Rania — PS 4 — 2025-01-29

Very nice! Some comments below and on p. 10.

10/10

Section 11 Problems Cont.

In[73]:=

In[74]:=
```
(*11.16 Word cloud of last letters in words from WordList[]*)
WordCloud[StringTake[StringReverse[WordList[]], 1]]
(*11.17 Roman numberals for the year 1959*)
RomanNumeral[1959]
(*11.8 Maximum string length of any Ruman numeral year from 1 to 20*)
Take[Reverse[Sort[StringLength[Table[RomanNumeral[x], {x, 1, 2020}]]]], 1]
(*11.19 Word cloud of first charc of the Roman numerals up to 100*)
WordCloud[StringTake[Table[RomanNumeral[x], {x, 1, 100}], 1]]
(*11.20 Using Length to find the length of Russian alphabet*)
StringLength[StringJoin[Alphabet["Russian"]]]
(*11.21 Upercase Greek alphabet*)
ToUpperCase[Alphabet["Greek"]]
(*11.22 Bar Chart of letter numbers in "wolfram"*)
BarChart[LetterNumber["wolfram"]]
(*11.23 FromLetterNumber to make a string of 1000 random letters*)
StringJoin[FromLetterNumber[RandomInteger[25, 1000] + 1]]
(*11.24 Random five letter words*)
Table[StringJoin[FromLetterNumber[RandomInteger[25, 5] + 1]], 1000]
(*11.25 Wolfram translated in Greek*)
Transliterate["wolfram", "Greek"]
(*11.26 10 copies of wolf, ram emojis*)
StringJoin[Table["🐺🐏", 10]]
(*11.27 Arabic alphabet translated to English*)
Transliterate[Alphabet["Arabic"], "English"]
(*11.28 "A" in a black-white image*)
ColorNegate[Rasterize[Style["A", 200]]]
(*11.29 Interactive selector of a character in alphabet as an image*)
Manipulate[Rasterize[Style[FromLetterNumber[n], 100]], {n, 1, 26, 1}]
(*11.30 Interactive selector of black-
 on-white outlines from the alphabet by a menu*)
Manipulate[ColorNegate[EdgeDetect[Rasterize[Style[letter, 100]]]],
 {letter, Alphabet[]}]
(*11.31 "Vision simulator" blurs letter "A" from  to 50*)
Manipulate[Blur[Rasterize[Style["A", 100]], n], {n, 1, 50, 2}]
```

Your solution to 11.18 returns {13} instaead of 13. I did Max[StringLength[RomanNumeral[Range[2020]]]].

Nice that you did RandomInteger[25]+1 not RandomInteger[26] which will produce some space characters.

Out[74]=

Out[75]=

MCMLIX

Out[76]=

{13}

Out[77]=

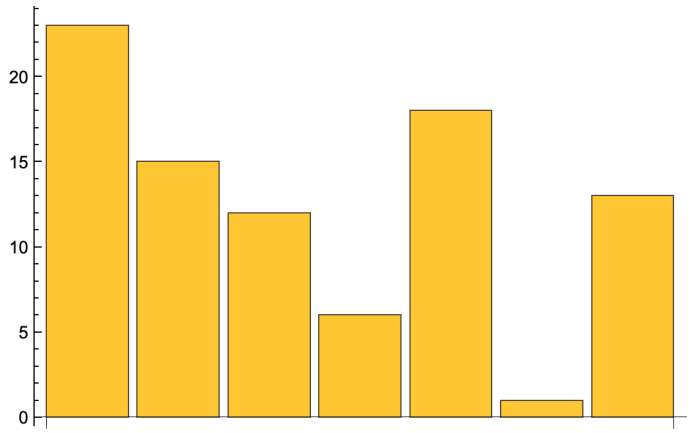
Out[78]=

33

Out[79]=

{Α, Β, Γ, Δ, Ε, Ζ, Η, Θ, Ι, Κ, Λ, Μ, Ν, Ξ, Ο, Π, Ρ, Σ, Τ, Υ, Φ, Χ, Ψ, Ω}

Out[80]=



Out[81]=

yhzncxwkkwvyggzksnanfpqrjedjnbwauetwajxppercbjoevqoakbapzmzrxalhllqaqtpuxabxgavnw
viclthgcbhmocstcemgepmpncenygsiunvrtldzeyguohmfshrimevdkfeiwwdnlvguaurychdrywa
hlakgbjrravnsuxtnnqedmslwxzwuipcypbglmhbjfriklyvbhhuxmzxipylnhuloifotzrbjqhtby
kktceqjekyoazsuydxdzjwxbqmjzeajngzrvpmxenvawyfftromxnskgzdzkkmewyzvrbnjyayxhxe
qcnabbvwdnbjubuzinpijqvmznwixzayuewfqrwugnlyydcrkcgtqxlugweksavnochlzwnxuaijgl
jnlpzssiidlaxrkyjzoptxcppuhibfhnznpyfxycaaababmkcfdhwywljvccnlpbumebdflelpisbn
bybyeewhicckjnukvlzcxnteyjyprlaccgmticcpjjniokmcwmxivtcdeiubvdowktmzfdbmrhqicd
tpmjphqaxisfawyifupkftokdrinpeqntdyzesbhegnxcorkgszmfhtkploutfvwkpxvdveojjuzju
uaxvzheqokevxgzwnlclgvitevltzudibdufudqztkjczorreevhiokngastbisaoigidvlxdjitvx
hvvjkshxqbgjiosehpucfgkxubfaathurgnwxbjoxnviexamlekqdllftmvwxsdcytblexpaazfgof
cyvslpkdyicgerhdxvjtfybpbtowliiaodltrcxhyyiccpzsnhpypquadxtpsqrqobdcedccdzlmbh
yiopcnxztnuqugghjrxhitfgadqbemqvzcrhlwvdeegqgjzofuugmgnimcsuzisuovhwresvkwjzre
hkuxcxaxsynkznbqdaubheebhaqxgyumeulqkiaryvdgwkrzvkgfqbtpabgap

Out[82]=

{netkf, pnlvk, gjbdy, gcsat, tbakj, ureaa, ewhpb, oujcv, xosgs, btfki, pwkek,
 bhork, pliky, nafaa, fjvyr, qsmon, prfuo, ypxly, batbu, faerq, mzrbc, lrmkk,
 zpzcu, muaeq, foyuv, gxntk, ynfhc, uogni, jgimu, qmehg, sdkug, bepnp, dztjt,
 nfinu, emfig, lvopt, ebohv, krvtk, zvsrg, dusgv, cetoi, sxvhy, mfkez, ohfhv,
 ogirz, wgatc, zjtoq, wduim, eyged, dqhfe, dzupm, exfud, zsumd, swdfu, fhfou,
 ttbsh, rqmtb, uldkt, ifjpj, fnvrr, nyydb, gcxvs, itlqu, rbgfy, jhoec, gqpzq,
 xnkxi, pzkan, ayhnw, eykhz, dfkqt, nlejn, gumnv, awqnd, pjmva, tkddb, wrsjh,
 wxeza, yxcbt, qwttx, mqmxg, fzoeq, rhczf, wassx, jhkvr, wpzry, yndmh, ckoxl,
 bmrve, txlff, wghmp, caufq, loiys, gydhe, rutnx, cqszx, uaidw, cfrst, xdyqw,
 eadpi, vngqp, wyxek, jjxsv, mtpwy, aafoc, gdfzg, vrtju, kxraz, wcfer, qsdfx,
 lqeri, vygtu, zbhtk, cbmqp, eodyw, kuveu, hbiyx, cicxx, rcxng, zfcfg, ahtst,
 yaqed, ercra, bjduj, fkrgu, zrrci, zkgac, sqfoh, nmuew, zjqjo, toajo, cjxeq,
 cvhoa, iwlim, euvtt, lvfla, kohla, nehop, wztqs, muzjq, msxja, spthx, eysge,
 catdx, ozflg, hqexd, urrkp, myhvu, peptp, vnyte, vwswn, jekjc, yxwlx, meuqm,
 ypezu, etnks, omuwl, erjst, izhxa, lrenz, oqtnm, perfu, gggso, fushr, qlfgs,
 mtuzj, xzagb, mrfsy, mrylk, zltku, mlpnz, zcrhk, wtjgp, qwxrj, oduij, yembg,
 bmuxc, fvdjy, dubml, qekwd, faeov, gpwxk, gimso, fpezg, hhvpy, jlxud, ogbia,

yzrre, jmxep, ailro, ydntk, ntexa, sbyev, lxjzb, tjvsy, nduoz, taehn, laxlg,
ayzjo, imrju, bfikz, xsjpq, mtzol, ydgkq, mhope, gygfc, ljddf, jkwmg, sliqs,
ygizc, ntxoo, lyomq, fhhxm, nebjp, byghn, zhnii, tqdye, xdqtu, jhqac, exuoh,
ztfqe, ynweh, cylyj, cdqer, oebps, izhmg, gfoqo, axxxx, qvaez, scfmx, xfvat,
rcebw, mktdn, uozkb, ujrhp, vgirh, cuugx, klfjl, bdvja, rrwya, vdfpr, nwkne,
sodbn, dikvg, luklg, ptqtf, pstyp, xbgjx, ojjaa, syrzw, bddnz, ntzmb, urcdh,
kygws, tjpxq, uayft, tgmhc, ovjvd, gsywz, bxpgu, pqlxo, rsgpg, fspel, ntgsh,
diimp, vmeym, yjfos, bfyfq, aufpq, zodws, wjlzt, wlvoo, jbdfc, maqfd, oollq,
gwyae, hmhnd, fxfew, dwnsb, ianhw, gmxch, zfhph, tthfd, zgkdf, sakdi, idago,
pqkus, tcqvr, ofrwu, tvxzy, qsnyq, xoboe, xqcxu, kakqj, tzkdd, haehe, hbdah,
hvtmv, xsjuc, vejlb, rkcpm, stihe, erxic, bwdib, gftqv, jufzr, zekkd, levix,
ydtlw, pqqjk, pvtpy, wiscv, khugb, aqbfk, mtiks, msnpr, papfm, nyjcz, lvkmr,
gcbei, hovdl, dmopx, zktxj, exuip, vtqbh, yzrrv, wzogv, xtdwn, dgwrk, covla,
iyjdk, zkbyg, houht, qufsb, vhmzs, pxemr, esacv, usfjq, ksixc, rnmyy, nasfg,
mmgxe, heahj, bwdgn, jlcyw, odoov, menuh, uimlk, jsvtj, druvo, ytaum, zdmgp,
wnyhu, cggpz, aqmyw, cgqwu, luvja, yagqz, jfitb, ojnkm, famuc, ftlmo, cgicu,
bhqwh, gpmzg, pmikx, xgouv, aanzi, icpqy, tfqtt, poaib, nmydl, fvswz, jvplj,
dkchm, uqths, qnxan, bqjwt, nkeil, vclaj, iunxp, vplxx, yhlmk, gxdua, urvjk,
fnopn, dutbe, tjzjt, fbshy, xtpuy, kwbff, katjj, oqabg, uwctw, lcmrk, wajka,
pomfb, lzucx, idnwm, goctc, yuzig, fzdla, vlkdu, gyjgz, tafpe, xeedv, xwdhr,
mzpvp, floym, yknvz, xevjk, igted, bgynd, omttp, tzwwq, rjzru, grsfu, mmish,
lscmu, nfgtm, owkgi, yzqbp, uejfa, umudd, aduua, mupqr, whjsw, kmvhf, jpgok,
fhdns, rzxax, sgxkd, hfpwy, llzsb, gzvyq, bwmze, italo, dinur, bsngr, mfjys,
hrpkk, eeyop, yehpn, bgcts, cqebp, fqmyr, plpdd, vfdmt, rshho, rsrip, czdnr,
ljgmf, ssomm, ucehl, cuojx, xhuhy, oxekl, ttajc, dvkmu, pvyjj, lzbfh, jhntv,
srpuh, tbbks, qjlot, sksjc, gzpwq, qcacp, hrkpx, ruvds, mwsay, kyxww, qfiux,
sbfvc, dklfs, ulzcb, wlmoh, jyrfc, pgseg, ygkbz, agabo, dhpsh, jcroq, rtnnl,
dfygp, scshz, wzosb, kkxrf, voefq, llirv, dwpic, kdnes, eudgz, ycvmg, qnoqu,
gkqfm, kfwwx, uprsu, ewdzd, jlsiv, coers, tkjvz, zofqj, lpgqu, mdalf, ujlsx,
lgnob, monut, tlzap, mzfdw, sirbn, pduji, heifs, rxzjt, yajem, mfcnp, pscfz,
owoih, itesb, dencc, fhyhy, jjmsz, pyzcq, dtdqh, othbb, iegpr, oksvc, oauwv,
wfthj, fskmu, mnffw, dvqsj, pqpvx, ykcap, jwrna, isual, sxtkr, lszwk, ruacj,
ivlmc, msmau, mtdew, amqil, ebwpz, hdobr, icxon, kiwea, mwabv, eimom, hrfky,
tfhsi, hlqmw, imgyk, sdsqu, fdxjq, pxrjq, gjoah, baugn, ujdkg, ntikt, nadbd,
kdubs, tccyj, pgvtq, relxo, xhctz, amqop, whwwy, xhxvl, kspcp, eepce, zdgno,
awvte, hdece, wpevn, vuwsu, knpmy, xbjoz, bdbwp, qsgrc, qltar, tssvn, jjujn,
zzywe, xpmqv, ooigt, lranu, rofke, cvehw, qoqcc, rlytq, ralcx, wjneh, ammjo,
fadth, mqmpg, sccfr, xwmvv, xbumd, aajhc, wotwp, uyxfz, scbvo, tiqhu, erhhy,
aaali, mqung, puznk, eaqvr, xatqn, gswlg, xqxmt, vwlrx, yjygh, wercv, agfvo,
lebgf, byjgn, znbai, hlmfc, ynwci, dzxwa, fodfs, loucd, hwhoj, zoadi, uvrop,
lvhnu, womsk, mwvxs, spwhe, hzuxy, twjai, jmvcx, ygajj, xqrao, owmza, riitu,
avuer, ccnba, yekxb, uczwq, bcsyu, fnafx, zqcbr, wljno, fzthk, bdctc, fowim,
dvqju, nzeky, cbbkz, jwfve, hjzhl, qqhid, xiwqg, mctgh, izzbc, tyrvn, ljsae,
kmpch, hrhka, zqunc, quexx, ljjoj, ldrdv, poajt, idetp, nzpux, mjaaw, xyfcz,

wacvo, ldgli, ulrss, npslv, tykdp, zpbtl, yesli, mqtml, rwshk, wuxnd, lnbms,
tzacq, xnnuv, xknqb, ouoop, sogib, fkghr, asnpp, neabh, vvkxh, xxqfn, vuqme,
kuwkd, deleo, pqdfa, wzjxz, gexxo, vmqdb, ttabi, pkiyd, vjhxz, bapgb, lexfj,
dnodq, cwbkb, bhbed, zlvqc, fwzsk, zlgrv, jzrgp, rcwzk, lhhbv, vgklh, xrljh,
ppfya, cgswh, ydymw, bxwrv, iobpl, xualt, fwapn, hzcxk, zmxuu, bqwri, pltjd,
gnrli, hmyuf, xpspz, quchs, uraas, naicq, hbard, oxzri, tuhii, zkjzp, mdsbh,
lncjo, sfryh, yokll, jbard, vbqkw, ylogf, rxlsg, kfsrv, ecpek, zycrn, jjrki,
ptidb, glanh, crpum, avxfi, oyjyg, kddlv, euxwa, dzqnb, mhdjd, tnlou, kusdo,
dnoqv, yakzg, hvdwj, tdbkd, pkgci, wuwxj, aaxec, xnqfo, fmaxe, gftxe, ynfqb,
xwozb, uyrnx, mqdge, pjrew, olejs, cysey, qigry, xsaca, kmwhr, cglvq, vjbur,
rliua, mltiv, yhvqj, jhkpr, uivps, zvnvm, pdhqc, mryof, tgrhy, lgtck, agwxj,
ldjxb, jgwes, prkdd, qntmg, xqkvb, ywhci, qvidx, dwgvu, whvwy, anbcv, gpuoe,
egxol, kehaf, fmgin, lxezq, lkfcb, obnkv, yesei, euwen, hrvbo, eiwbv, terph,
ryspu, sugrq, wpxxz, ipxvg, eephv, yrhgn, dummq, fuhmv, nqaxv, xqoai, daoiu,
eurjh, vrjth, zvdkk, duvjp, qgiad, vtzsu, otvxl, kwjjz, knijy, vponk, roaic,
ccgxy, dubfg, rpcyj, jtcci, ybjei, kxzuv, fbxkb, aaksb, grgxq, cflrb, pcpme,
bavku, fmqwm, rdjgy, jvikh, gosbj, hhken, jqyoy, lxowo, uvzuz, bypup, wvkma,
nrzwd, mtbin, lzhzv, zizfi, zgmih, bfxju, hioad, mzcmd, aymtf, mqxgh, mrcsz,
kqmrv, pdewn, epfpz, lqsgm, oenpj, cvoiy, jphrv, smden, jictm, hyeqt, csdvi,
tblwt, rvzex, qobqo, nldic, lxwdk, okbuu, vzazc, atljh, wqwsp, vdxxf, yjlup,
phacc, kwgrv, zsizx, wmfmm, nezhd, syckj, gcmxp, iodok, qtlbq, xuqdh, lfozt,
mcmky, pkzbr, fnzct, lskqo, evooz, cpvja, efdai, tdyfv, qnzvr, coego, tgzgy,
qcrba, fpcys, vhmax, gmgcc, hkcss, dhvyh, fewyo, armis, ctjbb, mdxgy, qzvpc,
nwrso, qggwz, ojsqb, jjqqe, inolp, phtks, jfslx, moaza, ekjed, wlglt,
iviin, fcrzs, npcqe, dioyo, nyzzo, naira, naugo, bnxbv, xwfnm, gpfdm, pxjtu,
aflle, fbejt, bxhvw, esetg, jrhyf, qhhhk, zrhwt, qbdmz, fyqtz, jospu, wzfma,
keoxw, rneil, bpktn, jhnyx, zwlkk, ckekr, xtoum, ziufy, crenx, zbbxm, uaqnp,
gxwll, pzjzn, phrkc, hvkxn, nqlka, yomnd, umglv, qacoo, qvrvb, cmstg, gynvb,
wxvfj, rvcwg, cxgsl, tjjmq, uqbuc, kybli, uaemu, jucak, lekal, araqb, fmykd,
phdnt, xztcz, zasvj, aimne, gkkfu, nhuas, mirhd, aipdf, jbprb, dcqfy, qesqu}

Out[83]=

βολφραμ

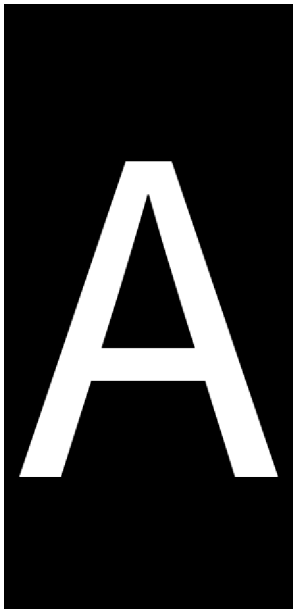
Out[84]=

🐺🐏🐺🐏🐺🐏🐺🐏🐺🐏🐺🐏🐺🐏🐺🐏🐺🐏🐺🐏

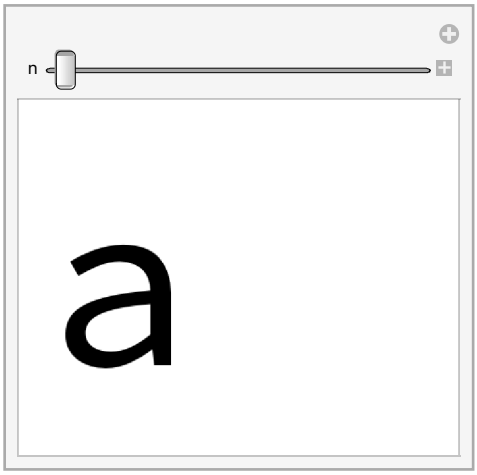
Out[85]=

{ạ, b, t, tẖ, j, ḥ, kẖ, d, dẖ, r, z, s, sẖ, ṣ, ḍ, ṭ, ẓ, ʿ, gẖ, f, q, k, l, m, n, h, w, y}

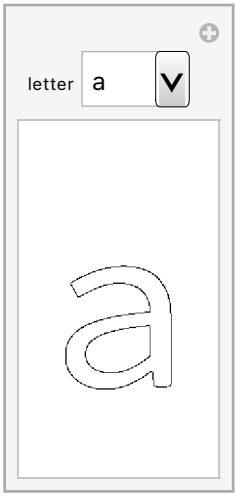
Out[86]=



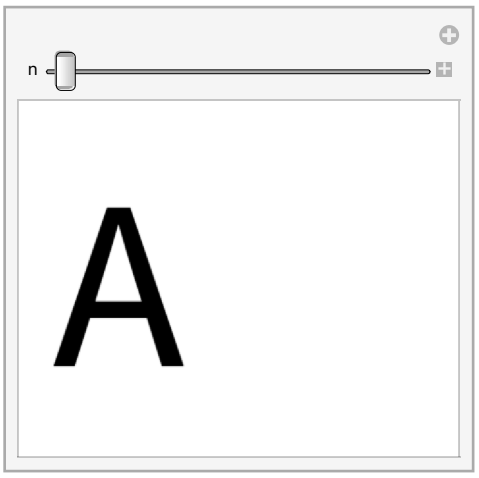
Out[87]=



Out[88]=



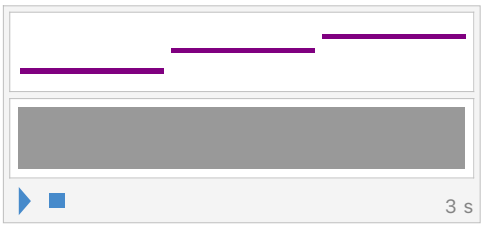
Out[89]=



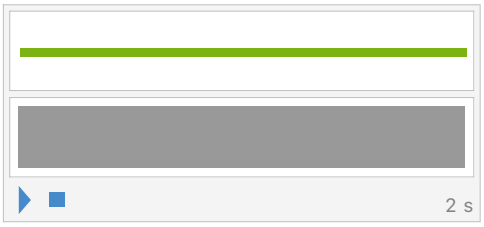
Section 12 Problems

In[90]:=
```
(*12.1 Sequence of notes with pitches 0,4,7*)
Sound[{SoundNote[0], SoundNote[4], SoundNote[7]}]
(*12.2 Two seconds of A note on Cello*)
Sound[SoundNote["A", 2.0, "Cello"]]
(*12.3 Riff of notes from pitch 0 → 48 in steps of 1,
with each note lasting 0.05 seconds*)
Sound[Table[SoundNote[p, .05], {p, 0, 48, 1}]]
(*12.4 Sequence of notes going from pitch 12→0 in steps of 1*)
Sound[Reverse[Table[SoundNote[p, 1], {p, 1, 12}]]]
(*12.5 Sequence of 5 notes,
staring at middle C and going up by octave at a time*)
Sound[Table[SoundNote[p, 1], {p, 5, (8 * 4 + 5), 8}]]
(*12.6 Sequence of 10 notes on a trumpet with random pitches from 0 →
 12 and duration 0.2 seconds*)
Sound[Table[SoundNote[RandomInteger[12], .2, "Trumpet"], 10]]
(*12.7 Sequence of 10 notes with random pitches up
 to 12 and random durations up to 10 tenths of a second*)
Sound[Table[SoundNote[RandomInteger[12], RandomInteger[10] / 10], 10]]
(*12.8 0.1-second notes with pitches given by the digits of 2^31*)
Sound[Table[SoundNote[p, .1], {p, IntegerDigits[2^31]}]]
(*12.9 Sound from letters in CABBAGE, each note for 0.3 seconds in guitar*)
Sound[Table[SoundNote[p, .3, "Guitar"], {p, Characters["CABBAGE"]}]]
(*12.10 0.1-second notes with pitches given
  by the letter numbers of the characters in “wolfram”*)
Sound[Table[SoundNote[p, .1], {p, LetterNumber["wolfram"]}]]
```

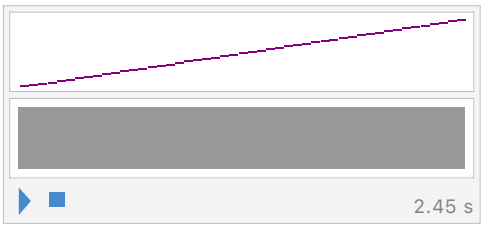
Out[90]=



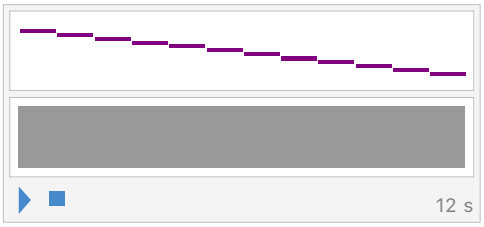
Out[91]=



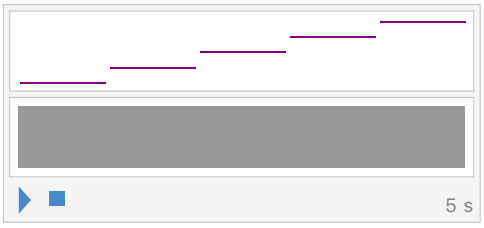
Out[92]=



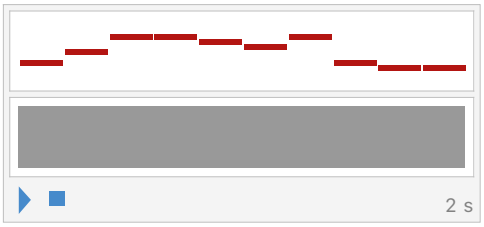
Out[93]=



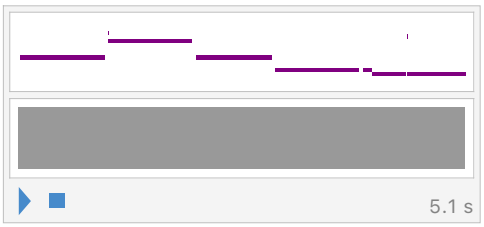
Out[94]=



Out[95]=



Out[96]=



Out[97]=

Out[98]=

Out[99]=

Section 13 Problems

In[100]:=

```
(*13.1 A 12x12 multiplication table *)
Grid[Table[a * b, {a, 12}, {b, 12}]]
(*13.2 A 5x5 for Roman Numerals*)
Grid[RomanNumeral[Table[a * b, {a, 5}, {b, 5}]]]
(*13.3 10x10 grid of random colors*)
Grid[Table[RandomColor[], 10, 10]]
(*13.4 10x10 grid of randomly colored random integers between 0 and 10*)
Grid[Table[Style[RandomInteger[10], RandomColor[]], 10, 10]]
(*13.5 Grid of all possible strings
 consisting of pairs of letters of the alphabet*)
Grid[Table[StringJoin[{a}, {b}], {a, Alphabet[]}, {b, Alphabet[]}]]
(*13.6 Visualize {1,4,3,5,2} with a pie chart,number line,
line plot and bar chart.Place these in a 2x2 grid*)
numList = {1, 4, 3, 5, 2}
Grid[{{PieChart[numList], NumberLinePlot[numList]},
  {ListLinePlot[numList], BarChart[numList]}}] (*ALL IN THE SYNTAX!*)
(*13.7 Array plot of hue values x*y where x and y run from 0→
 1 in steps of 0.05*)
ArrayPlot[Table[Hue[(x * y) / 20], {x, 20}, {y, 20}]](*THIS IS SO COOL*)
(*13.8 Array plot of hue values x/y,
where x and y each run from 1 to 50 in steps of 1*)
ArrayPlot[Table[Hue[(x / y)], {x, 50}, {y, 50}]]
(*13.9 Array plot of the lengths of Roman
 numeral strings in a multiplication table up to 100x100*)
ArrayPlot[Table[StringLength[RomanNumeral[a * b]], {a, 100}, {b, 100}]]
```

Yes, it can be very compact :)

The hue plots are cool. On this one he meant you to divide by 400!

I did

```
ArrayPlot[
   Table[Hue[i j],
   {i, Range[0, 1, 0.05]},
   {j, Range[0, 1, 0.05]}
]
```

I thought the last one was perhaps the coolest.

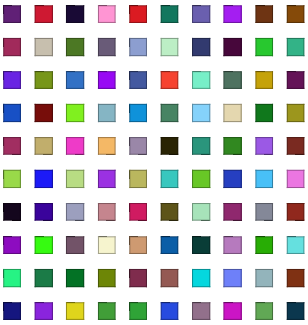
Out[100]=

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

Out[101]=

I	II	III	IV	V
II	IV	VI	VIII	X
III	VI	IX	XII	XV
IV	VIII	XII	XVI	XX
V	X	XV	XX	XXV

Out[102]=



Out[103]=

8	3	4	5	10	6	7	9	4	0
2	5	9	6	6	3	6	2	10	4
0	8	7	3	7	1	9	0	7	6
3	10	6	2	5	3	1	2	8	2
7	9	8	4	5	8	2	5	7	4
9	7	2	7	7	10	4	2	6	3
4	0	2	7	7	7	4	1	6	9
8	7	6	7	10	10	3	7	0	3
2	7	0	6	1	9	3	5	1	0
2	0	1	9	9	4	1	5	2	6

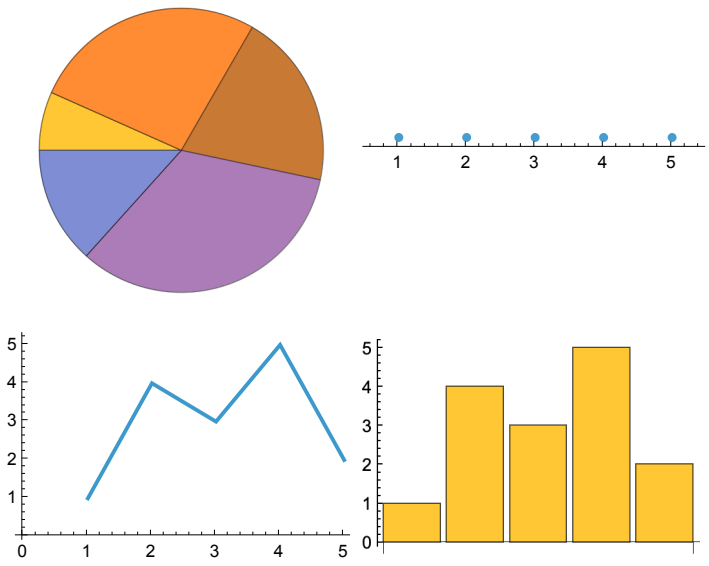
Out[104]=

```
aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az
ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp bq br bs bt bu bv bw bx by bz
ca cb cc cd ce cf cg ch ci cj ck cl cm cn co cp cq cr cs ct cu cv cw cx cy cz
da db dc dd de df dg dh di dj dk dl dm dn do dp dq dr ds dt du dv dw dx dy dz
ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ew ex ey ez
fa fb fc fd fe ff fg fh fi fj fk fl fm fn fo fp fq fr fs ft fu fv fw fx fy fz
ga gb gc gd ge gf gg gh gi gj gk gl gm gn go gp gq gr gs gt gu gv gw gx gy gz
ha hb hc hd he hf hg hh hi hj hk hl hm hn ho hp hq hr hs ht hu hv hw hx hy hz
ia ib ic id ie if ig ih ii ij ik il im in io ip iq ir is it iu iv iw ix iy iz
ja jb jc jd je jf jg jh ji jj jk jl jm jn jo jp jq jr js jt ju jv jw jx jy jz
ka kb kc kd ke kf kg kh ki kj kk kl km kn ko kp kq kr ks kt ku kv kw kx ky kz
la lb lc ld le lf lg lh li lj lk ll lm ln lo lp lq lr ls lt lu lv lw lx ly lz
ma mb mc md me mf mg mh mi mj mk ml mm mn mo mp mq mr ms mt mu mv mw mx my mz
na nb nc nd ne nf ng nh ni nj nk nl nm nn no np nq nr ns nt nu nv nw nx ny nz
oa ob oc od oe of og oh oi oj ok ol om on oo op oq or os ot ou ov ow ox oy oz
pa pb pc pd pe pf pg ph pi pj pk pl pm pn po pp pq pr ps pt pu pv pw px py pz
qa qb qc qd qe qf qg qh qi qj qk ql qm qn qo qp qq qr qs qt qu qv qw qx qy qz
ra rb rc rd re rf rg rh ri rj rk rl rm rn ro rp rq rr rs rt ru rv rw rx ry rz
sa sb sc sd se sf sg sh si sj sk sl sm sn so sp sq sr ss st su sv sw sx sy sz
ta tb tc td te tf tg th ti tj tk tl tm tn to tp tq tr ts tt tu tv tw tx ty tz
ua ub uc ud ue uf ug uh ui uj uk ul um un uo up uq ur us ut uu uv uw ux uy uz
va vb vc vd ve vf vg vh vi vj vk vl vm vn vo vp vq vr vs vt vu vv vw vx vy vz
wa wb wc wd we wf wg wh wi wj wk wl wm wn wo wp wq wr ws wt wu wv ww wx wy wz
xa xb xc xd xe xf xg xh xi xj xk xl xm xn xo xp xq xr xs xt xu xv xw xx xy xz
ya yb yc yd ye yf yg yh yi yj yk yl ym yn yo yp yq yr ys yt yu yv yw yx yy yz
za zb zc zd ze zf zg zh zi zj zk zl zm zn zo zp zq zr zs zt zu zv zw zx zy zz
```

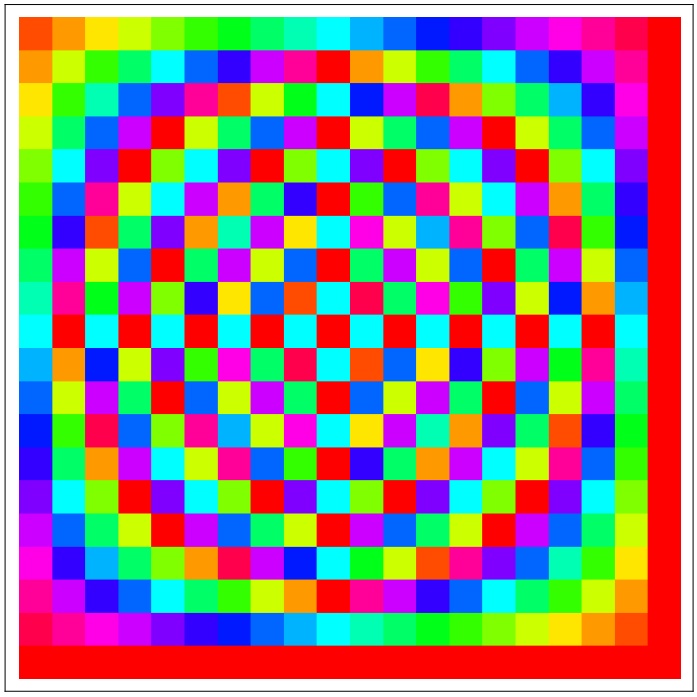
Out[105]=

```
{1, 4, 3, 5, 2}
```

Out[106]=



Out[107]=



Out[108]=

Out[109]=