir = array("A","A","I","I","U","U","F","F","F","F","e","o","E","O",);

if (arr($text,'/['.flat($ac).'][n][+]/') && !arr($text,'/['.flat($ac).'][n][+]$/') && !sub(array("Ahan"),blank(0),blank(0),0) && $sarvanamasthana1===1 && $sambuddhi===0 && $inhan===0 && $inhan1===0 && $aptRn===0 && $casambu!==1)

{

$text = two($ac,array("n+"),$acdir,array("n+"),0);

echo "<p class = sa >By sarvanAmasthAne cAsambuddhau (6.4.8) :</p>";

echo "<p class = sa >सर्वनामस्थाने चासम्बुद्धौ (६.४.८) :</p>";

echo "<p class = pa >alo'ntyAtpUrva upadhA (1.1.65) </p>";

echo "<p class = pa >अलोऽन्त्यात्पूर्व उपधा (१.१.६५) </p>";

display(3);

}

### /\* dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118) \*/

if ($dhatu===1 && in\_array($fo,array("dfnBU","karaBU","kAraBU","punarBU")) && in\_array($so,$acsup))

{

$text = three(array("dfnBU","punarBU"),array("+"),$ac,array("dfnBv","punarBv"),array("+"),$ac,0);

$text = three(array("karaBU","kAraBU"),array("+"),$ac,array("karaBv","kAraBv",),array("+"),$ac,1);

echo "<p class = sa >By dRnkarapunaHpUrvasya bhuvo yaN vaktavyaH (vA 4118) :</p>";

if (in\_array($fo,array("karaBU","kAraBU")))

{

$text=one(array("karaBv+e","karaBU+A+e","karaBv+as","karaBU+A+as","karaBv+i","karaBU+Am","kAraBv+e","kAraBU+A+e","kAraBv+as","kAraBU+A+as","kAraBv+i","kAraBU+Am"),array("karaBU+e","karaBv+A+e","karaBU+as","karaBv+A+as","karaBU+i","karaBv+Am","kAraBU+e","kAraBv+A+e","kAraBU+as","kAraBv+A+as","kAraBU+i","kAraBv+Am",),0);

echo "<p class = hn >There is pAThabheda here. Some hold that there is dIrgha kAra word here. SK has adopted both the readings, therefore we have kept them optional.</p>";

}

echo "<p class = sa >दृन्करपुनःपूर्वस्य भुवो यण्‌ वक्तव्यः (वा ४११८) :</p>";

if (in\_array($fo,array("karaBU","kAraBU")))

{

echo "<p class = hn >दीर्घपाठे करपूर्वस्य उवङेव । ह्रस्वपाठे करपूर्वस्य यणेव इति विवेकः ।</p>";

}

display(0);

}

### /\* varSAbhvazca (6.8.84) \*/

if ($dhatu===1 && $first==="varzABU" && in\_array($so,$acsup))

{

$text = two(array("varzABU"),array("+"),array("varzABv"),array("+"),0);

echo "<p class = sa >By varSAbhvazca (6.8.84) :</p>";

echo "<p class = sa >वर्षाभ्वश्च (६.४.८४) :</p>";

display(0);

}

if ($fo==="dfmBU")

{

$dfmBU=1; // 0 - the word is not dRmbhU. 1 - the word is dRmbhU.

} else { $dfmBU=0; }

### /\* na bhUsudhiyoH (6.4.85) \*/

if (in\_array($fo,array("suDI","praDI")) && $gender==="n")

{

$dhatu=1;

}

if (in\_array($fo,array("praDI")) && $gender==="n")

{

$eranekaca=1;

}

if ( $dhatu===1 && (arr($text,'/[B][U][+]/')||$fo==="suDI") && arr($text,'/[iuIU][+]['.flat($ac).']/') && in\_array($so,$sup) && $dfmBU===0)

{

echo "<p class = sa >By na bhUsudhiyoH (6.4.85) :</p>";

echo "<p class = sa >न भूसुधियोः (६.४.८५) :</p>";

display(3);

$nabhusu = 1; // 0 - the word neither ends with bhU, nor is it sudhI. 1 - The word ends with bhU or is sudhI.

} else { $nabhusu = 0; }

### /\* kvau luptaM na sthAnivat (vA 431) \*/

// Not displayed because it is difficult to teach sthnanivadbhav to machine now. Will come back to it if I can teach it some day.

### /\* aci znudhAtubhruvAM yvoriyaGuvaGau (6.4.77) \*/

// znu pending.

if (($dhatu===1||$fo==="BrU") && arr($text,'/[iuIU][+]['.flat($ac).']/') && $pada==="pratyaya" && ($eranekaca===0 || ($eranekaca===1 && anekAca($fo)===false ) || (arr($text,'/[B][U][+]/')||$fo==="suDI") ) && $dfmBU===0 )

{

$text = three(array("i","I","u","U"),array("+"),$ac,array("iy","iy","uv","uv"),array("+"),$ac,0);

echo "<p class = sa >By aci znudhAtubhruvAM yvoriyaGuvaGau (6.4.77) :</p>";

echo "<p class = hn >gatikAraketarapUrvapadasya yaN neSyate (vA 5034) mandates that eranekAco.../oH supi rule doesn't apply in cases where the pUrvapada is neither gati nor kAraka. iyaG or uvaG is applied in that case. :</p>";

echo "<p class = sa >अचि श्नुधातुभ्रुवां य्वोरियङुवङौ (६.४.७७) :</p>";

echo "<p class = hn >गतिकारकेतरपूर्वपदस्य यण्‌ नेष्यते (वा ५०३४) से गति / कारक से भिन्न पूर्वपद होने पर एरनेकाचो.../ओः सुपि सूत्र नहीं लागू होता । इयङ्‌ या उवङ्‌ की प्रवृत्ति होती है । :</p>";

display(3);

}

### /\* gatikAraketarapUrvapadasya yaN neSyate (vA 5034) \*/

// This is attached with eranekAco... So, trying to put a note and making the iyaG and yaN optional.

### /\* eranekAco'saMyogapUrvasya (6.4.82) \*/

if ($dhatu===1 && arr($text,'/[iI][+]['.pc('ac').']/') && in\_array($fo,array("unnI")) && $pada==="pratyaya" && anekAca($fo) && $eranekaca===1 && $nabhusu===0)

{

echo "<p class = pa >As the vizeSaNa 'dhAtunA saMyogasya' mandates that the saMyoga has to belong to dhAtu only for prohibiting 'eranekAco..', the prohibition doesn't apply here.</p>";

echo "<p class = pa >धातुना संयोगस्य विशेषणादिह स्यादेव यण्‌ (एरनेकाचो इत्यनेन सूत्रेण) </p>";

display(0);

$unni=1; // 0 - the word is not unnI. 1 - the word is unnI

} else { $unni=0; }

if ($dhatu===1 && (arr($text,'/['.flat($ac).']['.flat($hl).'][iI][+]['.flat($ac).']/')||$unni===1) && $pada==="pratyaya" && anekAca($fo) && $eranekaca!==0 && $nabhusu===0)

{

$text = three(array("i","I"),array("+"),$ac,array("y","y"),array("+"),$ac,0);

echo "<p class = sa >By eranekAco'saMyogapUrvasya (6.4.82) :</p>";

echo "<p class = hn >gatikAraketarapUrvapadasya yaN neSyate (vA 5034) mandates that this rule doesn't apply in cases where the pUrvapada is neither gati nor kAraka. iyaG or uvaG is applied in that case. :</p>";

echo "<p class = sa >एरनेकाचोऽसंयोगपूर्वस्य (६.४.८२) :</p>";

echo "<p class = hn >गतिकारकेतरपूर्वपदस्य यण्‌ नेष्यते (वा ५०३४) से गति / कारक से भिन्न पूर्वपद होने पर यह सूत्र नहीं लागू होता । इयङ्‌ या उवङ्‌ की प्रवृत्ति होती है । :</p>";

display(3);

}

### /\* oH supi (6.4.83) \*/

if ($dhatu===1 && in\_array($fo,array("ullU")) && $pada==="pratyaya" && anekAca($fo) && $$eranekaca===1 && $nabhusu===0 && in\_array($so,$sup) && $dfmBU===0)

{

echo "<p class = pa >As the vizeSaNa 'dhAtunA saMyogasya' mandates that the saMyoga has to belong to dhAtu only for prohibiting 'oH supi', the prohibition doesn't apply here.</p>";

echo "<p class = pa >धातुना संयोगस्य विशेषणादिह स्यादेव यण्‌ (ओः सुपि इत्यनेन सूत्रेण) </p>";

display(0);

$ullU=1; // 0 - word is not ullU. 1 - word is ullU.

} else { $ullU=0; }

if ($dhatu===1 && ( arr($text,'/[uU][+]['.flat($ac).']/') || $ullU===1 )&& $pada==="pratyaya" && anekAca($fo) && $eranekaca===1 && $nabhusu===0 && in\_array($so,$sup) && $dfmBU===0)

{

$text = three(array("u","U"),array("+"),$ac,array("v","v"),array("+"),$ac,0);

echo "<p class = sa >By oH supi (6.4.83) :</p>";

echo "<p class = hn >gatikAraketarapUrvapadasya yaN neSyate (vA 5034) mandates that this rule doesn't apply in cases where the pUrvapada is neither gati nor kAraka. iyaG or uvaG is applied in that case. :</p>";

echo "<p class = sa >ओः सुपि (६.४.८३) :</p>";

echo "<p class = hn >गतिकारकेतरपूर्वपदस्य यण्‌ नेष्यते (वा ५०३४) से गति / कारक से भिन्न पूर्वपद होने पर यह सूत्र नहीं लागू होता । इयङ्‌ या उवङ्‌ की प्रवृत्ति होती है । :</p>";

display(3);

}

### /\* ami pUrvaH (6.1.107) \*/

if ( sub(array("a","A","i","I","u","U","f","F","x"),array("+am"),blank(0),0))

{

$text = two(array("a","A","i","I","u","U","f","F","x"),array("am"),array("a","A","i","I","u","U","f","F","x"),array("m"),0);

echo "<p class = sa >By ami pUrvaH (6.1.107) :</p>";

echo "<p class = sa >अमि पूर्वः (६.१.१०७) :</p>";

display(0);

$amipUrva = 1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $amipUrva = 0; }

### /\* sAntamahataH saMyogasya (6.4.10) \*/

if ( ( (arr($text,'/['.flat($ac).'][nM][s][+]/') && $dhatu===1)|| $fo==="mahat" )&& $sarvanamasthana1===1 && $sambuddhi===0)

{

echo "<p class = pa >sAntamahataH saMyogasya (6.4.10) doesn't apply here. </p>";

echo "<p class = pa >सान्तमहतः संयोगस्य (६.४.१०) इत्यत्र सान्तसंयोगोऽपि प्रातिपदिकस्यैव गृह्यते, न तु धातोः, महच्छब्दसाहचर्यात्‌ ।</p>";

display(0);

}

if ( ( (arr($text,'/['.flat($ac).'][nM][s][+]/') && $dhatu===0)|| $fo==="mahat" )&& $sarvanamasthana1===1 && $sambuddhi===0)

{

$text = two($ac,array("ns+","Ms+"),$acdir,array("ns+","Ms+"),0);

$text = one(array("mahant"),array("mahAnt"),0);

echo "<p class = sa >By sAntamahataH saMyogasya (6.4.10) :</p>";

echo "<p class = sa >सान्तमहतः संयोगस्य (६.४.१०) :</p>";

echo "<p class = pa >alo'ntyAtpUrva upadhA (1.1.65) </p>";

echo "<p class = pa >अलोऽन्त्यात्पूर्व उपधा (१.१.६५) </p>";

display(0);

}

if (arr($text,'/['.flat($ac).'][nM][s][+]/') && ends(array($fo),array("mahat"),1) && $sarvanamasthana1===1 && $sambuddhi===0)

{

$text = two($ac,array("ns+"),$acdir,array("ns+"),0);

echo "<p class = sa >By sAntamahataH saMyogasya (6.4.10) :</p>";

echo "<p class = sa >सान्तमहतः संयोगस्य (६.४.१०) :</p>";

echo "<p class = pa >alo'ntyAtpUrva upadhA (1.1.65) </p>";

echo "<p class = pa >अलोऽन्त्यात्पूर्व उपधा (१.१.६५) </p>";

display(0);

}

if ( ends(array($fo),array("Danuz"),1) && $sarvanamasthana1===1 && $sambuddhi===0)

{

$text = two($ac,array("nz+"),$acdir,array("nz+"),0);

echo "<p class = sa >By sAntamahataH saMyogasya (6.4.10) :</p>";

echo "<p class = sa >सान्तमहतः संयोगस्य (६.४.१०) :</p>";

echo "<p class = pa >Because Satva is asiddha. </p>";

echo "<p class = pa >षत्व के असिद्ध होने के कारण । </p>";

display(0);

}

### /\* Rta ut (6.1.111) \*/

if ( (arr($text,'/[fx][+][a]/')) && in\_array($so,array("Nasi!","Nas")) && $pada==="pratyaya")

{

$text = two(array("f","x"),array("+a"),array("ur","ul"),array("+"),0);

echo "<p class = sa >By Rta ut (6.1.111) :</p>";

echo "<p class = sa >ऋत उत्‌ (६.१.१११) :</p>";

display(0);

$text = one(array("r+s"),array("r"),0);

echo "<p class = sa >By rAtsasya (8.2.24) :</p>";

echo "<p class = sa >रात्सस्य (८.२.२४) :</p>";

display(0);

}

### /\* auto'mzasoH (6.1.93) \*/

if (sub(array("o"),array("+"),array("a"),0) && in\_array($so,array("am","Sas")))

{

$text = two(array("o"),array("+a"),array(""),array("+A"),0);

echo "<p class = sa >By auto'mzasoH (6.1.93) :</p>";

echo "<p class = sa >औतोऽम्शसोः (६.१.९३) :</p>";

display(0);

}

### /\* ato bhisa ais (7.1.9) \*/

if (arr($text,'/[a][+]/') && $so === "Bis" && $nedamadas===0 && $fo!=="adas" )

{

$text = two(array("a"),array("Bis"),array("a"),array("Es"),0);

echo "<p class = sa >By ato bhisa ais (7.1.9) :</p>";

echo "<p class = sa >अतो भिस ऐस्‌ (७.१.९) :</p>";

display(5);

$atobhis = 1; // 0 - this sUtra has not applied. 1 - this sUtra has applied.

} else { $atobhis = 0; }

### /\* yasyeti ca (6.4.148) \*/

if (arr($text,'/[aI][+][I]/') && $bham===1 && $auGazyA===0)

{

$text = two(array("a","I"),array("I"),array("",""),array("I"),0);

echo "<p class = sa >By yasyeti ca (6.4.148) :</p>";

echo "<p class = sa >यस्येति च (६.१.१४८) :</p>";

display(3);

}

### /\* do-while loop for sapAdasaptAdhyAyI \*/

// Most of alterations in prakRti and pratyayas are over before this stage. Now in next step we keep all rules of sapAdasaptAdhyAyI, and create a do-while loop.

// This do - while loop is necessary, because there is no order in which rules of sapAdasaptAdhyAyI will apply. This do - while loop will continue till the $text which is input is the same as output.

// Therefore the cause for application may arise after application of any rule. So created a do-while loop which will check till the input and output are the same i.e. there is no difference after the application of all the sUtras.

// This will ensure that there is no rule of sapAdasaptAdhyAyI which would have applied, but didn't apply.

$start = 1; // remembering that this is the first round of application. After one loop, it will become 2. We have to stop application of some sUtras twice. There this is useful.

do // starting the loop.

{

$original = $text ; // remembering the member entering the loop. We will have to compare it with the output.

### /\* osi ca (7.3.104) \*/

if ($so === "os" && arr($text,'/[a][+]/') && $start ===1)

{

$text = two(array("a"),array($second),array("e"),array($second),0);

echo "<p class = sa >By osi ca (7.3.104) :</p>";

echo "<p class = sa >ओसि च (७.३.१०४) :</p>";

display(3);

}

### /\* na tisRcatasR (6.4.4) \*/

if ($so === "Am" && sub(array("tisf","catasf"),array("+"),blank(0),0))

{

echo "<p class = sa >By na tisRcatasR (6.4.4) :</p>";

echo "<p class = hn >This prevents application of 'nAmi'.</p>";

echo "<p class = sa >न तिसृचतसृ (६.४.४) :</p>";

echo "<p class = hn >यह सूत्र नामि की प्रवृत्ति को निषिद्ध करता है ।</p>";

display(3);

$natisf=1; // 0 - no prevention of nAmi. 1 - prevention of nAmi.

} else { $natisf = 0; }

### /\* explanation of prarINAm \*/

if (sub(array("rI"),array("+"),array("nAm"),0) && arr(array($fo),'/[r][E]$/'))

{

echo "<p class = hn >According to mAdhava, rAyo hali applies here to convert it to prarANAm. But it is wrong according to Siddhantakaumudi.</p>";

echo "<p class = hn >माधव के अनुसार यहाँ रायो हलि से आत्त्व होता है । परन्तु सिद्धान्तकौमुदी के अनुसार यह गलत है ।</p>";

display(0);

}

### /\* nAmi (6.4.3) and nR ca (6.4.6) \*/

if (arr($text,'/['.flat($ac).'][+][n][A][m]$/') && $start ===1 && !in\_array($fo,$sarvanama) && $natisf === 0)

{

if(arr($text,'/[n][f][+][n][A][m]$/'))

{

$text = two(array("nf"),array("nAm"),array("nF"),array("nAm"),1);

echo "<p class = sa >By nR ca (6.4.6) :</p>";

echo "<p class = sa >नृ च (६.४.६) :</p>";

display(3);

}

elseif (sub(array("a+"),array("nAm"),blank(0),0))

{

echo "<p class = pa >Even though, supi ca (7.3.102) is a parasUtra, it doesn't operate here. The causes are: (1) violation of sannipAtaparibhASA and (2) ArambhasAmarthya of 'nAmi'. :</p>";

echo "<p class = pa >यद्यपि सुपि च (७.३.१०२) परसूत्र है, फिर भी उसकी प्रवृत्ति यहाँ नहीं होती है । सन्निपातपरिभाषा के विरोध से और नामि च सूत्र के प्रारम्भ के सामर्थ्य से । </p>";

display(0);

echo "<p class = sa >By nAmi (6.4.3) :</p>";

echo "<p class = sa >नामि (६.४.३) :</p>";

$text = two($ac,array("nAm"),array("A","A","I","I","U","U","F","F","F","F","e","E","o","O"),array("nAm"),0);

}

else

{

echo "<p class = sa >By nAmi (6.4.3) :</p>";

echo "<p class = sa >नामि (६.४.३) :</p>";

$text = two($ac,array("nAm"),array("A","A","I","I","U","U","F","F","F","F","e","E","o","O"),array("nAm"),0);

}

display(3);

$nami = 1; // 0 - doesn't prevent application of supi ca. 1 - prevents application of supi ca.

}

else

{

$nami = 0;

}

### /\* bahuvacane jhalyet (7.3.103) \*/

if ((in\_array($so,array("Byas","sup","Bis")) || ($sut===1 && $sAmaAkam===0)) && arr($text,'/[a][+][Bs]/') && $start ===1)

{

$text = two(array("a"),array("Byas","su","sAm","Bis"),array("e"),array("Byas","su","sAm","Bis"),0);

echo "<p class = sa >By bahuvacane jhalyet (7.3.103) :</p>";

if (in\_array($so,array("Byas","Bis")) && arr($text,'/[e][+][B]/'))

{

echo "<p class = hn >This rule 'overrides supi ca'.</p>";

}

echo "<p class = sa >बहुवचने झल्येत्‌ (७.३.१०३) :</p>";

if (in\_array($so,array("Byas","Bis")) && arr($text,'/[e][+][B]/'))

{

echo "<p class = hn >'सुपि च' इति दीर्घे प्राप्ते परत्वादस्य सूत्रस्य प्रवृत्तिः ।</p>";

}

display(3);

$bahuvacane = 1; // 0 - doesn't prevent application of supi ca. 1 - prevents application of supi ca.

} else { $bahuvacane = 0; }

### /\* supi ca (7.3.102) \*/

if (in\_array($so,$sup) && arr($text,'/[a][+]['.pc('yY').']/') && $amipUrva===0 && $start === 1 && $bahuvacane === 0 && $nami === 0 && $Ne!==1)

{

if (arr($text,'/[a][+]['.pc('yY').']/') && in\_array($fo,array("idam","idakam")) )

{

echo "<p class = pa >Adyantavadekasmin (1.1.21) :</p>";

echo "<p class = pa >आद्यन्तवदेकस्मिन्‌ (१.१.२१) :</p>";

display(0);

}

$text = two(array("a"),array("+"),array("A"),array("+"),0);

echo "<p class = sa >By supi ca (7.3.102) :</p>";

echo "<p class = sa >सुपि च (७.३.१०२) :</p>";

display(3);

}

if ($Ne===1 && $start === 1)

{

$text = one(array("a+ya"),array("A+ya"),0);

echo "<p class = sa >By supi ca (7.3.102) :</p>";

echo "<p class = hn >'sannipAtalakSaNo vidhiranimittaM tadvighAtasya (pa 86) doesn't apply here. Its anityatva has been shown by kaSTAya kramaNe. </p>";

echo "<p class = sa >सुपि च (७.३.१०२) :</p>";

echo "<p class = hn >'सन्निपातलक्षणो विधिरनिमित्तं तद्विघातस्य (प ८६) यहाँ लागू नहीं होता है । कष्टाय क्रमणे से उसका अनित्यत्व ज्ञापित होता है ।</p>";

display(3);

}

### /\* pragRhya section \*/

### /\* plutapragRhyA aci nityam (6.1.125) \*/

// There is no definition of pluta / pragRhya here. So we will code that as and when case arises.

### /\* iko'savarNe zAkalyasya hrasvazca (6.1.127) \*/ // Right now coded for only dIrgha. Clarify wheter the hrasva preceding also included?

$ik = array("i","I","u","U","f","F","x","X");

$nonik = array("a","A","e","E","o","O");

if (sub($ik,array("+"),$nonik,0) && $pada==="pada")

{

// for pragRhya, it is difficult to tell the machine that it is not to be combined. So we have added one additional space. e.g. "a" -> "a ".

$text = two(array("i+","I+"),array("a","A","u","U","f","F","x","X","e","o","E","O"),array("i +","i +"),array("a","A","u","U","f","F","x","X","e","o","E","O"),1);

$text = two(array("u+","U+"),array("a","A","i","I","f","F","x","X","e","o","E","O"),array("u +","u +"),array("a","A","i","I","f","F","x","X","e","o","E","O"),1);

$text = two(array("f+","F+"),array("a","A","u","U","i","I","e","o","E","O"),array("f +","f +"),array("a","A","u","U","i","I","e","o","E","O"),1);

$text = two(array("x+","X+"),array("a","A","u","U","i","I","e","o","E","O"),array("x +","x +"),array("a","A","u","U","i","I","e","o","E","O"),1);

echo "<p class = sa >By iko'savarNe zAkalyasya hrasvazca (6.1.127) :</p>

<p class = hn >Note that this will not apply in samAsa. Also when followed by a 'sit' pratyaya, it will not apply. e.g. pArzva</p>";

echo "<p class = sa >इकोऽसवर्णे शाकल्यस्य ह्रस्वश्च (६.१.१२७) :</p>

<p class = hn >समास व सित्‌ प्रत्यय परे होने पर यह लागू नहीं होता । जैसे कि पार्श्व ।</p>";

display(0);

}

### /\* upasargAdRti dhAtau (6.1.11) and vA supyApizaleH (6.1.12) \*/

$akarantaupasarga = array("pra","apa","ava","upa",);

$changedupasarga = array("prAr","apAr","avAr","upAr");

$changedupasarga1 = array("prar","apar","avar","upar");

$changedupasarga2 = array("prAl","apAl","avAl","upAl");

$changedupasarga3 = array("pral","apal","aval","upal");

// for $verbs\_ru and $verbs\_changed, please see function.php.

if ((sub($akarantaupasarga,$verbs\_ru,blank(0),0) && !sub(array("prafRa"),blank(0),blank(0),0))||sub($akarantaupasarga,array("xkArIy"),blank(0),0))

{

if (arr($text,'/[I][y]/'))

{

$text = two($akarantaupasarga,$verbs\_ru,$changedupasarga,$verbs\_changed,1);

}

else

{

$text = two($akarantaupasarga,$verbs\_ru,$changedupasarga,$verbs\_changed,0);

}

$text = two($akarantaupasarga,array("xkArIy"),$changedupasarga2,array("kArIy"),1);

$text = two($akarantaupasarga,array("xkArIy"),$changedupasarga3,array("kArIy"),0);

echo "<p class = sa >By upasargAdRti dhAtau (6.1.11) and vA supyApizaleH (6.1.12) :</p>";

echo "<p class = hn >In case akArAnta upasarga is followed by RkArAdi nAmadhAtu, there is optional vRddhi ekAdeza. If there is dIrgha RUkAra at the start of dhAtu or nAmadhAtu, upasargAdRti dhAtau and vA supyApizaleH don't apply. iko yaNaci and uraNraparaH apply.</p>";

echo "<p class = sa >उपसर्गादृति धातौ (६.१.११) तथा वा सुप्यापिशलेः (६.१.१२) :</p>";

echo "<p class = hn >अकारान्त उपसर्ग से ऋकारादि नामधातु परे होने पर विकल्प से वृद्धि एकादेश होता है । यदि धातु या नामधातु ॠकार से आरंभ होता है, तब उपसर्गादृति धातौ तथा वा सुप्यापिशलेः लागू नहीं होते हैं । अतः इको यणचि व उरण्रपरः ही लागू होते हैं ।</p>";

display(0);

$upas = 1; // 0 - doesn't prevent application of RtyakaH. 1 - prevents application of RtyakaH.

} else { $upas = 0; }

### /\* RtyakaH (6.1.128) \*/

$ak = array("a","A","i","I","u","U","f","F","x","X");

$akrt = array("a ","A ","i ","I ","u ","U ","f ","F ","x ","X ");

if (arr($text,'/['.flat($ak).'][+][fx]/') && $start===1 && $pada ==="pada" && $upas ===0 )

{

if (sub($ak,array("f","x"),blank(0),0))

{

$text = two ($ak,array("f","x"),$akrt,array("f","x"),1);

echo "<p class = sa >By RtyakaH (6.1.128) :</p>

<p class = hn >Note: This applies only to padAnta. </p>";

echo "<p class = sa >ऋत्यकः (६.१.१२८) :</p>

<p class = hn >Note: This applies only to padAnta. </p>";

display(0);

}

}

if ($upas === 1)

{

echo "<p class = hn >RtyakaH is barred by upasargAdRti dhAtau. </p>

<p class = hn >ऋत्यकः उपसर्गादृति धातौ से बाधित हुआ है ।</p><hr>";

}

### /\* vAkyasya TeH pluta udAttaH (8.2.82) \*/

// This is adhikArasutra. Nothing to code here.

### /\* pratyabhivAde'zUdre (8.2.83) \*/

### /\* dUrAddhUte ca (8.2.84) \*/

### /\* haihe prayoge haihayoH (8.2.85) \*/

### /\* guronanRto'nantyasyApyekaikasya prAcAm (8.2.86) \*/

### /\* aplutavadupasthite (6.1.129) \*/

### /\* I3 cAkravarmaNasya (6.1.130) \*/

// These two are not possible to code, because it will depend on the speaker's choice. Maybe I will add some ajax for user feedback. Pending.

### /\* IdUdeddvivacanaM pragRhyam (1.1.11) \*/

// not possible to code till we get the word forms of all words and check whether it is dvivacana or not. Pending

### /\* adaso mAt (1.1.12) \*/

if (sub(array("amI"),blank(0),blank(0),0) && $fo === "amI" && $start===1)

{

$text = two (array("amI"),$ac,array("amI "),$ac,1);

echo "<p class = sa >By adaso mAt (1.1.12) :</p>";

echo "<p class = sa >अदसो मात्‌ (१.१.१२) :</p>";

display(0);

}

if (sub(array("amU"),blank(0),blank(0),0)&& $fo === "amU" &&$start===1)

{

$text = two (array("amU"),$ac,array("amU "),$ac,1);

echo "<p class = sa >By adaso mAt (1.1.12) :</p>";

echo "<p class = sa >अदसो मात्‌ (१.१.१२) :</p>";

display(0);

}

### /\* ze (1.1.13) \*/

// Not possible to know whether one form has ze or not. ajax for feedback pending.

### /\* nipAta ekAjanAG (1.1.14) \*/

$nipata = array("a","A","i","I","u","U","e","E","o","O"); // list of ekAc nipAta.

$nipata1 = array("a ","A ","i ","I ","u ","U ","e ","E ","o ","O "); // adding a space for pragRhya.

if (in\_array($first,$nipata) && sub(array($first),$ac,blank(0),0) && $start===1)

{

$text = two ($nipata,$ac,$nipata1,$ac,0);

echo "<p class = sa >By nipAta ekAjanAG (1.1.14) :</p>";

echo "<p class = sa >निपात एकाजनाङ्‌ (१.१.१४) :</p>";

display(0);

}

### /\* ot (1.1.15) \*/

$ot = array("o","aho","ho","utAho","aTo");

$ot1 = array("o ","aho ","ho ","utAho ","aTo ");

if (in\_array($first,$ot) && sub(array($first),$ac,blank(0),0) && $start===1)

{

$text = two ($ot,$ac,$ot1,$ac,0);

echo "<p class = sa >By ot (1.1.15) :</p>";

echo "<p class = sa >ओत्‌ (१.१.१५) :</p>";

display(0);

}

### /\* sambuddhau zAkalyasyetAvanArSe (1.1.16) \*/

if (arr($text,'/[o][+]/') && $second === "iti" && $start===1)

{

$text = two(array($first),$ac,array($first." "),$ac,1);

echo "<p class = sa >By sambuddhau zAkalyasyetAvanArSe (1.1.16) :</p>

<p class = hn >Note: This rule will apply only in case the 'o'kAra at the end of the first word is for sambuddhi and the 'iti' is anArSa (of non-vedic origin).</p>";

"<p class = sa >सम्बुद्धौ शाकल्यस्येतावनार्षे (१.१.१६) :</p>

<p class = hn >यह नियम तभी लागू होगा जब प्रथम शब्द के अन्त का ओकार सम्बुद्धि के लिए हो और अनार्ष 'इति' शब्द उसके परे हो ।</p>";

display(0);

}

### /\* UYaH (1.1.17) \*/

if ($first === "u" && $second === "iti" && $start===1)

{

$text = two(array("u"),array("iti"),array("u "),array("iti"),1);

echo "<p class = sa >By uYaH (1.1.17) :</p>";

echo "<p class = sa >उञः (१.१.१७) :</p>";

display(0);

}

### /\* U! (1.1.18) \*/

// Here ! has been used for anunAsika.

if ($first === "u" && $second === "iti" && $start===1)

{

$text = two(array("u"),array("iti"),array("U! "),array("iti"),1);

echo "<p class = sa >By U! (1.1.17) :</p>";

echo "<p class = sa >ऊँ (१.१.१७) :</p>";

display(0);

}

### /\* maya uYo vo vA (8.3.33) \*/

if (sub(array("Sam","kim","tvam","tad"),array("u"),$ac,0))

{

$text = three(array("Sam","kim","tvam","tad"),array("u"),$ac,array("Sam","kim","tvam","tad"),array("v"),$ac,1);

echo "<p class = sa >By maya uYo vo vA (8.3.33) :</p>";

echo "<p class = sa >मय उञो वो वा (८.३.३३) :</p>";

display(0);

}

### /\* IdUtau ca saptamyarthe (1.1.19) \*/

/\*$idut = array("I","U"); $idut1 = array("I ","U ");

if (preg\_match('/[IU]$/',$first) && sub(array("I","U"),$ac,blank(0),0) && $pada ==="pada")

{

$text = two($idut,$ac,$idut1,$ac,1);

echo "<p class = sa >By IdUtau ca saptamyarthe (1.1.19) :</p>";

echo "<p class = hn >N.B.: This will apply only in case the I/U at the end of the first word have been used in sense of saptamI vibhakti. Otherwise this pragRhyatva will not be there.</p>";

echo "<p class = sa >ईदूतौ च सप्तम्यर्थे (१.१.१९) :</p>";

echo "<p class = hn >यदि प्रथम पद के अन्त में ई / ऊ सप्तमी के अर्थ में प्रयुक्त हुए हों तभी यह नियम लागू होगा ।</p>";

display(0);

}\*/ // vedic in nature. gives too many wrong results. commented right now.

### /\* zakandhvAdiSu pararUpaM vAcyam (vA 3632) \*/

$shakandhu1 = array("Saka","karka","kula","manas","hala","lANgala","patan","mfta"); // first word of zakandhvAdi

$shakandhu2 = array("anDu","anDu","awA","IzA","IzA","IzA","aYjali","aRqa"); // second word of zakandhvAdi

$shakandhu = array("Sak","kark","kul","man","hal","lANgal","pat","mArt"); // replacement of first word.

if (sub($shakandhu1,$shakandhu2,blank(0),0))

{

$text = two($shakandhu1,$shakandhu2,$shakandhu,$shakandhu2,0);

echo "<p class = sa >By zakandhvAdiSu pararUpaM vAcyam (vA 3632) :</p>";

echo "<p class = sa >शकन्ध्वादिषु पररूपं वाच्यम्‌ (वा ३६३२) :</p>";

display(0);

}

$shakandhu1 = array("sIman","sAra"); // ajax possible. pending.

$shakandhu2 = array("anta","aNga");

$shakandhu = array("sIm","sAr");

if (sub($shakandhu1,$shakandhu2,blank(0),0))

{

$text = two($shakandhu1,$shakandhu2,$shakandhu,$shakandhu2,1);

$text = one(array("sIman+ant","sAra+aNg"),array("sIm+Ant","sAr+ANg"),0);

echo "<p class = sa >By zakandhvAdiSu pararUpaM vAcyam (vA 3632) :</p>";

echo "<p class = hn >Note: the sImanta - kezaveSa and sAraGga - pazu/pakSI - Then only this will apply.</p>";

echo "<p class = sa >शकन्ध्वादिषु पररूपं वाच्यम्‌ (वा ३६३२) :</p>";

echo "<p class = hn >यदि सीमन्त केशवेश के अर्थ में और सारङ्ग पशु-पक्षी के अर्थ में प्रयुक्त हुए हो, तभी यह नियम लागू होता है ।</p>";

display(0);

}

### /\* omAGozca (6.1.95) \*/

$aag = array("om","OM","Aj","Acy","AYc","Anakt","Att","As","er","Eray","okz","ArcC","Arpit","Ohyat","oQ","arSyAt"); // list for OM and AG.

if (sub(array("a","A"),$aag,blank(0),0))

{

$text = two(array("a","A"),array("om","OM"),blank(2),array("om","om"),0);

$text = two(array("a","A"),$aag,blank(2),$aag,0);

echo "<p class = sa >By omAGozca (6.1.95) :</p>

<p class = hn >The om or AG following the a,A gets converted to pararUpa. </p>";

echo "<p class = sa >ओमाङोश्च (६.१.९५) :</p>

<p class = hn >अ/आ के परे यदि ओम्‌ या आङ्‌ हो तो पररूप होता है । </p>";

display(0);

}

### /\* ATazca (6.1.90) \*/

// right now only ANnadyAH case is taken. Add the other cases as and when they arise.

if ((arr($text,'/[+][A][+]['.flat($ac).']/') && $ANnadyAH===1) )

{

$text = two(array("+A+"),$ac,array("+"),vriddhi($ac),0);

echo "<p class = sa >By ATazca (6.1.90) :</p>";

echo "<p class = sa >आटश्च :</p>";

display(0);

}

### /\* striyAH (6.4.79) \*/

if (ends(array($fo),array("strI","stri"),1) && arr($text,'/[s][t][r][Ii][+]['.pc('ac').']/') && in\_array($so,$acsup) && !in\_array($so,array("am","Sas","Am")))

{

$text = one(array("strI+","stri+"),array("striy+","striy+"),0);

echo "<p class = sa >By striyAH (6.4.79) :</p>";

echo "<p class = sa >स्त्रियाः (६.४.७९) :</p>";

display(3);

}

### /\* nAdici (6.1.104) \*/

$ic = array("i","I","u","U","f","F","x","X","e","o","E","O");

if (arr($text,'/[a][+]['.flat($ic).']/') && !arr($text,'/[a][+][aA]/') && (in\_array($so,$prathama)))

{

echo "<p class = pa >By nAdici (6.1.104) :</p>

<p class = hn >N.B. : This is exception to prathamayoH pUrvasavarNaH. </p>";

echo "<p class = pa >नादिचि (६.१.१०४) :</p>

<p class = hn >यह नियम प्रथमयोः पूर्वसवर्णः का अपवाद है ।</p>";

display (0);

$nadici = 1; // 0 - this rule has not applied. 1 - this rule has applied.

} else { $nadici = 0; }

### /\* dIrghAjjasi ca (6.1.105) \*/

if ((arr($text,'/[AIUFeEoO][+]['.flat($ic).']/')||((sub($dirgha,array("+"),array("as"),0)) && $so==="jas")) && (in\_array($so,$prathama)))

{

echo "<p class = pa >By dIrghAjjasi ca (6.1.105) :</p>

<p class = hn >N.B. : This is exception to prathamayoH pUrvasavarNaH. </p>";

echo "<p class = pa >दीर्घाज्जसि च (६.१.१०५) :</p>

<p class = hn >यह नियम प्रथमयोः पूर्वसवर्णः का अपवाद है ।</p>";

display (0);

$nadici1 = 1; // 0 - this rule has not applied. 1 - this rule has applied.

} else { $nadici1 = 0; }

### /\* prathamayoH pUrvasavarNaH (6.1.102) \*/

// Not coded well. Please revisit.

$ak = array("a","A","i","I","u","U","f","F","x","X");

$akreplace = array("A","A","I","I","U","U","F","F","F","X");

if (sub($ak,array("+"),array("a","O"),0) && in\_array($so,$prathama) && $nadici === 0 && $nadici1 === 0)

{

if (sub(array("a"),array("a","O"),blank(0),0)&& in\_array($so,$prathama) && $nadici === 0 && $nadici1 === 0)

{

echo "<p class = hn >ato guNe (6.1.96) is exception to only akaH savarNe dIrghaH (6.1.101) and not to prathamayoH pUrvasavarNaH (6.1.102), because of the paribhASA 'purastAdapavAdA anantarAnvidhInbAdhante nottarAn (pa 60). Therefore pURvasavarNadIrgha can occur. :</p>";

echo "<p class = hn >'पुरस्तादपवादा अनन्तरान्विधीन्बाधन्ते नोत्तरान्‌ (प ६०) परिभाषा के कारण अतो गुणे (६.१.९६) सिर्फ अकः सवर्णे दीर्घः (६.१.१०१) का अपवाद है, प्रथमयोः पूर्वसवर्णः (६.१.१०२) का नहीं । अतः पूर्वसवर्णदीर्घ हो सकता है ।</p>";

display (0);

}

$text = two($ak,array("a","O"),$akreplace,array("",""),0);

echo "<p class = sa >By prathamayoH pUrvasavarNaH (6.1.102) :</p>

<p class = hn >N.B. : This applies to only in prathamA and dvitIyA vibhakti, and not in other cases. </p>";

echo "<p class = sa >प्रथमयोः पूर्वसवर्णः (६.१.१०२) :</p>

<p class = hn >यह प्रथमा और द्वितीया विभक्तियों में लागू होता है ।</p>";

display (0);

$prathamayoh = 1; // 0 - this rule has not applied. 1 - this rule has applied. Useful in tasmAcChaso naH puMsi.

} else { $prathamayoh = 0; }

### /\* tasmAcChaso naH puMsi (6.1.103) \*/

if ($prathamayoh ===1 && $so === "Sas" && $gender==="m")

{

$text = one(array("+s"),array("+n"),0);

echo "<p class = sa >By tasmAcChaso naH puMsi (6.1.103) :</p>";

echo "<p class = sa >तस्माच्छसो नः पुंसि (६.१.१०३) :</p>";

display(0);

}

### /\* Rti savarNe R vA (vA 3640) and lRti savarNe lR vA (vA 3641) \*/

$ruti1 = array("f","F","x","X");

$ruti2 = array("f");

$lruti2 = array("x");

if (sub($ruti1,array("f","x"),blank(0),0))

{

$text = two($ruti1,array("f"),blank(count($ruti1)),$ruti2,1);

$text = two($ruti1,array("x"),blank(count($ruti1)),$lruti2,1);

echo "<p class = sa >By Rti savarNe R vA (vA 3640) and lRti savarNe lR vA (vA 3641) :</p>";

echo "<p class = sa >ऋति सवर्णे ऋ वा (वा ३६४०) तथा लृति सवर्णे लृ वा (वा ३६४१) :</p>";

display(0);

}

### /\* akaH savarNe dIrghaH (6.1.101) \*/

$ak1 = array("a","a","A","A","i","i","I","I","u","u","U","U","f","f","F","F","f","f","F","F","x","x","X","X","x","x","X","X");

$ak2 = array("a","A","a","A","i","I","i","I","u","U","u","U","f","F","f","F","x","X","x","X","f","F","f","F","x","X","x","X");

if (sub($ak1,$ak2,blank(28),1))

{

$text = two(array("a","A"),array("a","A"),array("A","A"),blank(2),0);

$text = two(array("i","I"),array("i","I"),array("I","I"),blank(2),0);

$text = two(array("u","U"),array("u","U"),array("U","U"),blank(2),0);

$text = two(array("f","F","x","X"),array("f","F","x","X"),array("F","F","F","F"),blank(4),0);

$text = two(array("x","X"),array("x","X"),array("F","F"),blank(2),0);

echo "<p class = sa >By akaH savarNe dIrghaH (6.1.101) :</p>";

echo "<p class = sa >अकः सवर्णे दीर्घः (६.१.१०१) :</p>";

display(0);

}

### /\* ato guNe (6.1.17) \*/

if (sub(array("a"),array("a","e","o"),blank(0),0) && !sub(array("pra","upa"),array("a","e","o"),blank(0),0) && $pada === "pratyaya" )

{

$text = two(array("a"),array("a","e","o"),blank(1),array("a","e","o"),0);

echo "<p class = sa >By ato guNe (6.1.17) :</p>";

echo "<p class = sa >अतो गुणे (६.१.१७) :</p>";

display(0);

}

### /\* hrasvasya guNaH (7.3.108) \*/

if (arr($text,'/[iufx][+][s]/') && $so==="su!" && $sambuddhi===1 && $amba===0)

{

$text = two(array("i","u","f","x"),array("+"),array("e","o","ar","al"),array("+"),0);

echo "<p class = sa >By hrasvasya guNaH (7.3.108) :</p>";

echo "<p class = sa >ह्रस्वस्य गुणः (७.३.१०८) :</p>";

display(3);

}

if (arr($text,'/[iufx][+]$/') && $so==="su!" && $sambuddhi===1 && $svamo===1)

{

$text = two(array("i","u","f","x"),array("+"),array("e","o","ar","al"),array("+"),1);

echo "<p class = sa >By hrasvasya guNaH (7.3.108) :</p>";

echo "<p class = sa >ह्रस्वस्य गुणः (७.३.१०८) :</p>";

display(3);

}

### /\* halGyAbbhyo dIrghAtsutisyapRktaM hal (6.1.68) and apRkta ekAlpratyayaH (1.2.41) \*/

// GyAp pending. only hal handled now.

if ((arr($text,'/['.pc('hl').'][+][sts]$/') || $GI===1 || $Ap===1 )&& in\_array($so,array("su!","ti","si",)))

{

echo "<p class = pa >By apRkta ekAlpratyayaH (1.2.41) :</p>";

echo "<p class = pa >अपृक्त एकाल्प्रत्ययः (१.२.४१) :</p>";

display(0);

}

if ((arr($text,'/['.pc('hl').'][+][sts]$/') )&& in\_array($so,array("su!","ti","si")))

{

$text = two($hl,array("+s","+t"),$hl,array("+","+"),0);

echo "<p class = sa >By halGyAbbhyo dIrghAtsutisyapRktaM hal (6.1.68) :</p>";

echo "<p class = sa >हल्ङ्‍याब्भ्यो दीर्घात्सुतिस्यपृक्तं हल्‌ (६.१.६८) :</p>";

display(0);

$pada="pada"; // there is no pratyaya left now.

}

if ($GI===1 && arr($text,'/[I][+][st]$/') && in\_array($so,array("su!","ti","si")))

{

$text = two(array("I"),array("+s","+t"),array("I"),array("+","+"),0);

echo "<p class = sa >By halGyAbbhyo dIrghAtsutisyapRktaM hal (6.1.68) :</p>";

echo "<p class = sa >हल्ङ्‍याब्भ्यो दीर्घात्सुतिस्यपृक्तं हल्‌ (६.१.६८) :</p>";

display(0);

$pada="pada"; // there is no pratyaya left now.

}

if ( $Ap===1 && arr($text,'/[A][+][st]$/') && in\_array($so,array("su!","ti","si")))

{

$text = two(array("A"),array("+s","+t"),array("A"),array("+","+"),0);

echo "<p class = sa >By halGyAbbhyo dIrghAtsutisyapRktaM hal (6.1.68) :</p>";

echo "<p class = sa >हल्ङ्‍याब्भ्यो दीर्घात्सुतिस्यपृक्तं हल्‌ (६.१.६८) :</p>";

display(0);

$pada="pada"; // there is no pratyaya left now.

}

### /\* patch for varShABU sambodhana \*/

if ($dhatu===1 && $first==="varzABU" && sub(array("varzABU"),array("+"),array("s"),1) && $pada==="pratyaya" )

{

$text = two(array("varzABU"),array("+"),array("varzABu"),array("+"),1);

echo "<p class = sa >kaiyaTa believes it to be 'he varSAbhUH'. The alternative belief is 'he varSAbhu'.</p>";

echo "<p class = sa >भेकजातौ नित्यस्त्रीत्वाभावाद्‌ हे वर्षाभूः कैयटमते । मतान्तरे तु हे वर्षाभु ।</p>";

display(0);

}

### /\* eGhrasvAtsambuddheH (6.1.69) and ekavacanaM sambuddhiH (2.3.49) \*/ // removed the last letter, not as in sutra. Look out for issues if any crops up.

if ($sambuddhi === 1 && $so === "su!" && (sub($hrasva,array("+"),array("s","m"),0)||sub(array("e","o"),array("+"),array("s","m"),0)) && arr($text,'/[+][sm]$/'))

{

$text = one(array("a+s","i+s","u+s","f+s","x+s","e+s","o+s","a+m"),array("a+","i+","u+","f+","x+","e+","o+","a+"),0); // this was tried. But it gives wrong results in kroSTrI sambuddhi.

/\* foreach ($text as $value)

{

if(substr($value,-1)!=="+")

{

$value1[] = substr($value,0,strlen($value)-1);

}

else

{

$value1[] = $value;

}

}

$text = $value1;

$value1 = array();\*/

echo "<p class = sa >By eGhrasvAtsambuddheH (6.1.69) :</p>";

echo "<p class = sa >एङ्ह्रस्वात्सम्बुद्धेः (६.१.६९) :</p>";

display(0);

}

### /\* dazca (7.2.109) \*/

if (sub(array("ida","idak","idA","idAn",),blank(0),blank(0),0) && in\_array($fo,array("idam","idakam")) && in\_array($so,$sup))

{

$text = one(array("ida+","idak","idA+","idAn+","idakA+"),array("ima+","imak","imA+","imAn+","imakA+"),0);

echo "<p class = sa >By dazca (7.2.109) :</p>";

echo "<p class = sa >दश्च (७.२.१०९) :</p>";

display(3);

}

### /\* iko yaNaci (6.1.77) \*/

if(sub(array('i','I','u','U'),prat('ac'),blank(0),0))

{

$text = two(array('i','I','u','U'),prat('ac'),array('y','y','v','v'),prat('ac'),0);

echo "<p class = sa >By iko yaNaci (6.1.77) :</p>";

echo "<p class = sa >इको यणचि (६.१.७७) :</p>";

display(4);

}

if(sub(array("f","F","x","X"),prat('ac'),blank(0),0))

{

$text = two(array("f","F","x","X"),prat('ac'),array("r","r","l","l"),prat('ac'),0);

echo "<p class = sa >By iko yaNaci (6.1.77) :</p>";

echo "<p class = sa >इको यणचि (५.१.७७) :</p>";

$sthanivadbhav = 1; // 0 - no sthAnivadbhAva. 1 - sthAnivadbhAva.

display(4);

}

else

{

$sthanivadbhav = 0;

}

### /\* sarvatra vibhASA goH (6.1.122) \*/

$go = array("go"); $aonly = array("a");

if(sub($go,$aonly,blank(0),0) && ( $pada==="pada" || $bham===1 || sub(array("goanc"),blank(0),blank(0),0)))

{

$text = two($go,$aonly,array("go "),$aonly,1);

echo "<p class = sa >By sarvatra vibhASA goH (6.1.122)</p>

<p class = hn >it is optionally kept as prakRtibhAva :</p>";

echo "<p class = sa >सर्वत्र विभाषा गोः (६.१.१२२)</p>

<p class = hn >पाक्षिक रूप से प्रकृतिभाव भी होता है ।</p>";

display(0);

}

### /\* avaG sphoTAyanasya (6.1.123) \*/

if (sub($go,prat('ac'),blank(0),0) && ( $pada==="pada" || $bham===0 || sub(array("goanc"),blank(0),blank(0),0)) )

{

$text = two($go,prat('ac'),array('gava'),prat('ac'),1);

echo "<p class = sa >By avaG sphoTAyanasya (6.1.123) </p>

<p class = hn >it is optionally converted to avaG :</p>";

echo "<p class = sa >अवङ्‌ स्फोटायनस्य (६.१.१२३) </p>

<p class = hn >पाक्षिक रूप से अवङ्‌ भी होता है ।</p>";

display(0);

}

### /\* indre ca (6.1.124) \*/

if (sub($go,array("indra"),blank(0),0) && ( $pada==="pada" || $bham===1 || sub(array("goanc"),blank(0),blank(0),0)))

{

$text = two($go,array("indra"),array('gava'),array("indra"),0);

echo "<p class = sa >by indre ca (6.1.124) :</p>";

echo "<p class = sa >इन्द्रे च (६.१.१२४) :</p>";

display(0);

}

### /\* eGaH padAntAdati (6.1.109) \*/

if (sub(array("e","o"),array("a"),blank(0),0) && ( $pada==="pada" || sub(array("goanc"),blank(0),blank(0),0)))

{

$text = two(prat('eN'),array("a"),prat('eN'),array("'"),0);

echo "<p class = sa >By eGaH padAntAdati (6.1.109) : </p>";

echo "<p class = sa >एङः पदान्तादति (६.१.१०९) : </p>";

display(0);

}

### /\* eco'yavAyAvaH (7.1.78) \*/

$ayavayavah = array("ay","av","Ay","Av");

if (sub(prat('ec'),prat('ac'),blank(0),0))

{

$text = two(prat('ec'),prat('ac'),$ayavayavah,prat('ac'),0);

echo "<p class = sa >By echo'yavAyAvaH (7.1.78) :</p>";

echo "<p class = sa >एचोऽयवायावः (७.१.७८) :</p>";

display(4);

}

### /\* vAnto yi pratyaye (6.1.71), goryutau CandasyupasaMkhyAnam (vA 3543), adhvaparimANe ca (vA 3544) dhAtostannimittasyaiva (6.1.80) \*/

$o = array("o","O");

$oo = 'oO';

$y = array("y");

$ab = array("av","Av");

$dhato = array("urRo","ro","no","sno","kzo","kzRo","Do","Dro","do","dro","sro","so","ko","Go","qo","cyo","jyo","pro","plo","ro","ho","Sro","hno","dyo","sko","po","lo","kno","mo","Bo","urRO","rO","snO","kzO","kzRO","DO","DrO","dO","drO","srO","sO","kO","GO","qO","cyO","jyO","prO","plO","rO","hO","SrO","hnO","dyO","skO","pO","lO","knO","mO","BO","Co","zo","So");

$dhato1 = array("urRa","ra","na","sna","kza","kzRa","Da","Dra","da","dra","sra","sa","ka","Ga","qa","cya","jya","pra","pla","ra","ha","Sra","hna","dya","ska","pa","la","kna","ma","Ba","urRA","rA","snA","kzA","kzRA","DA","DrA","dA","drA","srA","sA","kA","GA","qA","cyA","jyA","prA","plA","rA","hA","SrA","hnA","dyA","skA","pA","lA","knA","mA","BA","Ca","za","Sa");

if (sub($dhato,$y,blank(0),0) && $pada ==="pratyaya" && $start ===1)

{

$text = two($dhato,$y,$dhato1,array("vy"),0);

echo "<p class = sa >By dhAtostannimittasyaiva (6.1.77) : </p>";

echo "<p class = sa >धातोस्तन्निमित्तस्यैव (६.१.७७) : </p>";

display(0);

}

if (sub($o,$y,blank(0),0) && $pada ==="pratyaya" && !sub($dhato,$y,blank(0),0))

{

if (sub(array("gav"),$y,blank(0),0))

{

$text = two($o,$y,$ab,$y,0);

echo "<p class = sa >By vAnto yi pratyaye (6.1.71), goryutau CandasyupasaMkhyAnam (vA 3543), adhvaparimANe ca (vA 3544) : </p>

<p class = hn > If the 'y' following 'o/O' belongs to a pratyaya or the word 'go' is followed by 'yuti' in Chandas/ as a measure of distance (vA 3543, 3544).</p>";

echo "<p class = sa >वान्तो यि प्रत्यये (६.१.७१), गोर्यूतौ छन्दस्युपसंख्यानम्‌ (वा ३५४३), अध्वपरिमाणे च (वा ३५४४) : </p>

<p class = hn > यकारादि प्रत्यय के परे रहते या वैदिक भाषा / अध्वपरिमाण के अर्थ में यूति शब्द परे हो तब यह नियम लागू होता है ।</p>";

}

elseif (sub(array("o","O"),array("yat"),blank(0),0))

{

echo "<p class = sa >By dhAtostannimittasyaiva (6.1.77) : </p>";

echo "<p class = hn >Here the 'o' is because of 'y'. Therefore This doesn't apply. </p>";

echo "<p class = sa >धातोस्तन्निमित्तस्यैव (६.१.७७) :</p>";

echo "<p class = hn >ओयते में ओकार यकारनिमित्तक नहीं होने के कारण इस सूत्र से अवादेश नहीं हुआ ।</p>";

}

else

{

$text = two($o,$y,$ab,$y,0);

echo "<p class = sa >By vAnto yi pratyaye (6.1.71) : </p>";

echo "<p class = sa >वान्तो यि प्रत्यये (६.१.७१) :</p>";

}

display(0);

}

### /\* kSayyajayyau zakyArthe (6.1.81)\*/

if (sub(array("kze"),array("ya"),blank(0),0) && $pada ==="pratyaya" && $start === 1)

{

$text = two(array("kze"),array("ya"),array("kzay"),array("ya"),1);

echo "<p class = sa >By kSayyajayyau zakyArthe (6.1.81) :</p>

<p class = hn >If the word is to be used in the meaning of 'being capable of', then only it will be क्षय्य.</p>";

echo "<p class = sa >क्षय्यजय्यौ शक्यार्थे (६.१.८१) :</p>

<p class = hn >यदि क्षय कर सकने के अर्थ में प्रयोग होता है तभी क्षय्य होगा ।</p>";

display(0);

}

if (sub(array("je"),array("ya"),blank(0),0) && $pada ==="pratyaya" && $start === 1)

{

$text = two(array("je"),array("ya"),array("jay"),array("ya"),1);

echo "<p class = sa >By kSayyajayyau zakyArthe (6.1.81) :</p>

<p class = hn >If the word is to be used in the meaning of 'being capable of', then only it will be जय्य.</p>";

echo "<p class = sa >क्षय्यजय्यौ शक्यार्थे (६.१.८१) :</p>

<p class = hn >यदि जय कर सकने के अर्थ में प्रयोग होता है तभी जय्य होगा ।</p>";

display(0);

}

### /\* krayyastadarthe (6.1.82) \*/

if (sub(array("kre"),array("ya"),blank(0),0) && $pada ==="pratyaya" && $start ===1)

{

$text = two(array("kre"),array("ya"),array("kray"),array("ya"),1);

echo "<p class = sa >By krayyastadarthe (6.1.82) :</p>

<p class = hn >If the word is to be used in the meaning of 'for sale', then only it will be क्रय्य.</p>";

echo "<p class = sa >क्रय्यस्तदर्थे (६.१.८२) :</p>

<p class = hn >यदि बेचने के लिये रखा हुआ इस अर्थ में प्रयोग हो तभी क्रय्य शब्द बनता है </p>";

display(0);

}

### /\* khyatyAtparasya (6.1.112) \*/

if (arr($text,'/[Kt][y][+][a]/') && arr(array($fo),'/[Kt][iI]$/') && in\_array($so,array("Nasi!","Nas")) && $pada==="pratyaya")

{

$text = three(array("Ky","ty"),array("+"),array("a"),array("Ky","ty"),array("+"),array("u"),0);

echo "<p class = sa >By khyatyAtparasya (6.1.112) :</p>";

echo "<p class = sa >ख्यत्यात्परस्य (६.१.११२) :</p>";

display(0);

}

if (sub(array("lUny","kzAmy","prastImy"),array("+"),blank(0),0) && in\_array($so,array("Nasi!","Nas")) && $pada==="pratyaya")

{

$text = three(array("lUny","kzAmy","prastImy"),array("+"),array("a"),array("lUny","kzAmy","prastImy"),array("+"),array("u"),0);

echo "<p class = sa >By khyatyAtparasya (6.1.112) :</p>";

echo "<p class = sa >ख्यत्यात्परस्य (६.१.११२) :</p>";

display(0);

}

### /\* Exceptions to sasajuSo ruH \*/

### /\* etattadoH sulopo'konaJsamAse hali (6.1.132) \*/

if (sub(array("sas","ezas"),$hl,blank(0),0) && !sub(array("asa","anEza"),array("s"),$hl,0) && in\_array($fo,array("sas","ezas")))

{

$text = two(array("sas","ezas"),$hl,array("sa ","eza "),$hl,1);

echo "<p class = sa >By etattadoH sulopo&konaJsamAse hali (6.1.132) :</p>";

echo "<p class = sa >एतत्तदोः सुलोपोऽकोऽनञ्समासे हलि (६.१.१३२)) :</p>";

display(0);

}

### /\* so'ci lope cetpAdapUraNam (6.1.134) \*/

if (sub(array("sa"),array("s"),$ac,0))

{

$text = three(array("sa"),array("s"),$ac,array("sa"),array(""),$ac,1);

echo "<p class = sa >so'ci lope cetpAdapUraNam (6.1.134) :</p>

<p class = hn >N.B. : There is difference of opinion here. vAmana thinks that it applies only to RkpAda. Others think that it applies to zlokapAda also e.g. 'saiSa dAzarathI rAmaH'.</p>";

echo "<p class = sa >सोऽचि लोपे चेत्पादपूरणम्‌ (६.१.१३४) :</p>

<p class = hn >यहाँ मतान्तर है । वामन के मत में यह केवल ऋक्पाद में लागू होता है । अन्यों के मत में यह श्लोकपाद में भी लागू होता है । जैसे कि सैष दाशरथी रामः ।</p>";

display(0);

}

### /\* aniditAM patch for sraMs and dhvaMs \*/

// so that vasusraMsu.. may apply

if (sub(array("sraMs","DvaMs"),blank(0),blank(0),0))

{

$text = one(array("sraMs","DvaMs"),array("sras","Dvas"),0);

echo "<p class = sa >aniditAM hala upadhAyAH kGiti (6.4.24) :</p>";

echo "<p class = sa >अनिदितां हल उपधायाः क्ङिति (६.४.२४) :</p>";

display(0); $aniditAm = 1;

}

### /\* vasusraMsudhvaMsvanaDuhAM daH (8.2.72) \*/

$vasu1 = array("sras","Dvas","anaquh");

if (sub($vasu1,blank(0),blank(0),0) && $pada ==="pada")

{

$text = one(array("sras","Dvas","anaquh"),array("srad","Dvad","anaqud"),0);

echo "<p class = sa >By vasusraMsudhvaMsvanaDuhAM daH (8.2.72) :</p>";

echo "<p class = sa >वसुस्रंसुध्वंस्वनडुहां दः (८.२.७२) :</p>";

display(0);

}

if ((sub(array("vidvas","sedivas","uzivas","Suzruvas","upeyivas","anASvas"),array("+"),blank(0),0)) && $pada ==="pada" && !arr($text,'/[s][+]$/'))

{

$text = one(array("vidvas","sedivas","uzivas","Suzruvas","upeyivas","anASvas"),array("vidvad","sedivad","uzivad","Suzruvad","upeyivad","anASvad"),0);

echo "<p class = sa >By vasusraMsudhvaMsvanaDuhAM daH (8.2.72) :</p>

<p class = hn >N.B. : If 'vas' is used in sense of vasupratyayAnta as in 'vidvas', then only this conversion takes place. Not in cases like 'zivas'.</p>";

echo "<p class = sa >वसुस्रंसुध्वंस्वनडुहाः दः (८.२.७२) :</p>

<p class = hn >यदि वसुप्रत्ययान्त शब्द जैसे कि विद्वस्‌ इत्यादि में यह नियम लागू होता है । शिवस्‌ जैसे शब्दों में नहीं ।</p>";

display(0);

}

### /\* saMyogAnta patch for asmad / yuSmad \*/

// make it more specific to asmad / yuSmad. Right now there is nothing in the condition which restricts its application to asmad / yuSmad.

if (arr($text,'/['.pc('hl').'][s]$/') )

{

$text = last(array("nas","vas"),array("nass","vass"),0);

$text = last(array("s"),array(""),0);

echo "<p class = sa >By saMyogAntasya lopaH (8.2.23) :</p>";

echo "<p class = sa >संयोगान्तस्य लोपः (८.२.२३) :</p>";

display(0);

}

### /\* patch for asiddhatva of SakAra of pratyaya as in pipaThiS \*/

// SakARa of pipaThiS is because of a sUtra which is in tripAdi. so asiddhatva has to apply. We have not come to that sUtra. We take pipaThiS as the base word.

if (arr($text,'/[z][+]/') && $pada==="pada" && sub(array("pipaWiz","ASiz"),array("+"),blank(0),0) )

{

$text = two(array("z"),array("+"),array("r@"),array("+"),0); $R=array\_merge($R,array(1));

echo "<p class = sa >Satva is asiddha to rutva. Therefore sasajuSo ruH applies. </p>";

echo "<p class = sa >रुत्वं प्रति षत्वस्यासिद्धत्वात्‌ ससजुषो रुः इति रुत्वम्‌ ।</p>";

display(0);

$dhatu=1; // This patch is for dhAtus. So $dhatu is made 1.

}

### /\* patch for asiddhatva of SakAra of doS \*/

if (arr($text,'/[z][+]/') && $pada==="pada" && sub(array("doz","Danuz"),array("+"),blank(0),0) )

{

$text = two(array("z"),array("+"),array("r@"),array("+"),0); $R=array\_merge($R,array(1));

echo "<p class = sa >Satva is asiddha to rutva. Therefore sasajuSo ruH applies. </p>";

echo "<p class = sa >रुत्वं प्रति षत्वस्यासिद्धत्वात्‌ ससजुषो रुः इति रुत्वम्‌ ।</p>";

display(0); // this patch is for non dhAtus. therefore $dhatu is not altered.

}

### /\* sasajuSo ruH (8.2.66) \*/

if ((arr($text,'/[s][a][j][u][z][+]/') && $start===1 && $pada ==="pada" ))

{

$text = one(array("z+",),array("r@+",),0); $R=array\_merge($R,array(1));

echo " <p class = sa >By sasajuSo ruH (8.2.66) :</p>";

echo " <p class = sa >ससजुषो रुः (८.२.६६) :</p>";

display(0);

}

if ((arr($text,'/[H][+]/') && $start===1 && $pada ==="pada" )) // for neutralising users' tendency to enter visarga.

{

$text = one(array("H+",),array("r@+",),0); $R=array\_merge($R,array(1));

echo " <p class = hn >You have entered a visarga at the end of the first word. Usually it is derived from a sakAra at the end of the word.</p>";

echo " <p class = hn >आपने प्रथम शब्द के अन्त में विसर्ग का प्रयोग किया है । सामान्यतः यह सकारान्त शब्द से उद्भव होता है ।</p>";

display(0);

}

if (arr($text,'/[aAiIuUfFxXeEoO][s][+]/') && $start===1 && ( $pada ==="pada" || $so==="su!" ))

{

$text = one(array("s+"),array("r@+"),0); $R=array\_merge($R,array(1));

echo " <p class = sa >By sasajuSo ruH (8.2.66) :</p>";

echo " <p class = sa >ससजुषो रुः (८.२.६६) :</p>";

display(0); $r1=1; // 0 - first word doesn't have sasajuSo ruH. 1 - first owrd has sasajuSo ruH.

}

elseif ($start>1 && $r1!==0) { $r1 = 1; } else {$r1=0; }

if ( arr($text,'/['.pc('ac').'][s]$/') && $start===1 )

{

$text = last(array("s"),array("r@"),0);

echo "<p class = sa >By sasajuSo ruH (8.2.66) :</p>";

echo "<p class = sa >ससजुषो रुः (८.२.६६) :</p>";$r2 = 1;

display(0);

}

elseif ( arr($text,'/['.pc('ac').'][+][s]$/') && $start===1 )

{

$text=last(array("s"),array("r@"),0);

// $text = three($ac,array("+"),array("s"),$ac,array("+"),array("r@"),0); $R=array\_merge($R,array(1));

echo "<p class = sa >By sasajuSo ruH (8.2.66) :</p>";

echo "<p class = sa >ससजुषो रुः (८.२.६६) :</p>";

$r2 = 1; // 0 - second word doesn't have sasajuSo ruH. 1 -second word has sasajuSo ruH.

display(0);

}

elseif ($start>1 && $r2!==0)

{

$r2 = 1;

}

else

{

$r2=0;

}

if (arr($text,'/[H]$/') && $start===1 && $pada ==="pada" )

{

$text = last(array("H"),array("r@"),0); $R=array\_merge($R,array(1));

echo " <p class = hn >You have entered a visarga at the end of the second word. Usually it is derived from a sakAra at the end of the word.</p>";

echo " <p class = hn >आपने द्वितीय शब्द के अन्त में विसर्ग का प्रयोग किया है । सामान्यतः यह सकारान्त शब्द से उद्भव होता है ।</p>";

echo "<p class = sa >By sasajuSo ruH (8.2.66) :</p>";

echo "<p class = sa >ससजुषो रुः (८.२.६६) :</p>";

$r1= 1;

display(0);

}

### /\* ahan(8.2.68) and ro'supi (8.2.69) and rUparAtrirathantareSu vAcyam (vA 4847) \*/

$noahan = array("vftrahan","bahuvftrahan");

if (sub(array("ahan","Ahan","AhAn"),array("+"),blank(0),0) && !sub($noahan,blank(0),blank(0),0) && !(in\_array($so,$sup)) && $pada==="pada")

{

if ((strpos($so,"rUp")===0)||(strpos($so,"rAtr")===0)||(strpos($so,"raTantar")===0))

{

$text = one(array("ahan","Ahan","AhAn"),array("ahar@","Ahar@","AhAr@"),0); $R=array\_merge($R,array(1));

echo "<p class = sa >By ahan (8.2.68) and rUparAtrirathantareSu vAcyam (vA 4847).</p>";

echo "<p class = sa >अहन्‌ (८.२.६८) तथा रूपरात्रिरथन्तरेषु वाच्यम्‌ (वा ४८४७) ।</p>";

display(0);

}

else

{

$text = one(array("ahan","Ahan","AhAn"),array("ahar","Ahan","AhAr@"),0);

echo "<p class = sa >ro'supi (8.2.69) :</p>";

echo "<p class = sa >रोऽसुपि (८.२.६९) :</p>";

display(0);

}

}

if ( (arr($text,'/[aA][h][aA][n]$/')||arr($text,'/[aA][h][aA][n][+]/') )&& !sub($noahan,blank(0),blank(0),0) && (in\_array($so,$sup)) && $pada === "pada")

{

echo "<p class = sa >By ahan (8.2.68) :</p>";

echo "<p class = hn >This creates a nipAta for ahan, and shows that the na lopaH prAtipadikAntasya doesn't apply here.</p>";

echo "<p class = sa >अहन्‌ (८.२.६८) :</p>";

echo "<p class = hn >अनेन नलोपाभावं निपात्यते ।</p>";

display(0);

$text = one(array("ahan","Ahan","AhAn"),array("ahar@","Ahar@","AhAr@"),0); $R=array\_merge($R,array(1));

echo "<p class = sa >By ahan (8.2.68) :</p>";

echo "<p class = hn >This mandates rutva here.</p>";

echo "<p class = sa >अहन्‌ (८.२.६८) :</p>";

echo "<p class = hn >अनेन रुर्विधेयः ।</p>";

display(0);

}

### /\* samaH suTi (8.3.5) \*/ // have used @ as mark of anunAsika u of ru.

if (sub(array("sam"),array("s"),array("k"),0))

{

$text = three(array("sam"),array("s"),array("k"),array("saMr@"),array("s"),array("k"),0);

$text = one(array("Mr@"),array("!r@"),1); $R=array\_merge($R,array(1));

echo "<p class = sa >By samaH suTi (8.3.5), atrAnunAsikaH pUrvasya tu vA (8.3.2) and anunAsikAtparo'nusvAraH (8.3.4) :</p>";

echo "<p class = sa >समः सुटि (८.३.५), अत्रानुनासिकः पूर्वस्य तु वा (८.३.२) तथा अनुनासिकात्परोऽनुस्वारः (८.३.४) :</p>";

display(0);

}

### /\* khyAYAdeze na (vA 1591) \*/

if (sub(array("pum"),array("Ky"),blank(0),0))

{

echo "<p class = sa >By khyAYAdeze na (vA 1591) :</p>";

echo "<p class = sa >ख्याञादेशे न (वा १५९१) :</p>";

display(0); $pum = 1; // 0 - pumaH khayyampare is mandatory. 1 - pumaH khayyampare is optional.

} else { $pum = 0; }

### /\* pumaH khayyampare (8.3.6) \*/

$am = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O","h","y","v","r","l","Y","m","N","R","n");

if(sub(array("pum"),prat('Ky'),$am,0) && $pum === 0)

{

$text = three(array("pum"),prat('Ky'),$am,array("puMr@"),prat('Ky'),$am,0);

$text = one(array("Mr@"),array("!r@"),1);

echo "<p class = sa >By pumaH khayyampare (8.3.6), atrAnunAsikaH pUrvasya tu vA (8.3.2) and anunAsikAtparo'nusvAraH (8.3.4) :</p>";

echo "<p class = sa >पुमः खय्यम्परे (८.३.६), अत्रानुनासिकः पूर्वस्य तु वा (८.३.२) तथा अनुनासिकात्परोऽनुस्वारः (८.३.४) :</p>";

display(0);

}

### /\* nazChavyaprazAn (8.3.7) \*/

if (sub(array("n"),prat('Cv'),$am,0) && arr($text,'/[n][+]['.pc('Cv').']/') && $pada ==="pada")

{

$text = three(array("n"),prat('Cv'),$am,array("Mr@"),prat('Cv'),$am,0); $R=array\_merge($R,array(1));

$text = one(array("praSAMr@"),array("praSAn"),0);

$text = one(array("Mr@"),array("!r@"),1);

echo "<p class = sa >By nazChavyaprazAn (8.3.7), atrAnunAsikaH pUrvasya tu vA (8.3.2) and anunAsikAtparo'nusvAraH (8.3.4) :</p>";

echo "<p class = sa >नश्छव्यप्रशान्‌ (८.३.७), अत्रानुनासिकः पूर्वस्य तु वा (८.३.२) तथा अनुनासिकात्परोऽनुस्वारः (८.३.४) :</p>";

display(0);

}

### /\* nRUnpe (8.3.10) \*/

if (sub(array("nFn"),array("p"),blank(0),0) && $pada ==="pada")

{

$text = two(array("nFn"),array("p"),array("nFMr@"),array("p"),0);

$text = one(array("Mr@"),array("!r@"),1);

echo "<p class = sa >By nRUnpe (8.3.10), atrAnunAsikaH pUrvasya tu vA (8.3.2) and anunAsikAtparo'nusvAraH (8.3.4) </p>";

echo "<p class = sa >नॄन्पे (८.३.१०), अत्रानुनासिकः पूर्वस्य तु वा (८.३.२) तथा अनुनासिकात्परोऽनुस्वारः (८.३.४) : </p>";

display(0);

}

### /\* svatavAn pAyau (8.3.11) \*/

if (sub(array("svatavAn"),array("pAyu"),blank(0),0) && $pada ==="pada")

{

$text = two(array("svatavAn"),array("pAyu"),array("svatavA! "),array("pAyu"),0);

echo "<p class = sa >By svatavAn pAyau (8.3.11), atrAnunAsikaH pUrvasya tu vA (8.3.2)</p>";

echo "<p class = sa >स्वतवान्पायौ (८.३.११), अत्रानुनासिकः पूर्वस्य तु वा (८.३.२) : </p>";

display(0);

}

### /\* kAnAmreDite (8.3.12) \*/

if (sub(array("kAn"),array("kAn"),blank(0),0))

{

$text = two(array("kAn"),array("kAn"),array("kAMr@"),array("kAn"),0); $R=array\_merge($R,array(1));

$text = one(array("Mr@"),array("!r@"),1);

echo "<p class = sa >By kAnAmreDite (8.3.12), atrAnunAsikaH pUrvasya tu vA (8.3.2) and anunAsikAtparo'nusvAraH (8.3.4) :</p>";

echo "<p class = sa >कानाम्रेडिते (८.३.१२), अत्रानुनासिकः पूर्वस्य तु वा (८.३.२) तथा अनुनासिकात्परोऽनुस्वारः (८.३.४) :</p>";

display(0);

}

### /\* ato roraplutAdaplute (6.1.113) \*/

if (sub(array("ar@"),array("a"),blank(0),0))

{

$text = two(array("ar@"),array("a"),array("au"),array("a"),0);

echo "<p class = sa >By ato roraplutAdaplute (6.1.113) :</p>";

echo "<p class = sa >अतो रोरप्लुतादप्लुते (६.१.११३) :</p>";

display (0);

}

### /\* hazi ca (6.1.114) \*/

if (sub(array("a"),array("r@"),prat('hS'),0))

{

$text = three(array("a"),array("r@"),prat('hS'),array("a"),array("u"),prat('hS'),0);

echo "<p class = sa >By hazi ca (6.1.114) :</p>";

echo "<p class = sa >हशि च (६.१.११४) :</p>";

display (0);

}

### /\* ekaH pUrvaparayoH (6.1.84) \*/

// This is the adhikArasUtra. No vidhi mentioned.

// The following vArtikas are exception to AdguNaH. Otherwise after joining, it will be difficult to identify. So coded here.

### /\* akSAdUhinyAmupasaMkhyAnam (vA 3604) \*/

### /\* svAdireriNoH (vA 3606) \*/

### /\*prAdUhoDhoDyeSaiSyeSu (vA 3605) \*/

### /\* Rte ca tRtIyAsamAse (vA 3607) \*/

### /\* pravatsatarakambalavasanadazArNAnAmRNe (vA 3608-9) \*/

$v1 = array('akza','sva','pra','pra','pra','pra','suKa','pra','vatsatara','kambala','vasana','daSa','fRa','sva');

$v2 = array('Uhin','ir','Uh','UQ','ez','ezy','ft','fR','fR','fR','fR','fR','fR','Ir');

$v3 = array('akz','sv','pr','pr','pr','pr','suK','pr','vatsatar','kambal','vasan','daS','fR','sv');

$v4 = array('OhiR','Er','Oh','OQ','Ez','Ezy','Art','ArR','ArR','ArR','ArR','ArR','ArR','Er');

if (sub($v1,$v2,blank(0),0) && $pada === "pada")

{

$text = two($v1,$v2,$v3,$v4,0);

echo "<p class = sa >Applying the following vArtikas : akSAdUhinyAmupasaMkhyAnam (vA 3604), svAdireriNoH (vA 3606), prAdUhoDhoDyeSaiSyeSu (vA 3605), Rte ca tRtIyAsamAse (vA 3607), pravatsatarakambalavasanadazArNAnAmRNe (vA 3608-9)</p>";

echo "<p class = sa >अक्षादूहिन्यामुपसंख्यानम्‌ (वा ३६०४), स्वादेरेरिणोः (वा ३६०६), प्रादूहोढोढ्येषैष्येषु (वा ३६०५), ऋते च तृतीयासमासे (वा ३६०७), प्रवत्सतरकम्बलवसनदशार्णानामृणे (वा ३६०८-०९)</p>";

display(0);

}

### /\* etyedhatyuThsu (6.1.89) \*/

if (sub(array("a","A"),array("eti","ezi","emi","etu","Et","EtAm","EH","Es","Etam","Eta","Eva","Ema","ezyati","Ezyati","etA","eD","ED","Uh"),blank(0),0))

{

$text = two (array("a","A"),array("eti","ezi","emi","etu","Et","EtAm","EH","Es","Etam","Eta","Eva","Ema","ezyati","Ezyati","etA","eD","ED","Uh"),blank(2),array("Eti","Ezi","Emi","Etu","Et","EtAm","EH","Es","Etam","Eta","Eva","Ema","Ezyati","Ezyati","EtA","ED","ED","Oh"),0);

echo "<p class = sa >By etyedhatyuThsu (6.1.89) :</p>";

echo "<p class = sa >एत्येधत्यूठ्सु (६.१.८९) :</p>";

display(0);

}

### /\* AdguNaH (6.1.87) \*/

$forguna = array("i","I","u","U"); // the vowels which take guNa prakriyA. iko guNavRddhi.

$rep = array("e","e","o","o"); // replacement of guNa.

if (sub($aa,$forguna,blank(0),0))

{

$text = two($aa,$forguna,blank(2),$rep,0);

echo "<p class = sa >By AdguNaH (6.1.87) :</p>";

echo "<p class = sa >आद्गुणः (६.१.८७) :</p>";

display(0);

}

### /\* uraNraparaH (1.1.51) \*/

$forguna = array("f","F","x","X");

$rep = array("ar","ar","al","al");

if (sub($aa,$forguna,blank(0),0))

{

$text = two($aa,$forguna,blank(2),$rep,0);

echo "<p class = sa >By AdguNaH (6.1.87) and uraNraparaH (1.1.51) :</p>";

echo "<p class = sa >आद्गुणः (६.१.८७) तथा उरण्रपरः (१.१.५१) :</p>";

display(0);

}

### /\* eGi pararUpam (6.1.94) \*/

// Added it here because it is exception to vRddhireci.

for($i=0;$i<count($akarantaupasarga);$i++) // $akarantaupasarga - see function.php

{

$a\_upa\_without\_a[$i] = substr($akarantaupasarga[$i],0,count(str\_split($akarantaupasarga[$i]))-1);

}

if (sub($akarantaupasarga,prat('eN'),blank(0),0) && arr($text,'/[I][y]/') && in\_array($first,$akarantaupasarga))

{

$text = two($akarantaupasarga,prat('eN'),$a\_upa\_without\_a,prat('eN'),1);

echo "<p class = sa >By eGi pararUpam (6.1.94) and anuvRtti of vA supi :</p>";

echo "<p class = sa >एङि पररूपम्‌ (६.१.९४) तथा वा सुपि की अनुवृत्ति :</p>";

display(0);

}

elseif (sub($akarantaupasarga,prat('eN'),blank(0),0) && in\_array($first,$akarantaupasarga))

{

$text = two($akarantaupasarga,prat('eN'),$a\_upa\_without\_a,prat('eN'),0);

echo "<p class = sa >By eGi pararUpam (6.1.94) :</p>";

echo "<p class = sa >एङि पररूपम्‌ (६.१.९४) :</p>";

display(0);

}

### /\* eve cAniyoge (vA 3631) \*/

if (sub($aa,array("eva"),blank(0),0))

{

$text = two($aa,array("eva"),blank(2),array("eva"),1);

echo "<p class = sa >By eve cAniyoge (vA 3631) :</p>

<p class = hn >N.B. that the optionality applies only in case the eva is used for avadhAraNa.</p>" ;

echo "<p class = sa >एवे चावधारणे (वा ३६३१) :</p>

<p class = hn >जब 'एव' अवधारण के अर्थ में प्रयुक्त हुआ हो, तभी यह नियम लागू होता है ।</p>" ;

display(0);

}

### /\* vA supyapizaleH (6.1.92) \*/

// Not possible to know what is nAmadhAtu and what is not. Therefore added as comments. Not coded.

### /\* aco'ntyAdi Ti (1.1.64) \*/

// a saJjJAsUtra. No vidhi mentioned.

### /\* otvoShThayoH samAse vA (vA 3634) \*/

if (sub($aa,array("otu","ozW"),blank(0),0))

{

$text = two($aa,array("otu","ozW"),blank(2),array("otu","ozW"),1);

echo "<p class = sa >By otvoShThayoH samAse vA (vA 3634) :</p>

<p class = hn >If what you entered is a samAsa, it will be optionally converted. Otherwise ignore the pararUpa form.</p>";

echo "<p class = sa >ओत्वोष्ठ्योः समासे वा (वा ३६३४) :</p>

<p class = hn >यदि समास है तभी यह नियम लागू होगा । अन्यथा पररूप वाला रूप नहीं बनेगा ।</p>";

display(0);

}

### /\* nAmreDitasyAntasya tu vA (6.1.99), tasya paramAmreDitam (8.1.2) \*/

for($i=0;$i<count($text);$i++)

{

$tttt = explode("at",$text[$i]); // exploding the word to check whether it is AmreDita. e.g. paTatpaTat will be split - paT. at, paT, at.

if (count($tttt) > 1 )

{

if ($tttt[0] === $tttt[1]) // e.g. paT - paT.

{

$amredita = 1; // 0 - no AmreDita. 1 - AmreDita.

break;

}

else

{

$amredita = 0;

}

}

else

{

$amredita = 0;

}

}

if ( $amredita === 1 && $start === 1)

{

$text = two(array("at"),array("iti"),array("a"),array("iti"),1); // e.g. paTatpaTiti, paTatpaTatiti.

echo "<p class = sa >nAmreDitasyAntasya tu vA (6.1.99), tasya paramAmreDitam (8.1.2) :</p>

<p class = hn >When the 'at' happens to be at the end of an onaematopic word and it is followed by 'iti', its 'Ti' is elided. This rule doesn't apply on single vowel words like 'zrat'.</p>";

echo "<p class = sa >नाम्रेडितस्यान्तस्य तु वा (६.१.९९), तस्य परमाम्रेडितम्‌ (८.१.२) :</p>

<p class = hn >आम्रेडित में यह नियम लागू वैभाषिक तौर से होता है, और अन्तिम तकार का होता है । एकाच्‌ शब्दों में, जैसे कि श्रत्‌, यह लागू नहीं होता ।</p>";

display(0);

}

### /\* avyaktAnukaraNasyAta itau (6.1.98) \*/

$ff = preg\_split('/[aAiiuUfFxXeEoO]/',$first); // checking whether the first word has more than two vowels.

if (sub(array("at"),array("iti"),blank(0),0) && $amredita === 0 && $start ===1 && count($ff)>2)

{

$text = two($array("at"),array("iti"),blank(1),array("iti"),0);

echo "<p class = sa >By avyaktAnukaraNasyAta itau (6.1.98) :</p>

<p class = hn > When the 'at' happens to be at the end of an onaematopic word and it is followed by 'iti', its 'Ti' is elided. This rule doesn't apply on single vowel words like 'zrat'.</p>";

echo "<p class = sa >अव्यक्तानुकरणस्यात इतौ (६.१.९८) :</p>

<p class = hn >अव्यक्तानुकरण में प्रयुक्त हुए शब्द के अत्‌ के बाद में यदि इति हो तो यह नियम लागू होता है । एकाच्‌ शब्दों में, जैसे कि श्रत्‌, यह लागू नहीं होता ।</p>";

display(0);

}

### /\* vRddhireci (6.1.88) \*/

$aa = array("a","A"); // a and A

$vrrdhi = array("E","O","E","O","E","O","E","O"); // vRddhi of 'ec'.

if (sub($aa,prat("ec"),blank(0),0) )

{

$text = one(array("a+e","a+E","a+o","a+O","A+e","A+E","A+o","A+O"),array("E+","E+","O+","O+","E+","E+","O+","O+"),0);

$text = two($aa,prat('ec'),blank(2),$vrrdhi,0);

echo "<p class = sa >By vRddhireci (6.1.88) :</p>";

echo "<p class = sa >वृद्धिरेचि (६.१.८८) :</p>";

display(0);

$vriddhireci=1; // 0 - this sUtra has not applied. 1 - this sUtra has appied.

} else {$vriddhireci=0; }

### /\* udaH sthAstambhvoH pUrvasya (8.1.61) \*/

if(sub(array("ud","ut"),array("sTA","stam"),blank(0),0))

{

$text = two(array("ud","ut"),array('sTA','stam'),array("ud","ut"),array('TTA','Ttam'),0);

echo "<p class = sa >By udaH sthAstambhvoH pUrvasya (8.1.61) :</p>";

echo "<p class = sa >उदः स्थास्तम्भ्वोः पूर्वस्य (८.१.६१) :</p>";

display(0);

}

### /\* saMhitAyAm (6.1.72) \*/

// This is adhikArasUtra. Nothing to code here.

### /\* Che ca (6.1.73) \*/

if (sub($hrasva,array("C"),blank(0),0))

{

$text = two(array("a","i","u","f","x"),array("C"),array("at","it","ut","ft","xt"),array("C"),0);

echo "<p class = sa >By Che ca (6.1.73) :</p>";

echo "<p class = sa >छे च (६.१.७३) :</p>";

display(0);

}

### /\* AGmAGozca (6.1.74) \*/

if (($first === "A" || $first === "mA") && $start===1)

{

$text = two(array("A"),array("C"),array("At"),array("C"),0);

echo "<p class = sa >By AGmAGozca (6.1.74) :</p>";

echo "<p class = sa >आङ्माङोश्च (६.१.७४) :</p>";

display(0);

}

### /\* dIrghAt (6.1.75) and padAntAdvA (6.1.76) \*/

if (sub($dirgha,array("C"),blank(0),0) && $pada === "pratyaya" && $start === 1) // for $dirgha see function.php

{

$text = two($dirgha,array("C"),array("At","It","Ut","Ft","Xt","et","Et","ot","Ot"),array("C"),0);

echo "<p class = sa >By dIrghAt (6.1.75) padAntAdvA (6.1.76) :</p>

<p class = hn >N.B.: The 'tuk' Agama is optional in case the preceding dIrgha vowel is at the padAnta. Otherwise, it is mandatory to add.</p>";

echo "<p class = sa >दीर्घात्‌ (६.१.७५) तथा पदान्ताद्वा (६.१.७६) :</p>

<p class = hn >यदी दीर्घ स्वर पदान्त में हो तब तुक्‌ आगम लगाना पाक्षिक है । अन्यथा यह आवश्यक है ।</p>";

display(0);

}

if (sub($dirgha,array("C"),blank(0),0) && $pada === "pada" && $start ===1)

{

$text = two($dirgha,array("C"),$dirghata,array("C"),1);

echo "<p class = sa >By dIrghAt (6.1.75) padAntAdvA (6.1.76) :</p>

<p class = hn >N.B.: The 'tuk' Agama is optional in case the preceding dIrgha vowel is at the padAnta. Otherwise, it is mandatory to add.</p>";

echo "<p class = sa >दीर्घात्‌ (६.१.७५) तथा पदान्ताद्वा (६.१.७६) :</p>

<p class = hn >यदी दीर्घ स्वर पदान्त में हो तब तुक्‌ आगम लगाना पाक्षिक है । अन्यथा यह आवश्यक है ।</p>";

display(0);

}

### /\* yasmAtpratyayavidhistadAdi pratyaye'Ggam (2.4.13) \*/

// Pending to code.

### /\* creating a + elision for two ++ simultaneously. \*/

$text = one(array("++"),array("+"),0);

// adding 1 to $start. All sUtras where $start===1 has been mentioned in the condition won't apply in the second loop.

$start++;

}

while ($text !== $original); // looping till all the applicable sUtras of sapAdasaptAdhyAyI are exhausted. i.e. the original ($original) and the output ($text) are the same.

### /\* tripAdI functions \*/

// tripAdI functions are always applied in serial manner because of 'pUrvatrAsiddham'. Therefor there is no possibility of a sUtra being applied out of turn (Except explicitly mentioned in grammar books.)

// Therefore we have kept them in their order as per aSTAdhyAyI. Where there is violation, we have placed a patch.

### /\* na NisambuddhyoH (8.2.8) \*/

if (arr($text,'/[n][+]$/') && ( in\_array($so,array("Ni")) || (in\_array($so,array("su!")) && $sambuddhi===1)) && $bham===0 && $shi===0 && $ikoci===0 )

{

echo "<p class = sa >By na NisambuddhyoH (8.2.8) :</p>";

echo "<p class = sa >न ङिसम्बुद्ध्योः (८.२.८) :</p>";

display(0);

$Gisambu=1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else {$Gisambu=0; }

### /\* patches for maghavA bahulam \*/

if (sub(array("maGavAn"),blank(0),blank(0),0) && arr($text,'/[v][a][n][+]$/') && in\_array($so,$sup) && $bham===0 && $shi===0 && $ikoci===0 && $Gisambu===0 && $sambuddhi===0)

{

$text = two(array("maGavAn"),array("+"),array("maGavA"),array("+"),0);

echo "<p class = sa >By na lopaH prAtipadikAntasya (8.2.7) :</p>";

echo "<p class = sa >न लोपः प्रातिपदिकान्तस्य (८.२.७) :</p>";

display(0);

}

/\*if (sub(array("maGavan","Ahar@","Ahan"),blank(0),blank(0),0) && in\_array($so,$sup) && $bham===0 && $shi===0 && $ikoci===0 && $Gisambu===0 && $sambuddhi===0)

{

$text = two(array("maGavan","Ahar@","Ahan"),array("+"),array("maGavAn","AhAr@","AhAn"),array("+"),0);

echo "<p class = sa >By sarvanAmasthAne cAsambuddhau (6.4.8) :</p>";

echo "<p class = hn >Because of bahulagrahaNa in maghavan, saMyogAntasya lopaH is not asiddha here. :</p>";

echo "<p class = sa >सर्वनामस्थाने चासम्बुद्धौ (६.४.८) :</p>";

echo "<p class = hn >मघवा बहुलं के बहुलग्रहण के कारण, संयोगान्तलोप असिद्ध नहीं है । </p>";

display(0);

}\*/

### /\* NAvuttarapade pratiSedho vaktavyaH (vA 4785) \*/

// Pending because involves samAsa. Out of purview right now.

### /\* sambuddhau napuMsakAnAM nalopo vA vAcyaH (vA 4786) \*/

$napumsakanalopa=array(); // when there is nalopa in napuMsakaliGga, 1 is added in the array. useful for na lopaH prAtipadikAntasya.

if (arr($text,'/[n][+]$/') && in\_array($so,$sup) && $bham===0 && $shi===0 && $ikoci===0 && $so==="su!" && $sambuddhi===1 && $gender==="n")

{

$text = two(array("n"),array("+"),array(""),array("+"),1);

echo "<p class = sa >By sambuddhau napuMsakAnAM nalopo vA vAcyaH (vA 4786) :</p>";

echo "<p class = sa >सम्बुद्धौ नपुंसकानां नलोपो वा वाच्यः (वा ४७८६) :</p>";

display(0);

$napumsakanalopa=array\_merge($napumsakanalopa,array(1));

}

### /\* na lopaH prAtipadikAntasya (8.2.7) \*/

// parame vyoman is pending. Vedic in nature.

$nalopa=0; // 0 - no nalopa, is default. 1 - there is nalopa. Useful to display nalopaH supsvarasaJjJAtugvidhiSu kRti (8.2.2).

if (arr($text,'/[n][+]$/') && $nopadha===1 && !in\_array(1,$napumsakanalopa) )

{

$text = two(array("n"),array("+"),array(""),array("+"),0);

echo "<p class = sa >By na lopaH prAtipadikAntasya (8.2.7) :</p>";

echo "<p class = sa >न लोपः प्रातिपदिकान्तस्य (८.२.७) :</p>";

display(0);

$nalopa=1;

}

elseif (arr($text,'/[n][+]$/') && in\_array($so,$sup) && $pada="pada" && $shi===0 && $ikoci===0 && $Gisambu===0 && !in\_array(1,$napumsakanalopa))

{

$text = two(array("n"),array("+"),array(""),array("+"),0);

echo "<p class = sa >By na lopaH prAtipadikAntasya (8.2.7) :</p>";

echo "<p class = sa >न लोपः प्रातिपदिकान्तस्य (८.२.७) :</p>";

display(0);

$nalopa=1;

}

if (arr($text,'/[n][+]['.pc('hl').']/') && ( $astana===1 || $Satcatur===1) && !in\_array(1,$napumsakanalopa))

{

$text = two(array("n"),array("+"),array(""),array("+"),0);

echo "<p class = sa >By na lopaH prAtipadikAntasya (8.2.7) :</p>";

echo "<p class = sa >न लोपः प्रातिपदिकान्तस्य (८.२.७) :</p>";

display(0);

$nalopa=1;

}

elseif (arr($text,'/[n][+]['.pc('hl').']/') && !arr($text,'/[n][+][e]/') && ($pada==="pada"|| $so==="Am" ) && in\_array($so,$sup) && $bham===0 && $shi===0 && $ikoci===0 && $Gisambu===0 && !in\_array(1,$napumsakanalopa))

{

$text = two(array("n"),array("+"),array(""),array("+"),0);

echo "<p class = sa >By na lopaH prAtipadikAntasya (8.2.7) :</p>";

echo "<p class = sa >न लोपः प्रातिपदिकान्तस्य (८.२.७) :</p>";

display(0);

$nalopa=1;

}

### /\* nalopaH supsvarasaJjJAtugvidhiSu kRti (8.2.2) \*/

if ($nalopa===1)

{

echo "<p class = pa >By nalopaH supsvarasaJjJAtugvidhiSu kRti (8.2.2) :</p>";

echo "<p class = pa >नलोपः सुप्स्वरसञ्ज्ञातुग्विधिषु कृति (८.२.२) :</p>";

display(0);

}

### /\* patch for udan Adeza of udaka \*/

if (sub(array("udan","Asan"),array("+"),array("ByAm","Bir","Byar","su")))

{

$text = two(array("udan","Asan"),array("+"),array("uda","Asa"),array("+"),0);

echo "<p class = sa >By na lopaH prAtipadikAntasya (8.2.7) :</p>";

echo "<p class = sa >न लोपः प्रातिपदिकान्तस्य (८.२.७) :</p>";

display(0);

}

### /\* nazervA (8.2.63) \*/

if (arr($text,'/[n][a][S][+]/') && $pada==="pada")

{

$text = one(array("naS+",),array("nak+",),1);

echo " <p class = sa >By nazervA (8.2.63) :</p>";

echo " <p class = sa >नशेर्वा (८.२.६३) :</p>";

display(0);

}

### /\* kvinpratyayasya kuH (8.2.62) defining \*/

if (sub(array("S","z","s"),array("+"),blank(0),0) && ( $kvin===1 || ($kvip===1 && $fo==="dfS") ) && $pada==="pada" && $Asarva===1 && $goanc===0)

{

$kvinku=1; // 0 - not eligible for kvinpratyasya kuH. 1 - eligible for kvinpratyayasya kuH.

}

elseif(sub($hl,array("+"),blank(0),0) && ( $kvin===1 || ($kvip===1 && $fo==="dfS") ) && $pada==="pada" && !sub(array("S","z","s"),array("+"),blank(0),0) && $goanc===0)

{

$kvinku=1;

}

else

{

$kvinku=0;

}

/\*if (sub(array("S","z","s"),array("+"),blank(0),0) && ( $kvin===1 || ($kvip===1 && $fo==="diS") ) && $pada==="pada" && $Asarva===1)

{

$text1 = $text;

$text2 = two(array("S","z","s"),array("+"),array("K","k","k"),array("+"),0); // check for z and s.

$text = $text2;

echo "<p class = pa >There is difference of opinion here. According to one school, vrazca.. applies and z->S->g->g,k is done. According to the other school, kvinpratyayasya kuH is apavAda of Satva. Therefore z->K->K,k happens.</p>";

echo "<p class = pa >कुत्वस्यासिद्धत्वाद्‌ 'व्रश्च..' इति षः, तस्य जश्त्वे डः । तस्य कुत्वेन गः । तस्य चर्त्वेन पक्षे कः । तादृग्‌, तादृश्‌ ॥ 'षत्वापवादत्वात्‌ कुत्वेन खकारः' इति कैयटहरदत्तादिमते तु चर्त्वाभावपक्षे ख एव श्रूयते, न तु गः, जश्त्वं प्रति कुत्वस्य असिद्धत्वात्‌ ॥</p>";

display(0);

$text= $text1;

$kvinku=1;

}

elseif (sub($hl,array("+"),blank(0),0) && ( $kvin===1 || ($kvip===1 && $fo==="diS") ) && $pada==="pada" && !sub(array("S","z","s"),array("+"),blank(0),0))

{

$text = two($cu,array("+"),$ku,array("+"),0);

$text = two($Tu,array("+"),$ku,array("+"),0);

$text = two($tu,array("+"),$ku,array("+"),0);

$text = two($pu,array("+"),$ku,array("+"),0);

$text = two(array("h"),array("+"),array("g"),array("+"),0);

echo "<p class = sa >By kvinpratyayasya kuH (8.2.62) :</p>";

echo "<p class = sa >क्विन्प्रत्ययस्य कुः (८.२.६२) :</p>";

display(0); $kvinku=1;

}

else { $kvinku=0; }

\*/ // bracketed. Output seems fine after bracketing. Kept in case we need it in future.

### /\* vrazcabhrasjamRjayajarAjabhrAjacChazAM ca (8.2.35) \*/

// TubhrAjR dIptau and ejR bhejR bhrAjR dIptau are different. This is pending to code.

// parau vrajeH SaH padAnte (u 217) pending.

$vrasca = array("vfSc","sfj","mfj","yaj","rAj","BrAj","devej","parivrAj","Bfj","ftvij");

$vrashca = array("vfSz","sfz","mfz","yaz","rAz","BrAz","devez","parivrAz","Bfz","ftviz");

if ( (sub($vrasca,array("+"),prat("Jl"),0) || ( sub($vrasca,array("+"),blank(0),0) && $pada==="pada")) && $\_GET['cond1\_9\_3']!=="2" && ($kvinku===0 || ($fo==="asfj" && in\_array($so,array("su!","am")))) )

{

if (sub($vrasca,prat('Jl'),blank(0),0))

{

$text = two($vrasca,prat('Jl'),$vrashca,prat("Jl"),0);

}

else

{

$text = one($vrasca,$vrashca,0);

}

echo "<p class = sa >By vrazcabhrasjasRjamRjayajarAjabhrAjacChazAM SaH (8.2.35) :</p>";

echo "<p class = sa >व्रश्चभ्रस्जसृजमृजयजराजभ्राजच्छशां षः (८.२.३५) :</p>";

display(0);

$vras1 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $vras1 = 0; }

if (arr($text,'/[CS]$/'))

{

if ($dhatu===1)

{

if ($kvinku===1 && $Asarva===1 && arr(array($fo),'/[S]$/'))

{

$text = last(array("C","S"),array("z","z"),0);

}

else

{

$text = last(array("C","S"),array("z","z"),0);

}

echo "<p class = sa >By vrazcabhrasjasRjamRjayajarAjabhrAjacChazAM SaH (8.2.35) :</p>";

echo "<p class = sa >व्रश्चभ्रस्जसृजमृजयजराजभ्राजच्छशां षः (८.२.३५) :</p>";

}

else

{

if ($kvinku===1 && $Asarva===1 && arr(array($fo),'/[S]$/'))

{

// $text = last(array("C","S"),array("z","z"),1);

$text = last(array("C","S"),array("z","z"),0);

}

else

{

// $text = last(array("C","S"),array("z","z"),1);

$text = last(array("C","S"),array("z","z"),0);

}

echo "<p class = sa >By vrazcabhrasjasRjamRjayajarAjabhrAjacChazAM SaH (8.2.35) :</p>";

echo "<p class = hn >Some people hold that there is anuvRtti of 'dhAtoH' here. In that case Satva won't happen. It is optional.</p>";

echo "<p class = sa >व्रश्चभ्रस्जसृजमृजयजराजभ्राजच्छशां षः (८.२.३५) :</p>";

echo "<p class = hn >केचित्तु व्रश्चादिसूत्रे 'दादेर्धातोः इति सूत्रात्‌ 'धातोः' इत्यनुवर्तयन्ति । तन्मते षत्वं न भवति ।</p>";

}

display(0);

$vras3 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $vras3 =0; }

if (arr($text,'/[CS][+]/') && $pada === "pada")

{

if ($dhatu===1)

{

if ($kvinku===1 && $Asarva===1 && arr(array($fo),'/[S]$/'))

{

$text = two(array("C","S"),array("+"),array("z","z"),array("+"),0);

}

else

{

$text = two(array("C","S"),array("+"),array("z","z"),array("+"),0);

}

echo "<p class = sa >By vrazcabhrasjasRjamRjayajarAjabhrAjacChazAM SaH (8.2.35) :</p>";

echo "<p class = sa >व्रश्चभ्रस्जसृजमृजयजराजभ्राजच्छशां षः (८.२.३५) :</p>";

}

else

{

if ($kvinku===1 && $Asarva===1 && arr(array($fo),'/[S]$/'))

{

$text2=$text;

$text=$text1;

// $text = two(array("C","S"),array("+"),array("z","z"),array("+"),1);

$text = two(array("C","S"),array("+"),array("z","z"),array("+"),0);

$text = array\_merge($text,$text2);

$text = array\_unique($text);

$text1=array(); $text2=array();

}

else

{

// $text = two(array("C","S"),array("+"),array("z","z"),array("+"),1);

$text = two(array("C","S"),array("+"),array("z","z"),array("+"),0);

}

echo "<p class = sa >By vrazcabhrasjasRjamRjayajarAjabhrAjacChazAM SaH (8.2.35) :</p>";

echo "<p class = hn >Some people hold that there is anuvRtti of 'dhAtoH' here. In that case Satva won't happen. It is optional.</p>";

echo "<p class = sa >व्रश्चभ्रस्जसृजमृजयजराजभ्राजच्छशां षः (८.२.३५) :</p>";

echo "<p class = hn >केचित्तु व्रश्चादिसूत्रे 'दादेर्धातोः इति सूत्रात्‌ 'धातोः' इत्यनुवर्तयन्ति । तन्मते षत्वं न भवति ।</p>";

}

display(0);

$vras4 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $vras4 = 0; }

### /\* nimittApAye naimittikasyApyapAyaH (paribhASA) \*/

if (($vras1===1 && sub(array("vfSz"),blank(0),blank(0),0)) || (($vras3 ===1 || $vras4 ===1) && sub(array("cz"),blank(0),blank(0),0)))

{

$text = one(array("vfSz"),array("vfsz"),0);

$text = one(array("cz"),array("z"),0);

echo "<p class = sa >By nimittApAye naimittikasyApyapAyaH (paribhASA) :</p>";

echo "<p class = sa >निमित्तापाये नैमित्तिकस्याप्यपायः (परिभाषा) :</p>";

display(0);

}

### /\* vivikz patch for overcoming skoH saMyogAdyorante ca \*/

if (sub(array("vivikz"),array("+"),blank(0),0) && $pada==='pada')

{

$text = two(array("vivikz"),array("+"),array("viviS"),array("+"),0);

echo "<p class = sa >As katva is asiddha to skoH saMyogAdyorante ca, saMyogAntalopa happens. 'S' is changed to 's' by nimittApAye naimittikasyApyapAyaH. :</p>";

echo "<p class = sa >स्कोः संयोगाद्योरन्ते च इति कलोपे प्राप्ते कत्वस्य असिद्धत्वात्‌ संयोगान्तलोपः । सकारस्य लोपे 'निमित्ताभावे नैमित्तिकस्याप्यपायः' इति षत्वमपि निवर्तते । :</p>";

display(0);

$text = two(array("viviS"),array("+"),array("viviz"),array("+"),0);

echo "<p class = sa >By vrazcabhrasjasRjamRjayajarAjabhrAjacChazAM SaH (8.2.35) :</p>";

echo "<p class = sa >व्रश्चभ्रस्जसृजमृजयजराजभ्राजच्छशां षः (८.२.३५) :</p>";

display(0);

}

### /\* rakS, takS patch to bar application of skoH saMyogAdyorante ca \*/

if (sub(array("takz","rakz"),array("+"),blank(0),0) && $pada === "pada" && $Nyanta===1)

{

echo "<p class = pa >skoH saMyogAdyorante ca doesn't apply here because of sthAnivadbhAva of Nilopa.</p>";

echo "<p class = hn >'pUrvatrAsiddhe na sthAnivat' (vA 433) doesn't apply here, because it is overruled by 'tasya doSaH saMyogAdilopalatvaNatveSu (vA 440).</p>";

echo "<p class = pa >तक्षिरक्षिभ्यां ण्यन्ताभ्यां क्विपि तु 'स्कोः..' इति न प्रवर्तते । णिलोपस्य स्थानिवद्भावात्‌ ।</p>";

echo "<p class = hn >'पूर्वत्रासिद्धे न स्थानिवत्‌' (वा ४४३) इह नास्ति । 'तस्य दोषः संयोगादिलोपलत्वणत्वेषु (वा ४४०) इति निषेधात्‌ ।</p>";

display(0);

$rakS=1; // 0 - doesn't prevent skoH saMyogAdyorante ca. 1 - prevents skoH saMyogAdyorante ca.

} else { $rakS=0; }

### /\* pipak, vivak, didhak patch to bar application of skoH saMyogAdyorante ca \*/

//if (sub(array("vivakz","diDakz","pipakz"),array("+"),blank(0),0) && $pada === "pada" && $san===1) // removed for checking whether $san makes any difference.

if (sub(array("vivakz","diDakz","pipakz"),array("+"),blank(0),0) && $pada === "pada")

{

echo "<p class = pa >skoH saMyogAdyorante ca doesn't apply here because kutva is asiddha to it.</p>";

echo "<p class = pa >'स्कोः...' इति कलोपं प्रति कुत्वस्य असिद्धत्वात्‌ संयोगान्तलोपः ।</p>";

display(0);

$pipakS=1; // 0 - doesn't prevent skoH saMyogAdyorante ca. 1 - prevents skoH saMyogAdyorante ca.

} else { $pipakS=0; }

### /\* skoH saMyogAdyorante ca (8.2.29) \*/

if ((sub(array("s","k"),$hl,prat("Jl"),0) || arr($text,'/[sk]['.flat($hl).'][+]$/')) && $pada === "pada" && $rakS===0 && $pipakS===0)

{

$text = three(array("s","k"),$hl,prat("Jl"),array("",""),$hl,prat("Jl"),0);

$text = three($ac,array("s","k"),$hl,$ac,array("",""),$hl,0);

echo "<p class = sa >By skoH saMyogAdyorante ca (8.2.29) :</p>";

echo "<p class = sa >स्कोः संयोगाद्योरन्ते च (८.२.२९) :</p>";

display(0);

}

### /\* patch for cikIrz \*/

if (sub(array("cikIrz"),array("+"),blank(0),0) && ($pada==="pada" || arr($text,'/[r][z][+]$/')) )

{

$text = two(array("cikIrz"),array("+"),array("cikIr"),array("+"),0);

echo "<p class = sa >By rAtsasya (8.2.24) :</p>";

echo "<p class = sa >रात्सस्य (८.२.२४) :</p>";

display(0);

}

### /\* rAtsasya (8.2.24) \*/

if ((arr($text,('/[r][+][s]$/')) && $pada === "pratyaya") || (arr($text,('/[r][s][+]/')) && $pada === "pada") )

{

$text = one(array("r+s"),array("r"),0);

$text = two(array("rs"),array("+"),array("r"),array("+"),0);

echo "<p class = sa >By rAtsasya (8.2.24) :</p>";

echo "<p class = sa >रात्सस्य (८.२.२४) :</p>";

display(0);

}

if ((arr($text,('/[r][+][hyvrlYmGRnJBGQDjbgqdKPCWTcwtkpzS]$/')) && $pada === "pratyaya") || (arr($text,('/[r][hyvrlYmGRnJBGQDjbgqdKPCWTcwtkpzS][+]/')) && $pada === "pada") )

{

echo "<p class = pa >rAtsasya (8.2.24) prevents application of saMyogAntasya lopaH.</p>";

echo "<p class = pa >रात्सस्य (८.२.२४) से संयोगान्तस्य लोपः का प्रतिषेध होता है ।</p>";

display(0);

$ratsasya=1; // 0 - doesn't prevent saMyogAntasya lopaH. 1 - prevents saMyogAntasya lopaH.

} else { $ratsasya=0; }

### /\* saMyogAntasya lopaH (8.2.23) \*/

// coding pending because not clear. And also 'yaNaH pratiSedho vAcyaH' prohibits its application.

if ( ( sub(array("N"),$ku,array("+"),0) || sub(array("Y"),$cu,array("+"),0) || sub(array("R"),$Tu,array("+"),0) ||sub(array("m"),$pu,array("+"),0) ) && $ratsasya===0 && $pada==="pada" && $vriddhireci===0 ) // patch for nimittApAye naimittikasyApAyaH.

{

$text = three(array("N"),$ku,array("+"),array("n"),blank(count($ku)),array("+"),0);

$text = three(array("Y"),$cu,array("+"),array("n"),blank(count($cu)),array("+"),0);

$text = three(array("R"),$Tu,array("+"),array("n"),blank(count($Tu)),array("+"),0);

$text = three(array("m"),$pu,array("+"),array("n"),blank(count($pu)),array("+"),0);

echo "<p class = sa >By saMyogAntasya lopaH (8.2.23) and nimittApAye naimittikasyApyapAyaH (pa) :</p>";

echo "<p class = sa >संयोगान्तस्य लोपः (८.२.२३) तथा निमित्तापाये नैमित्तिकस्याप्यपायः (प) :</p>";

display(0);

}

elseif ( sub($hl,$hl,array("+"),0) && $bham===0 && $pada==="pada" && $ratsasya===0 && $vriddhireci===0 && !arr($text,'/['.pc('hl').']['.pc('hl').'][+]['.pc('ac').']/'))

{

$text = three($hl,$hl,array("+"),$hl,blank(count($hl)),array("+"),0);

echo "<p class = sa >By saMyogAntasya lopaH (8.2.23) :</p>";

echo "<p class = sa >संयोगान्तस्य लोपः (८.२.२३) :</p>";

display(0);

}

elseif ( sub($hl,array("+"),$hl,0) && arr($text,'/['.pc('hl').'][+]['.pc('hl').']$/') && $ratsasya===0 && $vriddhireci===0 )

{

$text = three($hl,array("+"),$hl,$hl,array("+"),blank(count($hl)),0);

echo "<p class = sa >By saMyogAntasya lopaH (8.2.23) :</p>";

echo "<p class = sa >संयोगान्तस्य लोपः (८.२.२३) :</p>";

display(0);

}

elseif ( (sub(array("M"),array("s"),array("+"),0) && $pada==="pada" && $vriddhireci===0) ) // patch for mAMsa. mAMspacanyA UkhAyAH is pending. ayasmayAdIni etc pending.

{

$text = three(array("M"),array("s+"),$ku,array("N+"),blank(count($hl)),$ku,0);

$text = three(array("M"),array("s+"),$cu,array("Y+"),blank(count($hl)),$cu,0);

$text = three(array("M"),array("s+"),$Tu,array("R+"),blank(count($hl)),$Tu,0);

$text = three(array("M"),array("s+"),$tu,array("n+"),blank(count($hl)),$tu,0);

$text = three(array("M"),array("s+"),$pu,array("m+"),blank(count($hl)),$pu,0);

$text = three(array("M"),array("s"),array("+"),array("m"),blank(count($hl)),array("+"),0);

echo "<p class = sa >By saMyogAntasya lopaH (8.2.23) and nimittApAye naimittikasyApyapAyaH (pa) :</p>";

echo "<p class = sa >संयोगान्तस्य लोपः (८.२.२३) तथा निमित्तापाये नैमित्तिकस्याप्यपायः (प) :</p>";

display(0);

}

elseif ( (sub(array("M"),$hl,array("+"),0) && $pada==="pada" && $vriddhireci===0) )

{

$text = three(array("M"),$hl,array("+"),array("M"),blank(count($hl)),array("+"),0);

echo "<p class = sa >By saMyogAntasya lopaH (8.2.23) :</p>";

echo "<p class = sa >संयोगान्तस्य लोपः (८.२.२३) :</p>";

display(0);

if (sub(array("M"),array("+"),blank(0),0))

{

$text = three(array("M"),array("+"),$ku,array("N"),array("+"),$ku,0);

$text = three(array("M"),array("+"),$cu,array("Y"),array("+"),$cu,0);

$text = three(array("M"),array("+"),$Tu,array("R"),array("+"),$Tu,0);

$text = three(array("M"),array("+"),$tu,array("n"),array("+"),$tu,0);

$text = three(array("M"),array("+"),$pu,array("m"),array("+"),$pu,0);

$text = two(array("M"),array("+"),array("m"),array("+"),0);

echo "<p class = sa >By nimittApAye naimittikasyApyapAyaH (pa) :</p>";

echo "<p class = sa >निमित्तापाये नैमित्तिकस्याप्यपायः (प) :</p>";

display(0);

}

}

### /\* coH kuH (8.2.30) \*/

$cu = array("c","C","j","J","Y"); // array of cavarga.

$ku = array("k","K","g","G","N"); // array of kavarga.

$noco = array("ac","ic","ec","Ec"); // array where coH kuH won't apply.

if ((arr($text,'/['.flat($cu).'][+]['.pc('Jl').']/')) && !in\_array($fo,$noco)&& ( $kvinku===0 || $ancu===1) )

{

$text = three($cu,array("+"),prat('Jl'),$ku,array("+"),prat('Jl'),0);

echo "<p class = sa >By coH kuH (8.2.30) :</p>";

echo "<p class = sa >चोः कुः (८.२.३०) :</p>";

display(0); $coku=1; // 0 - doesn't prevent kvinpratyayasya kuH. 1 - prevents kvinpratyayasya kuH.

}

elseif (!in\_array($so,$noco) && arr($text,'/['.flat($cu).'][+]$/') && $kvinku===0)

{

$text = two($cu,array("+"),$ku,array("+"),0);

echo "<p class = sa >By coH kuH (8.2.30) :</p>";

echo "<p class = sa >चोः कुः (८.२.३०) :</p>";

display(0); $coku=1;

}

else

{

$coku=0;

}

### /\* vA druhamuhaSNuhaSNihAm (8.2.34) \*/

$druh = array("druh","muh","snuh","snih");

if (sub($druh,blank(0),blank(0),0) && (arr($text,'/[+]$/') || arr($text,'/[+]['.pc("Jl").']/')) )

{

$text = one(array("druh","muh","snuh","snih"),array("druG","muG","snuG","sniG"),1);

echo "<p class = sa >By vA druhamuhaSNuhaSNihAm (8.2.34) :</p>";

echo "<p class = sa >वा द्रुहमुहष्णुहष्णिहाम्‌ (८.२.३४) :</p>";

display(0);

}

### /\* dAderdhAtorghaH (8.2.33) \*/

$dade = array("dah","dAh","dih","duh","dfh","drAh",);

if (sub($dade,blank(0),blank(0),0) && (arr($text,'/[+]$/') || arr($text,'/[+]['.pc("Jl").']/')) )

{

$text = one(array("dah","dAh","dih","duh","dfh","drAh","druh"),array("daG","dAG","diG","duG","dfG","drAG","druG"),0);

echo "<p class = sa >By dAderdhAtorghaH (8.2.33) :</p>";

echo "<p class = sa >दादेर्धातोर्घः (८.२.३३) :</p>";

display(0);

$hodha1 = 1; // 0 - doesn't prevent ho DhaH. 1 - prevents ho DhaH.

} else { $hodha1 = 0; }

### /\* naho dhaH (8.2.35) \*/

if (sub(array("nah"),blank(0),blank(0),2) && (arr($text,'/[+]$/') || arr($text,'/[+]['.pc("Jl").']/')) )

{

$text = one(array("nah",),array("naD"),0);

echo "<p class = sa >By naho dhaH (8.2.35) :</p>";

echo "<p class = sa >नहो धः (८.२.३५) :</p>";

display(0);

$hodha2 = 1; // 0 - doesn't prevent ho DhaH. 1 - prevents ho DhaH.

} else { $hodha2 = 0; }

### /\* AhasthaH (8.2.36) \*/

if (in\_array($first,array("Ah")) && (arr($text,'/[+]['.pc("Jl").']/')) )

{

$text = one(array("Ah",),array("AT"),0);

echo "<p class = sa >By AhasthaH (8.2.36) :</p>";

echo "<p class = sa >आहस्थः (८.२.३६) :</p>";

display(0);

$hodha3=1; // 0 - doesn't prevent ho DhaH. 1 - prevents ho DhaH.

} else { $hodha3 = 0; }

### /\* ho DhaH (8.2.32) \*/

if (arr($text,'/[h][+]/') && sub(array("h"),prat("Jl"),blank(0),0) && $hodha1===0 && $hodha2 === 0 && $hodha3 === 0 && $kvinku===0)

{

$text = two(array("h"),prat('Jl'),array("Q"),prat('Jl'),0);

echo "<p class = sa >ho DhaH (8.2.32) :</p>";

echo "<p class = sa >हो ढः (८.२.३२) :</p>";

display(0);

}

if (arr($text,'/[h][+]$/') && $pada ==="pada" && $hodha1===0 && $hodha2 === 0 && $hodha3 === 0 && $kvinku===0)

{

$text = two(array("h"),array("+"),array("Q"),array("+"),0);

echo "<p class = sa >ho DhaH (8.2.32) :</p>";

echo "<p class = sa >हो ढः (८.२.३२) :</p>";

display(0);

}

if (arr($text,'/[h]$/') && $hodha1===0 && $hodha2 === 0 && $hodha3 === 0 && $kvinku===0)

{

$text = last(array("h"),array("Q"),0);

echo "<p class = sa >ho DhaH (8.2.32) :</p>";

echo "<p class = sa >हो ढः (८.२.३२) :</p>";

display(0);

}

### /\* ekAco bazo bhaS jhaSantasya sdhvoH (8.2.37) \*/ // Not good code. Think hard.

$ekaco = array("gaD","gaB","gaQ","gAQ","gAD","gAQ","guD","guQ","gUQ","gfD","gfQ","graB","graQ","griQ","glaQ","qaQ","qiQ","quQ","daG","daG","daG","daG","diG","duG","duG","dfG","dfG","dfG","drAG","drAG","druG","druh","baD","baQ","bAQ","bAD","bAQ","bIB","buD","bfQ","beQ","braQ","druQ","gardaB"); // list of verbs satisfying the conditions.

$ekaco1 = array("GaD","GaB","GaQ","GAQ","GAD","GAQ","GuD","GuQ","GUQ","GfD","GfQ","GraB","GraQ","GriQ","GlaQ","QaQ","QiQ","QuQ","DaG","DaG","DaG","DaG","DiG","DuG","DuG","DfG","DfG","DfG","DrAG","DrAG","DruG","DruQ","BaD","BaQ","BAQ","BAD","BAQ","BIB","BuD","BfQ","BeQ","BraQ","DruQ","garDaB"); // list of verbs after application of this sUtra.

if (sub($ekaco,array("+"),blank(0),0) && ( arr($text,'/[+][sd]/') || arr($text,'/[+]$/') || $pada==="pada"))

{

$text = one($ekaco,$ekaco1,0);

echo "<p class = sa >By ekAco bazo bhaS jhaSantasya sdhvoH (8.2.37) :</p>";

echo "<p class = sa >एकाचो बशो भष्‌ झषन्तस्य स्ध्वोः (८.२.३७):</p>";

display(0);

}

### /\* jhalAM jazo'nte (8.2.39) \*/

if (arr($text,'/['.pc('Jl').']$/') )

{

if ($r2 ===1)

{

echo "<p class = sa >jhalAM jazo'nte is barred by sasajuSo ruH for second word. <hr>"; echo "<p class = sa >द्वितीय पद के लिए ससजुषो रुः से झलां जशोऽन्ते बाधित हुआ है । <hr>";

}

else

{

$text = last(prat('Jl'),savarna(prat('Jl'),prat('jS')),0);

}

echo "<p class = sa >By jhalAM jazo'nte (8.2.39), The padAnta is 'jhal' is replaced by 'jaz' :</p>";

echo "<p class = sa >झलां जशोऽन्ते (८.२.३९) :</p>";

display(0);

}

if (arr($text,'/['.pc('Jl').'][+]/') && ( $pada === "pada" && !arr(array($fo),'/[s]$/')) )

{

if ($r1 === 1 )

{

echo "<p class = sa >jhalAM jazo'nte is barred by sasajuSo ruH for first word. <hr>"; echo "<p class = sa >प्रथम पद के लिए ससजुषो रुः से झलां जशोऽन्ते बाधित हुआ है । <hr>";

}

else

{

$text = two(prat('Jl'),array("+"),savarna(prat('Jl'),prat('jS')),array("+"),0);

echo "<p class = sa >By jhalAM jazo'nte (8.2.39), The padAnta is 'jhal' is replaced by 'jaz' :</p>";

echo "<p class = sa >झलां जशोऽन्ते (८.२.३९) :</p>";

display(0);

}

}

### /\* kvinpratyaya patch for asRj \*/

if ($fo==="asfj" && in\_array($so,array("su!","am")) && $gender==="n")

{

$text=two(array("z"),array("+"),array("K"),array("+"),0);

echo "<p class = sa >By kvinpratyayasya kuH (8.2.62) :</p>";

echo "<p class = sa >क्विन्प्रत्ययस्य कुः (८.२.६२) :</p>";

display(0);

$text=two(array("K"),array("+"),array("g"),array("+"),0);

echo "<p class = sa >By jhalAM jazo'nte (8.2.39) :</p>";

echo "<p class = sa >झलां जशोऽन्ते (८.२.३९) :</p>";

display(0);

$asRj=1; // 0 - doesn't prevent reapplication of kvinpratyayasya kuH. 1 - prevents reapplication of kvinpratyayasya kuH.

} else {$asRj=0; }

### /\* kvinpratyayasya kuH (8.2.62) \*/

if (sub($hl,array("+"),blank(0),0) && ( $kvin===1 || ($kvip===1 && $fo==="diS") ) && $pada==="pada" && !sub(array("S","z","s"),array("+"),blank(0),0) && ( $kvinku===1 || $Asarva===0 ) && ($coku!==1||$goanc===1) && $asRj===0)

{

$text = two($cu,array("+"),$ku,array("+"),0);

$text = two($Tu,array("+"),$ku,array("+"),0);

$text = two($tu,array("+"),$ku,array("+"),0);

$text = two($pu,array("+"),$ku,array("+"),0);

$text = two(array("h","z"),array("+"),array("G","K"),array("+"),0);

//$text = two(array("S","s","y","r","v","l"),array("+"),array("S","s","y","r","v","l"),array("+"),0); // Pending to find kutva of S,z,s.

echo "<p class = sa >By kvinpratyayasya kuH (8.2.62) :</p>";

if ($kvip===1)

{

echo "<p class = hn >As 'tyadAdiSu dRSo'nAlocane kaYca (3.2.60) mandates kvin for dRz, others also take kutvam. </p>"; // special message fror kvip.

}

echo "<p class = sa >क्विन्प्रत्ययस्य कुः (८.२.६२) :</p>";

if ($kvip===1)

{

echo "<p class = hn >त्यदादिषु दृशोऽनालोचने कञ्च (३.२.६०) इति दृशेः क्विन्विधानादन्यत्रापि कुत्वम्‌ ।</p>";

}

display(0);

if(sub(array("G"),array("+"),blank(0),0))

{

$text = two(array("G"),array("+"),array("g"),array("+"),0);

echo "<p class = sa >By jhalAM jazo'nte (8.2.39), The padAnta is 'jhal' is replaced by 'jaz' :</p>";

echo "<p class = sa >झलां जशोऽन्ते (८.२.३९) :</p>";

display(0);

}

}

### /\* vizvasya vasurAToH (6.3.128) \*/

if (sub(array("viSva"),array("vasu","rAq"),blank(0),0))

{

$text = two(array("viSva"),array("vasu","rAq"),array("viSvA"),array("vasu","rAq"),0);

echo "<p class = sa >By vizvasya vasurAToH (6.3.128) :</p>";

echo "<p class = sa >विश्वस्य वसुराटोः (६.३.१२८) :</p>";

display (0);

}

### /\* bhobhagoaghoapUrvasya yo'zi (8.3.17) : \*/

$ash = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O","h","y","v","r","l","Y","m","N","R","n","J","B","G","Q","D","j","b","g","q","d");

if (sub(array("Bo","Bago","aGo","a","A"),array("r@"),$ash,0))

{

$text = three(array("Bo","Bago","aGo","a","A"),array("r@"),$ash,array("Bo","Bago","aGo","a","A"),array("y"),$ash,0);

echo "<p class = sa >By bhobhagoaghoapUrvasya yo'zi (8.3.17):</p>";

echo "<p class = sa >भोभगोअघोअपूर्वस्य योऽशि (८.३.१७) :</p>";

$bho = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

display (0);

} else { $bho =0; }

// Patch to convert the rutva before vowels and hash to repha.

if (arr($text,'/[r][@]/'))

{

echo "<p class = pa >By upadeze'janunAsika it (1.3.2) :</p>";

echo "<p class = pa >उपदेशेऽजनुनासिक इत्‌ (१.३.२) :</p>";

display(0);

$text = two(array("r@"),$ac,array("r"),$ac,0);

$text = two(array("r@"),prat('hS'),array("r"),prat('hS'),0);

$text = two(array("r@"),array("+"),array("r"),array("+"),0);

if (arr($text,'/[r][@]$/'))

{

$text = last(array("@"),array(""),0);

}

echo "<p class = sa >By tasya lopaH (1.3.9) :</p>";

echo "<p class = sa >तस्य लोपः (१.३.९) :</p>";

display(0);

}

### /\* SaDhoH kassi (8.2.41) \*/

if (sub(array("z","Q"),array("s"),blank(0),0))

{

$text = two(array("z","Q"),array("s"),array("k","k"),array("s"),0);

echo "<p class = sa >By SaDhoH kassi (8.2.41) :</p>";

echo "<p class = sa >षढोः कस्सि (८.२.४१) :</p>";

display(0);

}

### /\* mo no dhAtoH (8.2.64) \*/

if (arr($text,'/[m][+]/') && $dhatu===1 && $pada==="pada")

{

$text = one(array("m+",),array("n+",),0);

echo " <p class = sa >By mo no dhAtoH (8.2.64) :</p>";

echo " <p class = sa >मो नो धातोः (८.२.६४) :</p>";

display(0);

}

### /\* rvorupadhAyA dIrgha ikaH (8.2.76) \*/

//if ($dhatu===1 && ((sub(array("i","I","u","U","f","F","x","X",),array("r+","v+"),$hl,0) && $pada==="pada" ) || arr($text,'/[iIuUfFxX][rv]$/') || sub(array("i","I","u","U","f","F","x","X",),array("r","v"),array("+"),0)) && $pada==="pada") // This gives wrong application for Bis pratyaya.

if ($dhatu===1 && ((sub(array("i","I","u","U","f","F","x","X",),array("r+","v+"),$hl,0) && $pada==="pada" ) || sub(array("i","I","u","U","f","F","x","X",),array("r","v"),array("+"),0)) && $pada==="pada")

{

$text = three(array("i","I","u","U","f","F","x","X",),array("r","v"),array("+"),array("I","I","U","U","F","F","F","F",),array("r","v"),array("+"),0);

echo "<p class = sa >By rvorupadhAyA dIrgha ikaH (8.2.76) :</p>";

echo "<p class = sa >र्वोरुपधाया दीर्घ इकः (८.२.७६) :</p>";

display(0);

}

### /\* hali ca (8.2.77) \*/

$rvverb=array("ir","Ir","Ir","ur","kIv","kur","kfv","klIv","kziv","kzIv","kzIv","kzur","kzur","Kur","Kur","gir","gur","gUr","gUr","Gur","GUr","cir","cIv","cIv","cur","cUr","Cur","jIv","jur","jUr","tir","1tIv","tIv","3tIv","tur","tUr","div","div","div","dIv","dIv","DIv","Dur","Dur","Druv","niv","nIv","pIv","pur","pUr","Bur","mIv","mIv","mur","mur","mUr","mUr","rIv","SUr","Sriv","SrIv","zWiv","zWIv","siv","siv","sIv","sur","sUr","sTiv","sPur","sriv","srIv","hur",);

// pending to code for how to apply this to whole $rvverb. Right now just coded for pratidivan

if (sub(array("divn"),blank(0),blank(0),1))

{

$dhatu=1;

}

if ($dhatu===1 && sub($ik,array("r","v"),$hl,0) && (arr($text,'/[rv][+]/')|| arr($text,'/[v][n][+]/') ))

{

$text = three(array("i","I","u","U","f","F","x","X",),array("r","v"),$hl,array("I","I","U","U","F","F","F","F",),array("r","v"),$hl,0);

echo "<p class = sa >By hali ca (8.2.77) :</p>";

echo "<p class = sa >हलि च (८.२.७७) :</p>";

if ($allopo===1)

{

echo "<p class = hn >allopa doesn't have sthAnivadbhAva, because sthAnivadbhAva is barred in dIrghavidhi.</p>";

echo "<p class = hn >स्थानिवद्भाव का दीर्घविधि में निषेध होने से यहाँ अल्लोप का स्थानिवद्भाव नहीं है ।</p>";

}

display(0);

}

### /\* eta Idbahuvacane (8.2.81) \*/

if (sub(array("ad+e","ade+Bir","ade+Byar","ade+sAm","ade+su"),blank(0),blank(0),0) && in\_array($so,$bahusup) && $fo==="adas")

{

$text=one(array("ad+e","ade+Bir","ade+Byar","ade+sAm","ade+su"),array("amI","amI+Bir","amI+Byar","amI+sAm","amI+su"),0);

echo "<p class = sa >eta Idbahuvacane (8.2.81) :</p>";

echo "<p class = sa >एत ईद्बहुवचने (८.२.८१) :</p>";

display(0);

}

### /\* adaso'serdAdu do maH (8.2.80) \*/

// For proper adas forms.

$acmu = array("u","U","u","U","u","U","u","U","u","U","U","U","U","U",);

if (sub(array("ad"),blank(0),blank(0),0) && in\_array($fo,array("adas","adakas")) && !arr($text,'/[a][r][+]/') )

{

$text=two(array("ad"),$ac,array("am"),$acmu,0);

echo "<p class = sa >adaso'serdAdu do maH (8.2.80) :</p>";

echo "<p class = sa >अदसोऽसेर्दादु दो मः (८.२.८०) :</p>";

display(0);

}

// Not coded properly. coded only for adas-ancu combination.

if (sub(array("adadr"),blank(0),blank(0),0) )

{

$text=one(array("adadr"),array("amumu"),1);

$text=one(array("amumu"),array("adamu"),1);

echo "<p class = sa >adaso'serdAdu do maH (8.2.80) :</p>";

echo "<p class = hn >adaso'dreH pRthaGmutvaM kecidicCanti latvavat | kecidantyasadezasya netyeke'serhi dRSyate || </p>";

echo "<p class = sa >अदसोऽसेर्दादु दो मः (८.२.८०) :</p>";

echo "<p class = hn >अदसोऽद्रेः पृथङ्मुत्वं केचिदिच्छन्ति लत्ववत्‌ । केचिदन्त्यसदेशस्य नेत्येकेऽसेर्हि दृश्यते ॥</p>";

display(0);

}

### /\* vyorlaghuprayatnataraH zAkaTAyanasya (8.3.18) \*/

// This is regarding pronounciation. So no difference in word declention.

### /\* roH supi (8.3.16) \*/

if (arr($text,'/[r][+][s][u]$/') && $so==="sup" && !in\_array(1,$R))

{

echo "<p class = pa >roH supi (8.3.16) prevents application of kharavasAnayorvisarjanIyaH. </p>";

echo "<p class = pa >रोः सुपि (८.३.१६) से खरवसानयोर्विसर्जनीयः का प्रतिषेध होता है ।</p>";

display(0);

$roHsupi=1; // 0 - doesn't prevent kharavasAnayorvisarjanIyaH. 1 - prevents kharavasAnayorvisarjanIyaH.

} else { $roHsupi=0; }

### /\* kharavasAnayorvisarjanIyaH (8.3.15) \*/

if (arr($text,'/[+]['.pc('Kr').']/') && $roHsupi===0 && sub(array("r","r@"),array("+"),prat('Kr'),0) && $pada === "pada")

{

$text = two(array("r@","r"),prat("Kr"),array("H","H"),prat("Kr"),0);

echo "<p class = sa >By kharavasAnayorvisarjanIyaH (8.3.15) :</p>";

echo "<p class = sa >खरवसानयोर्विसर्जनीयः (८.३.१५) :</p>";

display(0);

}

if ( arr($text,'/[@r]$/')||arr($text,'/[r][+]$/') && $roHsupi===0)

{

$text = last(array("r@","r"),array("H","H"),0);

if (arr($text,'/[r][+]$/'))

{

$text = one(array("+"),array(""),0);

}

$text = last(array("r"),array("H"),0);

echo "<p class = sa >By kharavasAnayorvisarjanIyaH (8.3.15) :</p>";

echo "<p class = sa >खरवसानयोर्विसर्जनीयः (८.३.१५) :</p>";

display(0);

}

### /\* Dho Dhe lopaH (8.3.13) \*/

if (sub(array("Q"),array("Q"),blank(0),0))

{

$text = two(array('Q'),array('Q'),array(''),array('#Q'),0);

echo "<p class = sa >By Dho Dhe lopaH (8.3.13) :</p>";

echo "<p class = sa >ढो ढे लोपः (८.३.१३) :</p>";

display(0);

$dho = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $dho = 0; }

### /\* ro ri (8.3.14) \*/

if (sub(array("r"),array("r"),blank(0),0))

{

$text = two(array('r'),array('r'),array(''),array('#r'),0);

echo "<p class = sa >By ro ri (8.3.14) :</p>";

echo "<p class = sa >रो रि (८.३.१४) :</p>";

display(0);

$ro = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $ro = 0; }

### /\* Dhralope pUrvasya dIrgho'NaH (6.3.111) \*/

$ana = array("a","A","i","I","u","U","f","F","x","X");

$anna = array("A","A","I","I","U","U","F","F","X","X");

if (($ro ===1 || $dho===1) && sub($ana,array('#r',"#Q"),blank(0),0))

{

$text = two($ana,array('#r','#Q'),$anna,array(' r',' Q'),0);

echo "<p class = sa >By Dhralope pUrvasya dIrgho'NaH (6.3.111) :</p>";

echo "<p class = sa >ढ्रलोपे पूर्वस्य दीर्घोऽणः (६.३.१११) :</p>";

display(0);

}

### /\* lopaH zAkalyasya (8.3.19) and vyorlaghuprayatnataraH zAkaTAyanasya (8.3.18) \*/

$aa = array("a","A");$yv = array("y+","v+"); $space=array(" "," "); // creating some arrays.

if (sub($aa,$yv,$ac,0) && (arr($text,'/['.pc('ec').'][+]['.flat($ac).']/') || $bho === 1) && $pada === "pada")

{

echo "<p class = sa >By lopaH zAkalyasya (8.3.19) and vyorlaghuprayatnataraH zAkaTAyanasya (8.3.18) :</p>";

echo "<p class = sa >लोपः शाकल्यस्य (८.३.१९) तथा व्योर्लघुप्रयत्नः शाकटायनस्य (८.३.१८) :</p>";

$text = three($aa,$yv,$ac,$aa,array(" +"," +"),$ac,1);

display(0);

}

### /\* hali sarveSAm (8.3.22) \*/

if ($bho === 1 && sub(array("y+"),$hl,blank(0),0) && $pada==="pratyaya")

{

$text = three(array("Bo","Bago","aGo","A"),array("y+"),$hl,array("Bo","Bago","aGo","A"),array("+"),$hl,0);

echo "<p class = sa >By hali sarveSAm (8.3.22) :</p>";

echo "<p class = sa >हलि सर्वेषाम्‌ (८.३.२२) :</p>";

display(0);

}

if ($bho === 1 && sub(array("y+"),$hl,blank(0),0) && (in\_array($so,$sup) && $pada==="pada"))

{

$text = three(array("Bo","Bago","aGo","A"),array("y+"),$hl,array("Bo","Bago","aGo","A"),array("+"),$hl,0);

echo "<p class = sa >By hali sarveSAm (8.3.22) :</p>";

echo "<p class = sa >हलि सर्वेषाम्‌ (८.३.२२) :</p>";

display(0);

}

### /\* oto gArgyasya (8.3.20) \*/

if (arr($text,'/[o][y][+]/') && $bho ===1 && $pada === "pada")

{

$text = one(array("oy+"),array("o +"),0);

echo "<p class = sa >By oto gArgyasya (8.3.20) :</p>

<p class = hn >N.B. This rule applies only to the padAnta alaghuprayatna yakAra following 'o' only.</p>";

echo "<p class = sa >ओतो गार्ग्यस्य (८.३.२०) :</p>

<p class = hn >यह ओकार के परे आए हुए अलघुप्रयत्न पदान्त यकार को ही लागू होता है ।</p>";

display(0);

}

### /\* uJi ca pade (8.3.21) \*/

if ((sub(array("ay","av"),array("u "),blank(0),0)|| (sub(array("ay","av"),blank(0),blank(0),0) && $second === "u")) && $bho ===1 && $pada === "pada")

{

$text = two(array("ay","av"),array("u"),array("a","a"),array("u"),0);

echo "<p class = sa >By uJi ca pade (8.3.21) :</p>";

echo "<p class = sa >उञि च पदे (८.३.२१) :</p>";

display(0);

}

### /\* mo rAji samaH kvau (8.3.25) \*/

if (sub(array("sam"),array("rA"),blank(0),0))

{

$text = two(array("sam"),array("rA"),array("sam"),array("rA"),0);

echo "<p class = sa >By mo rAji samaH kvau (8.3.25) :</p>";

echo "<p class = sa >मो राजि समः क्वौ (८.३.२५) :</p>";

display(0);

$mo = 1; // 0 - doesn't prevent application of mo'nusvAraH. 1 - prevents application of mo'nusvAraH.

} else { $mo = 0; }

### /\* mo'nusvAraH (8.3.23) \*/

if (arr($text,'/[m][+]['.pc('hl').']/') && $pada ==="pada" && $mo === 0)

{

$text = two(array('m'),prat('hl'),array('M'),prat('hl'),0);

echo "<p class = sa >By mo'nusvAraH (8.3.23) :</p>

<p class = hn >N.B.: The conversion to anusvAra occurs only if the m is at the end of a pada. Otherwise this conversion doesn't apply. Ignore all consequentiality in that case.</p>";

echo "<p class = sa >मोऽनुस्वारः (८.३.२३) :</p>

<p class = hn >यदि मकार पदान्त में है तभी अनुस्वार में बदलता है । अन्यथा नहीं ।</p>";

display(0);

}

### /\* nazcApadAntasya jhali (8.3.24) \*/

if (arr($text,'/[mn][+]['.pc('Jl').']/') && $pada === "pratyaya" )

{

$text = two(array('m','n'),prat('Jl'),array('M','M'),prat('Jl'),0);

echo "<p class = sa >By nazcApadAntasya jhali (8.3.24) :</p>

<p class = hn >If n/m is inside a pada, it should be converted to anusvAra. So ignore the case which doesn't apply here.</p>";

echo "<p class = sa >नश्चापदान्तस्य झलि (८.३.२४) :</p>

<p class = hn >यदि नकार या मकार पदान्त में नहीं है तब भी यह नियम से अनुस्वार होता है ।</p>";

display(0);

}

if(arr($text,'/[mn]['.pc('Jl').']/') )

{

$text = two(array('m','n'),prat('Jl'),array('M','M'),prat('Jl'),2);

echo "<p class = sa >By nazcApadAntasya jhali (8.3.24) :</p>

<p class = hn >If n/m is inside a pada, it should be converted to anusvAra. So ignore the case which doesn't apply here.</p>";

echo "<p class = sa >नश्चापदान्तस्य झलि (८.३.२४) :</p>

<p class = hn >यदि नकार या मकार पदान्त में नहीं है तब भी यह नियम से अनुस्वार होता है ।</p>";

display(0);

}

### /\* he mapare vA (8.3.26) and yavalapare yavalA veti vaktavyam (vA 4902) \*/

if (sub(array("M"),array("hm","hy","hv","hl"),blank(0),0))

{

$text = two(array("M"),array("hy",),array("!y",),array("hy",),1);

$text = two(array("M"),array("hm",),array("m",),array("hy",),1);

$text = two(array("M"),array("hv",),array("!v",),array("hv",),1);

$text = two(array("M"),array("hl",),array("!l",),array("hl",),1);

echo "<p class = sa >By he mapare vA (8.3.26) and yavalapare yavalA veti vaktavyam (vA 4902) :</p>";

echo "<p class = sa >हे मपरे वा (८.३.२६) तथा यवलपरे यवला वेति वक्तव्यम्‌ (वा ४९०२) :</p>";

display(0);

}

### /\* napare naH (8.3.27) \*/

if (sub(array("M"),array("hn"),blank(0),0))

{

$text = two(array("M"),array("hn"),array("n"),array("hn",),1);

echo "<p class = sa >By napare naH (8.3.27) :</p>";

echo "<p class = sa >नपरे नः (८.३.२७) :</p>";

display(0);

}

### /\* GNoH kukTukzari (8.3.28) \*/

if (sub(array("N","R"),prat('Sr'),blank(0),0))

{

$text = two(array("N","R"),prat('Sr'),array("Nk","Rw"),prat('Sr'),1);

echo "<p class = sa >By GNoH kukTukzari (8.3.28) :</p>";

echo "<p class = sa >ङ्णोः कुक्टुक्शरि (८.३.२८) :</p>";

display(0);

}

### /\* DaH si dhuT (8.3.29) \*/

$dhut = 0; // 0 - no dhuT Agama. 1 - dhuT Agama.

if (sub(array("q"),array("s"),blank(0),0))

{

$text = two(array("q"),array("s"),array("q"),array("Ds"),1);

echo "<p class = sa >By DaH si dhuT (8.3.29) :</p>";

echo "<p class = sa >डः सि धुट्‌ (८.३.२९) :</p>";

display(0);

}

### /\* nazca (8.3.30) \*/

if (sub(array("n"),array("s"),blank(0),0))

{

$text = two(array("n"),array("s"),array("n"),array("Ds"),1);

echo "<p class = sa >By nazca (8.3.30) :</p>";

echo "<p class = sa >नश्च (८.३.३०) :</p>";

display(0);

}

### /\* zi tuk (8.3.31) \*/

if (arr($text,'/[n][+][S]/') && $pada === "pada")

{

$text = one(array("n+S"),array("nt+S"),1);

echo "<p class = sa >By zi tuk (8.3.31) :</p>";

echo "<p class = sa >शि तुक्‌ (८.३.३१) :</p>";

display(0);

}

### /\* Gamo hrasvAdaci GamuNnityam (8.3.32) \*/

// Here the Agama has to be affiliated to $ac. Patch is bad.

$nogamo = array("aR","ak","ik","uk","ac","ic","ec","aw","aR","iR","am","aS","al",); // array where this rule won't apply.

if (arr($text,'/['.flat($hrasva).'][NRn][+]['.flat($ac).']/') && $pada === "pada" && !in\_array($second,$nogamo) && !sub(array("pataYjal","sImant"),blank(0),blank(0),0))

{

$text = three($hrasva,array("N","R","n"),$ac,$hrasva,array("NN","RR","nn"),$ac,0);

echo "<p class = sa >By Gamo hrasvAdaci GamuNnityam (8.3.32) :</p>";

echo "<p class = sa >ङमो ह्रस्वादचि ङमुण्नित्यम्‌ (८.३.३२) :</p>";

display(0);

}

### /\* saheH sADaH saH (8.3.56) \*/

if (sub(array("sAq"),array("+"),blank(0),0))

{

$text = two(array("sAq"),array("+"),array("zAq"),array("+"),0);

echo "<p class = sa >By saheH sADaH saH (8.3.56) :";

echo "<p class = sa >सहेः साडः सः (८.३.५६) :";

display(0);

}

### /\* sampuGkAnAM so vaktavyaH (vA 4892) \*/

if (sub(array("saM","sa!","puM","pu!","kAM","kA!"),array("H"),blank(0),0))

{

$text = two(array("saM","sa!","puM","pu!","kAM","kA!"),array("H"),array("saM","sa!","puM","pu!","kAM","kA!"),array("s"),0);

echo "<p class = sa >By sampuGkAnAM so vaktavyaH (vA 4892) :";

echo "<p class = sa >सम्पुङ्कानां सो वक्तव्यः (वा ४८९२) :";

display(0);

}

### /\* samo vA lopameke (bhASya) \*/

if (sub(array("saMs","sa!s"),array("s"),$hl,0))

{

$text = two(array("saMs","sa!s"),array("s"),array("saM","sa!"),array("s"),1);

echo "<p class = sa >By samo vA lopameke (bhASya) :</p>";

echo "<p class = sa >समो वा लोपमेके (भाष्य) :</p>";

display(0);

}

### /\* dvistrizcaturiti kRtvo'rthe (8.3.43) \*/

if (sub(array("dviH","triH","catuH"),$ku,blank(0),0))

{

$text = two (array("dviH","triH","catuH"),$ku,array("dviz","triz","catuz"),$ku,1);

echo "<p class = sa >By dvistrizcaturiti kRtvo'rthe (8.3.43) :</p>

<p class = hn >N.B. This applies only in case of kRtvo'rthe.</p>";

echo "<p class = sa >द्विस्त्रिश्चतुरिति कृत्वोऽर्थे (८.३.४३):</p>

<p class = hn >यह नियम सिर्फ कृत्वोऽर्थ में ही लागू होता है ।</p>";

display(0);

$dvi1 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $dvi1 = 0; }

if (sub(array("dviH","triH","catuH"),$pu,blank(0),0))

{

$text = two (array("dviH","triH","catuH"),$pu,array("dviz","triz","catuz"),$pu,1);

echo "<p class = sa >By dvistrizcaturiti kRtvo'rthe (8.3.43) :</p>

<p class = hn >N.B. This applies only in case of kRtvo'rthe.</p>";

echo "<p class = sa >द्विस्त्रिश्चतुरिति कृत्वोऽर्थे (८.३.४३):</p>

<p class = hn >यह नियम सिर्फ कृत्वोऽर्थ में ही लागू होता है ।</p>";

display(0);

$dvi2 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $dvi2 = 0; }

### /\* muhusaH pratiSedhaH (vA 4911) \*/

if (sub(array("muhu"),array("H"),$pu,0)||sub(array("muhu"),array("H"),$ku,0))

{

$text = three(array("muhu"),array("H"),$pu,array("muhu"),array("H"),$pu,0);

$text = three(array("muhu"),array("H"),$ku,array("muhu"),array("H"),$ku,0);

echo "<p class = sa >By muhusaH pratiSedhaH (vA 4911) :</p>";

echo "<p class = sa >मुहुसः प्रतिषेधः (वा ४९११) :</p>";

display(0);

$muhu1 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $muhu1 = 0; }

### /\* kaskAdiSu ca (8.3.48) \*/

$kaska = array("kaHkaH","kOtaHkut","sarpiHkuRqik","BrAtuHputr","SunaHkarR","sadyaHkAl","sadyaHkI","sAdyaHk","kAMHkAn","DanuHkapAl","bahiHpal","barhiHpal","yajuHpAtr","ayaHkAnt","tamaHkARq","ayaHkARq","medaHpiRq","BAHkar","ahaHkar","kaH+kaH","kOtaH+kut","sarpiH+kuRqik","BrAtuH+putr","SunaH+karR","sadyaH+kAl","sadyaH+kI","sAdyaH+k","kAMH+kAn","DanuH+kapAl","bahiH+pal","barhiH+pal","yajuH+pAtr","ayaH+kAnt","tamaH+kARq","ayaH+kARq","medaH+piRq","BAH+kar","ahaH+kar"); // kaskAdi gaNa, before joining.

$kaskareplace = array("kaskaH","kOtaskut","sarpizkuRqik","BrAtuzputr","SunaskarR","sadyaskAl","sadyaskI","sAdyask","kAMskAn","DanuzkapAl","bahizpal","barhizpal","yajuzpAtr","ayaskAnt","tamaskARq","ayaskARq","medaspiRq","BAskar","ahaskar","kas+kaH","kOtas+kut","sarpiz+kuRqik","BrAtuz+putr","Sunas+karR","sadyas+kAl","sadyas+kI","sAdyas+k","kAMs+kAn","Danuz+kapAl","bahiz+pal","barhiz+pal","yajuz+pAtr","ayas+kAnt","tamas+kARq","ayas+kARq","medas+piRq","BAs+kar","ahas+kar"); // kaskAdi gaNa, after joining.

if(sub($kaska,blank(0),blank(0),0))

{

$text = one ($kaska,$kaskareplace,0);

echo "<p class = sa >By kaskAdiSu ca (8.3.48) ";

echo "<p class = sa >कस्कादिषु च (८.३.४८) ";

display(0);

$kaska = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $kaska = 0; }

### /\* isusoH sAmarthye (8.3.44) and nityaM samAse'nuttarapadasthasya (8.3.45) \*/

if (sub(array("iH","uH",),$ku,blank(0),0) && $dvi1===0 && $dvi2===0 && $muhu1 ===0)

{

$text = two (array("iH","uH"),$ku,array("iz","uz"),$ku,1);

echo "<p class = sa >By isusoH sAmarthye (8.3.44) and nityaM samAse'nuttarapadasthasya (8.3.45) :</p>

<p class = hn >N.B. This applies only in case of sAmarthya. If 'is' or 'us' pratyayas are at the end of first component of a compound, they are mandatorily converted to 'S'.</p>";

echo "<p class = sa >इसुसोः सामर्थ्ये (८.३.४४) तथा नित्यं समासेऽनुत्तरपदस्थस्य (८.३.४५) :</p>

<p class = hn >यह तभी लागू होता है जब सामर्थ्य में प्रयोग हुआ हो । यदि 'इस्‌' और 'उस्‌' प्रत्यय उत्तरपद में न हों तब आवश्यक रूप से शकार में परिवर्तन होता है ।</p>";

display(0);

$isu1 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $isu1 = 0; }

if (sub(array("iH","uH"),$pu,blank(0),0))

{

$text = two (array("iH","uH"),$pu,array("iz","uz"),$pu,1);

echo "<p class = sa >By isusoH sAmarthye (8.3.44) and nityaM samAse'nuttarapadasthasya (8.3.45) :</p>

<p class = hn >N.B. This applies only in case of sAmarthya. If 'is' or 'us' pratyayas are at the end of first component of a compound, they are mandatorily converted to 'S'.</p>";

echo "<p class = sa >इसुसोः सामर्थ्ये (८.३.४४) तथा नित्यं समासेऽनुत्तरपदस्थस्य (८.३.४५) :</p>

<p class = hn >यह तभी लागू होता है जब सामर्थ्य में प्रयोग हुआ हो । यदि 'इस्‌' और 'उस्‌' प्रत्यय उत्तरपद में न हों तब आवश्यक रूप से षकार में परिवर्तन होता है ।</p>";

display(0);

$isu2 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $isu2= 0; }

### /\* idudupadhasya cApratyayasya (8.3.41) \*/

$id = array("i","u",);

if (sub($iN,array("H"),$ku,0) && $dvi1===0 && $dvi2===0 && $isu1 ===0 && $isu2 ===0&& $muhu1 ===0 && $pada !== "pratyaya")

{

$text = three(array("i","u",),array("H"),$ku,array("i","u",),array("z"),$ku,1);

echo "<p class = sa >By idudupadhasya cApratyayasya (8.3.41) :</p>

<p class = hn >N.B. : the visarga will be converted to 'S' only if it is not followed by pratyaya.</p>";

echo "<p class = sa >इदुदुपधस्य चाप्रत्ययस्य (८.३.४१) :</p>

<p class = hn >यदि परे प्रत्यय न हो तभी षकार में परिवर्तन होता है ।</p>";

display(0);

}

if (sub($iN,array("H"),$pu,0) && $pada !== "pratyaya")

{

$text = three(array("i","u",),array("H"),$pu,array("i","u",),array("z"),$pu,1);

echo "<p class = sa >By idudupadhasya cApratyayasya (8.3.41) :</p>

<p class = hn >N.B. : the visarga will be converted to 'S' only if it is not followed by pratyaya.</p>";

echo "<p class = sa >इदुदुपधस्य चाप्रत्ययस्य (८.३.४१) :</p>

<p class = hn >यदि परे प्रत्यय न हो तभी षकार में परिवर्तन होता है ।</p>";

display(0);

}

### /\* ekAdezazAstranimittikasya na Satvam | kaskAdiSu bhrAtuSputrazabdasya pAThAt (vA 4915) \*/

// Pending to code.

### /\* iNaH SaH (8.3.39) \*/

if (sub($iN,array("H"),array("pAS","kalp","kAmy","ka","kAMy"),blank(0),0) && $dvi1===0 && $dvi2===0 && $isu1 ===0 && $isu2 ===0 && $muhu1 ===0)

{

$text = three($iN,array("H"),array("pAS","kalp","kAmy","ka","kAMy"),$iN,array("z"),array("pAS","kalp","kAmy","ka","kAmy"),0);

echo "<p class = sa >By iNaH SaH (8.3.39) :</p>";

echo "<p class = sa >इणः षः (८.३.३९) :</p>";

display(0); $inah = 1;

} else { $inah = 0; }

### /\* namaspurasorgatyoH (8.3.40) \*/

if (sub(array("namaH","puraH"),$ku,blank(0),0))

{

$text = two(array("namaH","puraH"),$ku,array("namas","puras"),$ku,1);

echo "<p class = sa >By namaspurasorgatyoH (8.3.40) :</p>

<p class = hn >N.B. : The conversion to namas / puras is done only in case it has gati saJjJA.</p>";

echo "<p class = sa >नमस्पुरसोर्गत्योः (८.३.४०) :</p>

<p class = hn >यदि गति संज्ञा हो तभी नमस्‌ / पुरस्‌ में परिवर्तन होता है ।</p>";

display(0);

$nama1 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $nama1 = 0; }

if (sub(array("namaH","puraH"),$pu,blank(0),0) && $nama1 !==1)

{

$text = two(array("namaH","puraH"),$pu,array("namas","puras"),$pu,1);

echo "<p class = sa >By namaspurasorgatyoH (8.3.40) :</p>

<p class = hn >N.B. : The conversion to namas / puras is done only in case it has gati saJjJA.</p>";

echo "<p class = sa >नमस्पुरसोर्गत्योः (८.३.४०) :</p>

<p class = hn >यदि गति संज्ञा हो तभी नमस्‌ / पुरस्‌ में परिवर्तन होता है ।</p>";

display(0);

$nama2 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $nama2 = 0; }

### /\* tiraso'nyatarsyAm (8.3.42) \*/

if (sub(array("tiraH"),$ku,blank(0),0))

{

$text = two (array('tiraH'),$ku,array('tiras'),$ku,1);

echo "<p class = sa >By tiraso'nyatarasyAm (8.3.42) :</p>";

echo "<p class = sa >तिरसोऽन्यतरस्याम्‌ (८.३.४२) :</p>";

display(0);

$tir1 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $tir1 = 0; }

if (sub(array("tiraH"),$pu,blank(0),0))

{

$text = two (array('tiraH'),$pu,array('tiras'),$pu,1);

echo "<p class = sa >By tiraso'nyatarasyAm (8.3.42) :</p>";

echo "<p class = sa >तिरसोऽन्यतरस्याम्‌ (८.३.४२) :</p>";

display(0);

$tir2 = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else { $tir2 = 0; }

### /\* ataH kRkamikaMsakumbhapAtrakuzAkarNISvanavyayasya (8.3.46) \*/

$atah = 0; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

if (sub(array("aH"),array("kAr","kAm","kAMs","kumBa","pAtra","kuSA","karRI"),blank(0),0) && $nama1 !== 1 && $nama2 !== 1 && $tir1===0 && $tir2===0 )

{

$text = two(array("aH"),array("kAr","kAm","kAMs","kumBa","pAtra","kuSA","karRI"),array('as'),array("kAr","kAm","kAMs","kumBa","pAtra","kuSA","karRI"),1);

echo "<p class = sa >By ataH kRkamikaMsakumbhapAtrakuzAkarNISvanavyayasya (8.3.46) :</p>

<p class = hn > This applies olny when there is compound and the word with 'as' is neither uttarapadastha nor avyaya.</p>";

echo "<p class = sa >अतः कृकमिकंसकुम्भपात्रकुशाकर्णीष्वनव्ययस्य (८.३.४६) :</p>

<p class = hn >यह तब लागू होता है, जब समास में 'अस्‌' से अन्त होनेवाला शब्द न तो उत्तरपदस्त हो न ही अव्यय हो ।</p>";

display(0);

$atah = 1;

}

### /\* adhazzirasI pade \*/

if (sub(array("aDaH","SiraH"),array("pada"),blank(0),0) )

{

$text = two(array("aDaH","SiraH"),array("pada"),array("aDas","Siras"),array("pada"),1);

echo "<p class = sa >By aDazzirasI pade (8.3.47) :</p>

<p class = hn > This applies olny when there is compound and the word 'adhas' or 'ziras' is not uttarapadastha.</p>";

echo "<p class = sa >अधश्शिरसी पदे (८.३.४७) :</p>

<p class = hn >यह नियम तभी लागू होता है जब अधस्‌ / शिरस्‌ उत्तरपदस्थ नहीं होते ।</p>";

display(0); $atah = 1;

}

### /\* so'padAdau (8.3.38), pAzakalpakakAmyeSviti vAcyam (vA 5033), anavyayasyeti vAcyam (vA 4902) and kAmye roreveti vAcyam (vA 4902) \*/

// anavyayasyeti vAcyam (vA 4901) is pending to code.

if (sub(array("H"),array("pAS","kalp","kAmy","ka"),blank(0),0) && $inah !== 1 && $nama1 !== 1 && $nama2 !== 1 && $dvi1===0 && $dvi2===0 && $isu1 ===0 && $isu2 ===0 && $tir1===0 && $tir2===0 && $muhu1 ===0 && $atah ===0)

{

$text = two(array("H"),array("kalp","kAmy","ka","kAMy"),array('s'),array("kalp","kAmy","ka","kAMy"),0);

$text = two(array("H"),array("pAS"),array('s'),array("pAS"),0);

if (preg\_match('/[sr]$/',$first))

{

$text = one(array('skAmy','skAMy'),array('HkAmy','HkAMy'),1);

}

echo "<p class = sa >By so'padAdau (8.3.38), pAzakalpakakAmyeSviti vAcyam (vA 5033), anavyayasyeti vAcyam (vA 4902) and kAmye roreveti vAcyam (vA 4902) :</p>";

echo "<p class = sa >सोऽपदादौ (८.३.३८), पाशकल्पककाम्येष्विति वाच्यम्‌ (वा ५०३३), अनव्ययस्येति वाच्यम्‌ (वा ४९०२) तथा काम्ये रोरेवेति वाच्यम् (वा ४९०२) :</p>";

display(0);

}

### /\* zarpare visarjanIyaH (8.3.35) \*/

if (sub(array("H"),prat('Kr'),prat('Sr'),0) )

{

echo "<p class = sa >By zarpare visarjanIyaH (8.3.35) :</p>";

echo "<p class = sa >शर्परे विसर्जनीयः (८.३.३५) :</p>";

display(0);

$zarpare = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied. 2 - visarjanIyasya saH. 3 - vA zari.

} else { $zarpare = 0; }

### /\* kupvoH &k&pau ca (8.3.37) \*/

// <p class = hn >Note that we have used & as jihvAmUlIya and \* as upadhmAnIya.

if((sub(array("H"),$ku,blank(0),0)||sub(array("H"),$pu,blank(0),0)) && $kaska !== 1 && $zarpare ===0)

{

$text = two(array("H"),$ku,array("&"),$ku,1);

$text = two(array("H"),$pu,array("&"),$pu,1);

echo "<p class = sa >By kupvoH &k&pau ca (8.3.37). :</p>";

echo "<p class = sa >कुप्वोः ᳲकᳲपौ च (८.३.३७). :</p>";

display(0);

$kupvo = 1; // 0 - This sUtra has not applied. 1 - This sUtra has applied.

} else {$kupvo = 0; }

### /\* visarjanIyasya saH (8.3.34) \*/

if(sub(array("H"),prat('Kr'),blank(0),0) && $zarpare !==1 && $kupvo ===0)

{

$text = two(array("H"),prat('Kr'),array("s"),prat('Kr'),0);

echo "<p class = sa >By visarjanIyasya saH (8.3.34) :</p>";

echo "<p class = sa >विसर्जनीयस्य सः (८.३.३४) :</p>";

display(0);

$zarpare = 2;

}

### /\* vA zari (8.3.36) \*/

if(sub(array("s"),prat('Sr'),blank(0),0) && $zarpare === 2)

{

$text = two(array("s"),array("S","z","s"),array("H"),array("S","z","s"),1);

echo "<p class = sa >By vA zari (8.3.36) :</p>";

display(0);

$zarpare = 3;

}

### /\* kharpare zari vA visargalopo vaktavyaH (vA 4906) \*/

if(sub(array("H"),prat('Sr'),prat('Kr'),0) && $zarpare === 3)

{

$text = three(array("H"),prat('Sr'),prat('Kr'),array(""),prat('Sr'),prat('Kr'),1);

echo "<p class = sa >By kharpare zari vA visargalopo vaktavyaH (vA 4906) :</p>";

echo "<p class = sa >खर्परे शरि वा विसर्गलोपो वक्तव्यः (वा ४९०६) :</p>";

display(0);

}

### /\* apadAntasya mUrdhanyaH (8.3.55), iNkoH (8.3.57) and AdezapratyayayoH (8.3.59) \*/

// Not coded perfectly, only according to the need of vibhaktis.

$iN1 = array("i","I","u","U","f","F","x","X","e","o","E","O","h","y","v","r","l","k","K","g","G","N");

if((sub(array("i","I","u","U","f","F","x","X","e","o","E","O","h","y","v","r","l","k","K","g","G","N"),array("+s"),blank(0),0)) && (in\_array($so,array("Am","sup"))|| in\_array(1,$samp)) || ($fo==="adas" && in\_array($so,array("Ne","Nasi!","Nas","Am","Ni"))) )

{

$text = two($iN1,array("+s"),$iN1,array("+z"),0);

echo "<p class = sa >By apadAntasya mUrdhanyaH (8.3.55), iNkoH (8.3.57) and AdezapratyayayoH (8.3.59) :</p>";

echo "<p class = sa >अपदान्तस्य मूर्धन्यः (८.३.५५), इण्कोः (८.३.५७) तथा आदेशप्रत्यययोः (८.३.५९) :</p>";

display(0);

}

### /\* numvisarjanIyazarvyavAye'pi (8.3.58) \*/

// Not coded perfectly, only according to the need of vibhaktis.

$iN1 = array("i","I","u","U","f","F","x","X","e","o","E","O","h","y","v","r","l","k","K","g","G","N");

$pr1= '/(['.flat($iN1).'])([M+]\*)([s])([uA])/'; // num

$pr2= '/(['.flat($iN1).'])([H+]\*)([s])([uA])/'; // visarga

$pr3= '/(['.flat($iN1).'])([S+]\*)([s])([uA])/'; // z

$pr4= '/(['.flat($iN1).'])([z+]\*)([s])([uA])/'; // S

$pr5= '/(['.flat($iN1).'])([s+]\*)([s])([uA])/'; // s

$prr = '$1$2z$4'; // replacement

if ( (arr($text,'/['.flat($iN1).']([HSzs+]\*)[s][uA]/')|| (arr($text,'/['.flat($iN1).']([HSzs+]\*)[s][uA]/') && $num===1 )) && sub(array("+su","+sAm"),blank(0),blank(0),0) )

{

foreach ($text as $value)

{

if(in\_array(1,$num))

{

$val[] = preg\_replace($pr1,$prr,$value);

}

$val[] = preg\_replace($pr2,$prr,$value);

$val[] = preg\_replace($pr3,$prr,$value);

$val[] = preg\_replace($pr4,$prr,$value);

$val[] = preg\_replace($pr5,$prr,$value);

}

$text = array\_diff($val,$text);;

$text = array\_unique($text);

$text = array\_values($text);

$val=array();

echo "<p class = sa >By numvisarjanIyazarvyavAye'pi (8.3.58) :</p>";

echo "<p class = sa >नुम्विसर्जनीयशर्व्यवायेऽपि (८.३.५८) :</p>";

display(0);

}

### /\* hanteratpUrvasya (8.4.22) \*/

if( sub($upasarga,array("han","Gn"),blank(0),0))

{

$text = two($upasarga,array("han","Gn"),$upasarga,array("haR","Gn"),0);

echo "<p class = sa >By hanteratpUrvasya (8.4.22) :</p>";

echo "<p class = sa >हन्तेरत्पूर्वस्य (८.४.२२) :</p>";

display(0);

}

if( arr($text,'/[G][n]/') && arr(array($fo),'/[h][a][n]/') && !sub($upasarga,array("han","Gn"),blank(0),0))

{

$text = one(array("han","Gn"),array("haR","Gn"),0);

echo "<p class = sa >By hanteratpUrvasya (8.4.22) :</p>";

echo "<p class = sa >हन्तेरत्पूर्वस्य (८.४.२२) :</p>";

display(0);

}

### /\* raSAbhyAM no NaH samAnapade (8.4.1) \*/

// pUrvasmAdapi vidhau sthAnivadbhAvaH , pUrvatrAsiddhe na sthAnivat (vA 433) and tasya doSaH saMyogAdilopalatvaNatveSu (vA 440) are pending to code.

if($pada === "pratyaya" && sub(array("r","z"),array("n"),blank(0),0))

{

$text = two(array("r","z"),array("n"),array("r","z"),array("R"),0);

echo "<p class = sa >By raSAbhyAM no NaH samAnapade (8.4.1) :</p>";

echo "<p class = sa >रषाभ्यां नो णः समानपदे (८.४.१) :</p>";

display(0);

}

### /\* ekAjuttarapade NaH (8.4.12) \*/

// very bad coding for this. Difficult to teach this samAsa, before machine understands samAsa. Therefore enumeration method is used here.

if (sub(array("punarBU+nAm",),blank(0),blank(0),0))

{

$ekajuttarapada=1; // Because there is ekAc uttarapada in this samAsa.

}

$rasek = '/([rzfF])([aAiIuUfFxXeoEOhyvrkKgGNpPbBmM+]\*)([n])/'; // pattern which qualifies for this sUtra.

$ras1ek = '$1$2R';

if( $ekajuttarapada===1 && arr($text,$rasek) && $hohante===0 )

{

foreach ($text as $value)

{

$value1[] = preg\_replace($rasek,$ras1ek,$value);

}

$text = $value1;

echo "<p class = sa >By ekAjuttarapade NaH (8.4.12) :</p>";

echo "<p class = sa >एकाजुत्तरपदे णः (८.४.१२) :</p>";

display(0);

}

if (sub(array("SrIpAn+i","vftrahan+"),blank(0),blank(0),0) && !arr($text,'/[+]$/')) // Patch for zrIpANi

{

$text=one(array("SrIpAn+i","vftrahan+"),array("SrIpAR+i","vftrahaR+"),0);

echo "<p class = sa >By ekAjuttarapade NaH (8.4.12) :</p>";

echo "<p class = sa >एकाजुत्तरपदे णः (८.४.१२) :</p>";

display(0);

}

### /\* Patch to remove the + sign \*/

// For dvitva, + sign pauses many problems. Now we don't have to remember what was prakRti and what was pratyaya. Therefore we can afford to remove + sign now.

if (arr($text,'/['.pc('hl').' ][+]['.pc('hl').']/') || arr($text,'/[HM!][+]['.pc('hl').']/'))

{

$text = three($hl,array("+"," +"),$hl,$hl,array(""," "),$hl,0);

$text = three(array("H","M","!"),array("+"),$hl,array("H","M","!"),array(""),$hl,0);

}

if ( arr($text,'/['.pc('ac').'HM! ][+]['.pc('ac').']/') || arr($text,'/[HM!][+]['.pc('ac').']/') )

{

$text = one(array("+"),array(" "),0);

}

if ( arr($text,'/['.pc('ac').' ][+]['.pc('hl').'MH]/') || arr($text,'/[+]$/') )

{

$text = one(array("+"),array(""),0);

}

if ( arr($text,'/['.pc('hl').'][+]['.pc('ac').']/') )

{

$text = one(array("+"),array(""),0);

}

### /\* aTkupvAGnumvyavAye'pi (8.4.2) and na padAntasya (8.4.37) \*/

### /\* RvarNAnnasya NatvaM vAcyam (vA 4969) \*/

### /\* na padAntasya 8.4.37) \*/

// The issue is identifying samAnapada. Can't be coded properly as of now.

$ras = '/([rzfF])([aAiIuUfFxXeoEOhyvrkKgGNpPbBmM+]\*)([n])/';

$rasend = '/([rzfF])([aAiIuUfFxXeoEOhyvrkKgGNpPbBmM+]\*)([n])$/';

$ras1 = '$1$2R';

if (arr($text,$rasend) && $hohante===0)

{

echo "<p class = pa >By na padAntasya 8.4.37), application of aTkupvAGnumvyavAye'pi (8.4.2) is barred. </p>";

echo "<p class = pa >न पदान्तस्य (८.४.३७) से अट्कुप्वाङ्नुम्व्यवायेऽपि का निषेध हुआ है । </p>";

display(0);

}

if (arr($text,$ras) && !arr($text,$rasend) && $hohante===0)

{

foreach ($text as $value)

{

if (preg\_match('/([rzfF])([aAiIuUfFxXeoEOhyvrkKgGNpPbBmM+]\*)([n])/',$value) )

{

$value1[] = preg\_replace($ras,$ras1,$value);

}

else

{

$value1[] = $value;

}

}

$text = $value1;

$value1 = array();

if(arr($text,'/([rz])([aAiIuUfFxXeoEOhyvrkKgGNpPbBmM+]\*)([R])/') && $hohante===0)

{

echo "<p class = sa >By aTkupvAGnumvyavAye'pi (8.4.2) :</p>";

echo "<p class = sa >अट्कुप्वाङ्नुम्व्यवायेऽपि (८.४.२) :</p>";

}

if(arr($text,'/([fF])([aAiIuUfFxXeoEOhyvrkKgGNpPbBmM+]\*)([R])/') && $hohante===0)

{

echo "<p class = sa >By RvarNAnnasya NatvaM vAcyam (vA 4969) :</p>";

echo "<p class = sa >ऋवर्णान्नस्य णत्वं वाच्यम्‌ (वा ४९६९) :</p>";

}

display(0);

}

### /\* stoH zcunA zcuH (8.4.40) \*/

$stu = array("s","t","T","d","D","n"); // sakAra and tavarga.

$zcu = array("S","c","C","j","J","Y"); // zakAra and cavarga.

if(sub($stu,$zcu,blank(0),0))

{

$text = two(array("s"),$zcu,array("S"),$zcu,0); // sakAra followed by cavarga.

$text = two(array("t"),$zcu,array("c"),$zcu,0); // tavarga followed by cavarga.

$text = two(array("T"),$zcu,array("C"),$zcu,0);

$text = two(array("d"),$zcu,array("j"),$zcu,0);

$text = two(array("D"),$zcu,array("J"),$zcu,0);

$text = two(array("n"),$zcu,array("Y"),$zcu,0);

echo "<p class = sa >By stoH zcunA zcuH (8.4.40) :</p>";

echo "<p class = sa >स्तोः श्चुना श्चुः (८.४.४०) :</p>";

display(0);

}

### /\* stoH zcunA zcuH (8.4.40) and zAt (8.4.44) \*/

$zcu1= array("c","C","j","J","Y"); // zAt prevents application in case of zakAra being first letter. Therefore created a new array without zakAra.

if(sub($zcu1,$stu,blank(0),0))

{

$text = two($zcu1,array("s"),$zcu1,array("S"),0);

$text = two($zcu1,array("t"),$zcu1,array("c"),0);

$text = two($zcu1,array("T"),$zcu1,array("C"),0);

$text = two($zcu1,array("d"),$zcu1,array("j"),0);

$text = two($zcu1,array("D"),$zcu1,array("J"),0);

$text = two($zcu1,array("n"),$zcu1,array("Y"),0);

$text = two(array("S"),array("s"),array("S"),array("S"),0); // z+s=z+z

echo "<p class = sa >By stoH zcunA zcuH (8.4.40) and zAt (8.4.44) :</p>";

echo "<p class = sa >स्तोः श्चुना श्चुः (८.४.४०) तथा शात्‌ (८.४.४४) :</p>";

display(0);

}

### /\* anAmnavatinagarINAmiti vAcyam (vA 5016) \*/

$shtu = array("z","w","W","q","Q","R",); // SakAra and Tavarga.

if (sub($shtu,array("nAm","navat","nagar"),blank(0),0) && $allopo!==1)

{

$text = two($shtu,array("nAm","navat","nagar"),$shtu,array("RAm","Ravat","Ragar"),0);

echo "<p class = sa >By na padAntATToranAm (8.4.42) and anAmnavatinagarINAmiti vAcyam (vA 5016) :</p>";

echo "<p class = sa >न पदान्ताट्टोरनाम्‌ (८.४.४२) तथा अनाम्नवतिनगरीणामिति वाच्यम्‌ (वा ५०१६) :</p>";

display(0);

if (sub($shtu,array("Ravat","Ragar"),blank(0),0) && $allopo!==1)

{

$text = two($shtu,array("Ravat","Ragar"),array("R","R","R","R","R","R"),array("Ravat","Ragar"),0);

echo "<p class = sa >By STunA STuH (8.4.41) :</p>";

echo "<p class = sa >ष्टुना ष्टुः (८.४.४१) :</p>";

display(0);

}

if (sub($shtu,array("RAm"),blank(0),0) && $allopo!==1) // yaro'nunAsike'nunAsiko vA is mandatory in pratyayas. nAm being a pratyaya, we are displaying this message.

{

$text = two($shtu,array("RAm"),array("R","R","R","R","R","R"),array("RAm"),0);

echo "<p class = sa >By yaro'nunAsike'nunAsiko vA (8.4.45) and pratyaye bhASAyAm nityam (vA) :</p>";

echo "<p class = sa >यरोऽनुनासिकेऽनुनासिको वा (८.४.४५) तथा प्रत्यये भाषायां नित्यम्‌ (वार्तिक) :</p>";

display(0);

}

}

### /\* stoH STunA STuH (8.4.41) and na padAntATToranAm (8.4.41) and toH Si (8.4.43) \*/

$Tu = array("w","W","q","Q","R",); $tu = array("t","T","d","D","n");

if(((sub($shtu,$stu,blank(0),0)|| sub($stu,$shtu,blank(0),0))) && $allopo===1 )

{

echo "<p class = pa >STunA STuH (8.4.41) is prevented by sthAnivadbhAva of allopa. The same result can be obtained by asiddhatva of bahiraGga allopa in kAryakAlapakSa.</p>";

echo "<p class = pa >पूर्वस्मादपि विधावल्लोपस्य स्थानिवद्भावान्न ष्टुत्वम्‌ । कार्यकालपक्षे बहिरङ्गस्य अल्लोपस्य असिद्धत्वाद्वा ।</p>";

display(0);

}

if(((sub($shtu,$stu,blank(0),0)|| sub($stu,$shtu,blank(0),0))) && $allopo===0 )

{

$text = two(array("z"),$stu,array("z"),$shtu,0);

$text = two(array("s"),$shtu,array("z"),$shtu,0);

$text = two(array("t"),$Tu,array("w"),$Tu,0);

$text = two(array("T"),$Tu,array("W"),$Tu,0);

$text = two(array("d"),$Tu,array("q"),$Tu,0);

$text = two(array("D"),$Tu,array("Q"),$Tu,0);

$text = two(array("n"),$Tu,array("R"),$Tu,0);

if ($pada === "pratyaya" && (sub($Tu,$tu,blank(0),0)))

{

$text = two(array("w"),$tu,array("w"),$Tu,0);

$text = two(array("W"),$tu,array("W"),$Tu,0);

$text = two(array("q"),$tu,array("q"),$Tu,0);

$text = two(array("Q"),$tu,array("Q"),$Tu,0);

$text = two(array("R"),$tu,array("R"),$Tu,0);

}

echo "<p class = sa >By STunA STuH (8.4.41) and na padAntATToraNam (8.4.42) and toH Si (8.4.43) :</p>";

echo "<p class = sa >ष्टुना ष्टुः (८.४.४१), न पदान्ताट्टोरणाम्‌ (८.४.४२) तथा तोः षि (८.४.४३) :</p>";

display(0);

}

### /\* yaro'nunAsike'nunAsiko vA (8.4.45) \*/

// this is applicable to only sparzas.

$yara = array("J","B","G","Q","D","j","b","g","q","d","K","P","C","W","T","c","w","t","k","p"); // array of yar varNas.

$anunasikarep = array("Y","m","N","R","n","Y","m","N","R","n","N","m","Y","R","n","Y","R","n","N","m"); // Their corresponding replacement.

$anunasika = array("N","Y","R","n","m"); // array of anunAsika.

if (sub($yara,array("+"),$anunasika,0) && $pada === "pada")

{

$text = three($yara,array("+"),$anunasika,$anunasikarep,array("+"),$anunasika,1);

echo "<p class = sa >By yaro'nunAsike'nunAsiko vA (8.4.45) :</p>";

echo "<p class = sa >यरोऽनुनासिकेऽनुनासिको वा (८.४.४५) :</p>";

display(0);

}

if (sub($yara,array("+"),$anunasika,0) && $pada === "pratyaya")

{

$text = two($yara,$anunasika,$anunasikarep,$anunasika,0);

echo "<p class = sa >By yaro'nunAsike'nunAsiko vA (8.4.45) :</p>";

echo "<p class = sa >यरोऽनुनासिकेऽनुनासिको वा (८.४.४५) :</p>";

display(0);

}

### /\* nAdinyAkroze putrasya (8.4.48) \*/

if (sub(array('putrAdin'),blank(0),blank(0),0))

{

echo "<p class = sa >By nAdinyAkroze putrasya (8.4.48) - If Akroza is meant : The dvitva doesn't happen. Otherwise dvitva will happen.</p>";

echo "<p class = sa >नादिन्याक्रोशे पुत्रस्य (८.४.४८) - यदि आक्रोश के अर्थ में प्रयुक्त हुआ है, तब द्वित्व नहीं होगा । अन्यथा द्वित्व होगा ।</p>";

}

### /\* vA hatajagdhayoH (vA 5022) \*/

if (sub(array("putrahatI"),blank(0),blank(0),0))

{

echo "<p class = sa >By vA hatajagdhayoH (vA 5022) :</p>";

echo "<p class = sa >वा हतजग्धयोः (वा ५०२२) :</p>";

display(0);

}

if (sub(array('putrajagDI'),blank(0),blank(0),0))

{

echo "<p class = sa >By vA hatajagdhayoH (vA 5022) :</p>";

echo "<p class = sa >वा हतजग्धयोः (वा ५०२२) :</p>";

display(0);

}

### /\* cayo dvitIyAH zari pauSkarasAderiti vAcyam (vA 5023) \*/

if (sub(array("N","R"),prat('Sr'),blank(0),0))

{

$text = two(array("Nk","Rw"),prat('Sr'),array("NK","RW"),prat('Sr'),1);

echo "<p class = sa >By cayo dvitIyAH zari pauSkarasAderiti vAcyam (vA 5023) :</p>";

echo "<p class = sa >चयोः द्वितीयाः शरि पौष्करसादेरिति वाच्यम्‌ (वा ५०२३) :</p>";

display(0); $cayo=1;

} else {$cayo = 0; }

### /\*anaci ca (8.4.47)\*/

// Here the sudhI + upAsya - what about the Asy - Assy is possbile ? Code gives it. But there are 4 options. Code gives two only.

// The cause for using $hrasva instead of $ac is that the dIrgha vowels are debarred by dIrghAdAcAyANAm.

// Here instead of using pratyAhAra hl, we shall do manual enumeration of all the members. Because of "anusvAravisargajihvAmUlIyopadhmAnIyayamAnAmakAropari zarSu ca pAThasyopasaGkhyAtatvenAnusvArasyApyactvAt (in derivation of samskAra)

$hrasvaplus = array("M","!","'"); // additionalities mentioned in saMskAra derivation.

$hala1 = array("y","v","l","Y","m","N","R","n","J","B","G","Q","D","j","b","g","q","d","K","P","C","W","T","c","w","t","k","p","S","z","s","M",);

$hala2 = array("h","y","v","r","l","Y","m","N","R","n","J","B","G","Q","D","j","b","g","q","d","K","P","C","W","T","c","w","t","k","p","S","z","s","M",); // added h,y

if(sub($hrasva,$hala1,$hala2,0))

{

$text = dvitva($hrasva,$hala1,$hala2,array(""),2,1);

if (sub($dirgha,$hala1,$hala2,0))

{

echo "<p class = sa >By anaci ca (8.4.47) and dIrghAdAcAryANAm (8.4.52) :</p>";

echo "<p class = sa >अनचि च (८.४.४७): तथा दीर्घादाचार्याणाम्‌ (८.४.५२) :</p>";

}

else

{

echo "<p class = sa >By anaci ca (8.4.47):</p>";

echo "<p class = sa >अनचि च (८.४.४७):</p>";

}

display(1);

}

if(sub($hrasvaplus,$hl,$hala2,0))

{

$text = dvitva($hrasvaplus,$hl,$hala2,array(""),2,1);

if (sub($dirgha,$hl,$hala2,0))

{

echo "<p class = sa >By anaci ca (8.4.47) and dIrghAdAcAryANAm (8.4.52) :</p>";

echo "<p class = sa >अनचि च (८.४.४७): तथा दीर्घादाचार्याणाम्‌ (८.४.५२) :</p>";

}

else

{

echo "<p class = sa >By anaci ca (8.4.47):</p>";

echo "<p class = sa >अनचि च (८.४.४७):</p>";

}

display(1);

}

if(checkarray($dirgha,$hl,array('r','l'),blank(0))!==0 && $sthanivadbhav===1) // for function checkarray, see function.php

{

$text = dvitva($dirgha,$hala1,$hala2,array(""),2,1);

if (sub($dirgha,$hl,array('r','l'),0))

{

echo "<p class = sa >By anaci ca (8.4.47) and dIrghAdAcAryANAm (8.4.52) :</p>";

echo "<p class = sa >अनचि च (८.४.४७): तथा दीर्घादाचार्याणाम्‌ (८.४.५२) :</p>";

}

else

{

echo "<p class = sa >By anaci ca (8.4.47):</p>";

echo "<p class = sa >अनचि च (८.४.४७):</p>";

}

display(1);

}

### /\* By anaci ca (according to mahAbhASya example of vAkk) \*/

if (arr($text,'/['.flat($ac).']['.flat($hl).']$/') || (preg\_match('/['.flat($ac).']['.flat($hl).']$/',$first) && $input === $first ))

{

foreach ($text as $value)

{

$split = str\_split($value);

$post = $split[count($split)-1];

if (in\_array($post,$hl))

{

$pre = chop($value,$post);

$value1[] = str\_replace($value,$pre.$post.$post,$value);

}

else

{

$value1[] = $value;

}

}

$text = array\_merge($text,$value1);

$text = array\_unique($text);

$text = array\_values($text);

$value1 = array();

if (sub($dirgha,$hala1,$hala2,0))

{

echo "<p class = sa >By anaci ca (8.4.47) and dIrghAdAcAryANAm (8.4.52) :</p>";

echo "<p class = sa >अनचि च (८.४.४७): तथा दीर्घादाचार्याणाम्‌ (८.४.५२) :</p>";

}

else

{

echo "<p class = sa >By anaci ca (8.4.47):</p>";

echo "<p class = sa >अनचि च (८.४.४७):</p>";

}

display(1);

}

### /\* zaraH khayaH (vA 5019) \*/

$shara = array("S","z","s",); // zar varNas.

if (sub($shara,prat('Ky'),blank(0),0))

{

$text = dvitva($shara,prat('Ky'),array(""),array(""),2,1);

echo "<p class = sa >zaraH khayaH (vA 5019) :</p>";

echo "<p class = sa >शरः खयः (वा ५०१९) :</p>";

display(1);

}

### /\* aco rahAbhyAM dve (8.4.46) \*/

$rh = array("r","h"); // r,h

if (sub($ac,$rh,prat('yr'),0))

{

if (sub($rh,array("S","z","s"),$ac,0)) // patch to show prohibition by zaro'ci.

{

echo "<p class = sa >By zaro'ci (8.4.48) :</p>";

echo "<p class = hn >N.B.: zaro'ci bars application of aco rahAbhyAm dve</p>";

echo "<p class = sa >शरोऽचि (८.४.४८) :</p>";

echo "<p class = hn >शरोऽचि अचो रहाभ्यां द्वे के प्रयोग का निषेध करता है ।</p>";

display(0);

}

else

{

$text = dvitva($ac,$rh,prat('yr'),array(""),3,1);

echo "<p class = sa >By aco rahAbhyAM dve (8.4.46) :</p>";

echo "<p class = sa >अचो रहाभ्यां द्वे (८.४.४६) :</p>";

display(1);

}

}

### /\* triprabhRtiSu zAkaTAyanasya (8.4.50)\*/

if (checkarray($ac,$hl,$hl,$hl) === 1)

{

echo "<p class = hn >N.B.: By triprabhRtiSu zAkaTAyanasya (8.4.50), the dvitva is optionally not done in cases where there are more than three hals appearing consecutively. e.g. indra - inndra. </p>";

echo "<p class = hn >त्रिप्रभृतिषु शाकटायनस्य (८.४.५०) - तीन या उससे ज्यादा हल्‌ अगर हो तब शाकटायन के मत में द्वित्व नहीं होता है ।</p>";

}

### /\* sarvatra zAkalyasya (8.4.51) \*/

// It is not coded separately. It is sent as a message in all display function when 1 is selected as option.

### /\* dIrghAdAcAryANAm (8-4-52) \*/

// Not coded separately, because we did dvitva only for $hrasva, and not for 'ac'. So this is already taken care of.

### /\* jhalAM jaz jhaSi (8.4.53) \*/

while(sub(array\_diff(prat('Jl'),prat('jS')),prat('JS'),blank(0),0) !== false) // check whether we should remove jaz from jhal?

{

if(sub(prat('Jl'),prat('JS'),blank(0),0))

{

$text = two(prat('Jl'),prat('JS'),savarna(prat('Jl'),prat('jS')),prat('JS'),0);

echo "<p class = sa >By jhalAM jaz jhazi (8.4.53):</p>";

echo "<p class = sa >झलां जश्‌ झशि (८.४.५३):</p>";

display(0);

}

}

### /\* yaNo mayo dve vAcye (vA 5018) yaN in paJcamI and may in SaSThI)\*/

if (sub($hrasva,prat('yR'),prat('my'),0))

{

$text = dvitva(prat('yR'),prat('my'),array(""),array(""),2,1);

echo "<p class = sa >By yaNo mayo dve vAcye (yaN in paJcamI and may in SaSThI) (vA 5018) :</p>";

echo "<p class = sa >यणो मयो द्वे वाच्ये (यण्‌ पञ्चमी तथा मय्‌ षष्ठी) (वा ५०१८) :</p>";

display(1);

}

if (sub($dirgha,prat('yR'),prat('my'),0) && $sthanivadbhav ===1)

{

$text = dvitva(prat('yR'),prat('my'),array(""),array(""),2,1);

echo "<p class = sa >By yaNo mayo dve vAcye (yaN in paJcamI and may in SaSThI) (vA 5018) :</p>";

echo "<p class = sa >यणो मयो द्वे वाच्ये (यण्‌ पञ्चमी तथा मय्‌ षष्ठी) (वा ५०१८) :</p>";

display(1);

}

### /\* yaNo mayo dve vAcye (vA 5018) may in paJcamI and yaN in SaSThI)\*/

if (sub($hrasva,prat('my'),prat('yR'),0))

{

$text = dvitva(prat('my'),prat('yR'),array(""),array(""),2,1);

echo "<p class = sa >By yaNo mayo dve vAcye (may in paJcamI and yaN in SaSThI) (vA 5018):</p>";

echo "<p class = sa >यणो मयो द्वे वाच्ये (मय्‌ पञ्चमी तथा यण्‌ षष्ठी) (वा ५०१८) :</p>";

display(1);

}

if (sub($dirgha,prat('my'),prat('yR'),0) && $sthanivadbhav ===1)

{

$text = dvitva(prat('my'),prat('yR'),array(""),array(""),2,1);

echo "<p class = sa >By yaNo mayo dve vAcye (may in paJcamI and yaN in SaSThI) (vA 5018):</p>";

echo "<p class = sa >यणो मयो द्वे वाच्ये (मय्‌ पञ्चमी तथा यण्‌ षष्ठी) (वा ५०१८) :</p>";

display(1);

}

### /\* vA'vasAne (8.4.54) \*/

if (arr($text,'/['.pc('Jl').']$/'))

{

$text = last(prat('Jl'),savarna(prat('Jl'),prat('cr')),1);

echo "<p class = sa >By vA'vasAne (8.4.54) :</p>";

echo "<p class = sa >वाऽवसाने (८.४.५४) :</p>";

display(0);

}

### /\* khari ca (8.4.55) \*/

$Jl1 = array("J","B","G","Q","D","j","b","g","q","d","K","P","C","W","T","c","w","t","k","p","S","z","s","h"); // complete jhal.

$Jl2 = array("J","B","G","Q","D","j","b","g","q","d","K","P","C","W","T","h"); // jhal without car.

if ($cayo===0)

{

while(sub($Jl2,prat('Kr'),blank(0),0) !== false) // this rule can apply add infinitum.

{

if ( (sub($Jl1,prat('Kr'),blank(0),0) || $dhut === 1))

{

$text = two($Jl1,prat('Kr'),savarna(prat('Jl'),prat('cr')),prat('Kr'),0);

echo "<p class = sa >By khari ca (8.4.55) :</p>";

echo "<p class = sa >खरि च (८.४.५५) :</p>";

display(0);

}

}

}

if (sub(prat('cr'),prat('Kr'),blank(0),0) || $dhut === 1) // parjanyavallakSaNapravRttiH.

{

echo "<p class = sa >By khari ca (8.4.55) :</p>";

echo "<p class = hn >N.B. By khari ca (8.4.55), 'car' varNas give 'car' varNas only as their savarNa :</p>";

echo "<p class = sa >खरि च (८.४.५५) :</p>";

echo "<p class = hn >खरि च (८.४.५५) से चर्‌ वर्णों का सवर्ण चर्‌ ही रहता है ।</p>";

display(0);

}

### /\* aNo'pragRhyasyAnunAsikaH (8.4.57) \*/

if (arr($text,'/[aAiIuUfFxX]$/'))

{

$text = last(array("a","A","i","I","u","U","f","F","x","X"),array("a!","A!","i!","I!","u!","U!","f!","F!","x!","X!"),1);

echo "<p class = sa >By aNo'pragRhyasyAnunAsikaH (8.4.57) :</p>";

echo "<p class = sa >अणोऽप्रगृह्यस्यानुनासिकः (८.३.५७) :</p>";

display(0);

}

### /\* anusvArasya yayi parasavarNaH (8.4.58) and vA padAntasya (8.4.59) \*/

$mm = array("My","Mv","Mr","Ml","MY","Mm","MN","MR","Mn","MJ","MB","MG","MQ","MD","Mj","Mb","Mg","Mq","Md","MK","MP","MC","MW","MT","Mc","Mw","Mt","Mk","Mp"); // anusvAra+yay.

$pa = array("!yy","!vv","!rr","!ll","YY","mm","NN","RR","nn","YJ","mB","NG","RQ","nD","Yj","mb","Ng","Rq","nd","NK","mP","YC","RW","nT","Yc","Rw","nt","Nk","mp"); // its replacement.

if (sub(array("M"),prat('yy'),blank(0),0))

{

$text = one($mm,$pa,1);

echo "<p class = sa >By anusvArasya yayi parasavarNaH (8.4.58) and vA padAntasya (8.4.59) :</p>

<p class = hn >N.B.: The change of anusvARa to parasavarNa is mandatory for non padAnta conjoints. For padAnta conjoints, it is optional.</p>";

echo "<p class = sa >अनुस्वारस्य ययि परसवर्णः (८.४.५८) तथा वा पदान्तस्य (८.४.५९) :</p>

<p class = hn >पदान्त में पाक्षिक है । अपदान्त के लिए अनिवार्य है ।</p>";

display(0);

}

### /\* torli (8.4.60) \*/

$to = array("tl","Tl","dl","Dl","nl"); // combinations satisfying rule conditions.

$lirep = array("ll","ll","ll","ll","l!l",); // its replacement.

while(sub($to,blank(0),blank(0),0) !== false)

{

if (sub($to,blank(0),blank(0),0))

{

$text = one($to,$lirep,0);

echo "<p class = sa >By torli (8.4.60) :</p>";

echo "<p class = sa >तोर्लि (८.४.६०) :</p>";

display(0);

}

}

### /\* jhayo ho'nyatarasyAm (8.4.62) \*/

$Jy = array("Jh","Bh","Gh","Qh","Dh","jh","bh","gh","qh","dh","Kh","Ph","Ch","Wh","Th","ch","wh","th","kh","ph",); // combination satisfying condition.

$h1 = array("JJ","BB","GG","QQ","DD","jJ","bB","gG","qQ","dD","KG","PB","CJ","WQ","TD","cJ","wQ","tD","kG","pB",); // its replacement.

if (sub($Jy,blank(0),blank(0),0))

{

$text = one($Jy,$h1,1);

echo "<p class = sa >By jhayo ho'nyatarasyAm (8.4.62) :</p>";

echo "<p class = sa >झयो होऽन्यतरस्याम्‌ (८.४.६२) :</p>";

display(0);

}

### /\* zazCho'Ti (8.4.63) and ChatvamamIti vAcyam (vA 5025) \*/

$Jy = array("JS","BS","GS","QS","DS","jS","bS","gS","qS","dS","KS","PS","CS","WS","TS","cS","wS","tS","kS","pS",); // combination satisfying condition.

$h1 = array("JC","BC","GC","QC","DC","jC","bC","gC","qC","dC","KC","PC","CC","WC","TC","cC","wC","tC","kC","pC",); // its replacement.

$aT = array("a","A","i","I","u","U","f","F","x","X","e","o","E","O","h","y","v","r","l","Y","m","G","R","n"); // am varNas.

if(sub($Jy,$aT,blank(0),0) && $pada === "pada")

{

$text = two($Jy,$aT,$h1,$aT,1);

echo "<p class = sa >By zazCho'Ti (8.4.63) and ChatvamamIti vAcyam (vA 5025) :</p>";

echo "<p class = sa >शश्छोऽटि (८.४.६३) तथा छत्वममीति वाच्यम्‌ (वा ५०२५) :</p>";

display(0);

}

### /\* halo yamAM yami lopaH (8.4.64) \*/

$duplicate = array("NN","YY","RR","nn","mm","yy","rr","ll","vv",); // combination satisfying condition.

$dup = array("N","Y","R","n","m","y","r","l","v",); // its replacement.

$hl = array("k","K","g","G","N","c","C","j","J","Y","w","W","q","Q","R","t","T","d","D","n","p","P","b","B","m","y","r","l","v","S","z","s","h"); // hal varNas.

if (sub($hl,prat('ym'),prat('ym'),0))

{

$text = two($hl,$duplicate,$hl,$dup,1);

echo "<p class = sa >By halo yamAM yami lopaH (8.4.64) :</p>";

echo "<p class = sa >हलो यमां यमि लोपः (८.४.६४) :</p>";

display(0);

}

### /\* jharo jhari savarNe (8.4.65) \*/

if(sub(prat('hl'),prat('Jr'),prat('Jr'),0))

{

for ($i=0;$i<count(prat('Jr'));$i++)

{$kkk = array("k","K","g","G");

$ccc = array("c","C","j","J","S");

$www = array("w","W","q","Q","z");

$ttt = array("t","T","d","D","s");

$ppp = array("p","P","b","B"); // savarNa groups.

$text = three(prat('hl'),$kkk,$kkk,prat('hl'),blank(4),$kkk,1);

$text = three(prat('hl'),$ccc,$ccc,prat('hl'),blank(5),$ccc,1);

$text = three(prat('hl'),$www,$www,prat('hl'),blank(5),$www,1);

$text = three(prat('hl'),$ttt,$ttt,prat('hl'),blank(5),$ttt,1);

$text = three(prat('hl'),$ppp,$ppp,prat('hl'),blank(4),$ppp,1);

}

echo "<p class = sa >By jharo jhari savarNe (8.4.65) :</p>";

echo "<p class = sa >झरो झरि सवर्णे (८.४.६५) :</p>";

display(0);

}

### /\* Final Display \*/

echo "<p class = sa >Final forms are :</p>";

echo "<p class = sa >आखिरी रूप हैं :</p>";

display(0);

### /\* setting the $pada back to pratyaya for next use \*/

$pada="pratyaya";

$it = array();

$itprakriti = array();

$itpratyaya = array();

}

?>

</body>

</html>

1. [↑](#footnote-ref-0)