

COSC 2436 Homework #4

Overview:

Exploiting patterns is an important part of breaking codes. It turns out that humans are poor at randomness, and often exhibit subtle breaks from randomness that can be exploited. One such example would be generating a set of random letters or a random password. A truly random password could include any character and may include repeats of characters. A human would feel that this is somehow not random enough and so might choose a key from the left side of the keyboard, followed by one on the right, and so on. This leads to an exploitable pattern. We will explore such patterns in actual words by using trees.

In this case, we will read two sets of letters from a file and use them to parse a large number of words from the **yawl.txt** file using the sets of letters. This file contains a large number of words that are generally accepted for word puzzles. We will then find words that match a number of requested letter patterns.

Input files

- The input files contain several lines indicating the details of the letter sets and requested patterns. All input files are well formed with valid values. There is no need for exception handling on the inputs and all files follow the same format.
- The first two lines provide sets of letters for a binary split. The lines begin with a single character that represents the set of letters. The remainder of the line is the letters that are part of this set, each separated by a comma and a space. **The two sets need not be disjoint (they may include overlaps) and may not include all the letters.** There will be exactly two non-empty sets of letters.
- The next lines are numbered and contain only the characters representing the two letter sets. They provide a requested pattern for words. There will always be requests, but the number of requests may vary. They will always be numbered starting from one and increasing by one. The requests will always be well formed and contain only the two characters representing the letter sets.

For example, the **input1.txt** contains:

```
L: Q, W, E, R, T, A, S, D, F, G, Z, X, C, V, B  
R: Y, U, I, O, P, H, J, K, L, N, M  
1. LRLR  
2. RRRR  
3. LLLL
```

The first line is designated “**L**” and includes all the letters that can be typed with the left hand on a standard QWERTY keyboard. The second line is designated “**R**” and represents all the letters that can be typed with the right hand. The next three lines are numbered and include a pattern of letters. The first pattern, for example, is words formed from letters typed with the left hand, then right, then left, and then right. The second pattern is the four-letter words that can be typed with only the right hand. The third pattern is the four-letter words that can be typed with only the left hand.

Output files

- The output files should contain one line for each of the requested patterns. The lines should be numbered to match the inputs, with a period and space following the number. The lines should contain all the words that match the requested pattern, separated by a comma and space.
- If there are no matching words, the line should contain the message: **No Matching Words**

For example, the **output1.txt** should contain:

1. AHEM, AIRN, AIRY, AITU, ALAN, ALAP, ALAY, ALSO, ALTO, AMAH, AMBO, AMEN, ANAL, ANAN, ANTI, APAY, AUTO, AYAH, BHEL, BIEN, BIRK, BIRL, BIRO, BISH, BISK, BITO, BLAH, BLAM, BLEY, BOAK, BOCK, BODY, BOGY, BORK, BORN, BOSH, BOSK, BOTH, BOWL, BOXY, BOZO, BUBO, BUBU, BUCK, BUDO, BUFO, BURK, BURL, BURN, BURP, BURY, BUSH, BUSK, BUSY, BYRL, CHAI, CHAL, CHAM, CHAO, CHAP, CHAY, CIAO, CIEL, CIRL, CITO, CITY, CLAM, CLAN, CLAP, CLAY, CLEM, COAL, COCK, COCO, CORK, CORM, CORN, CORY, COSH, COSY, COTH, COWL, COWP, COWY, COXY, COZY, CURL, CURN, CUSH, CUSK, CUSP, CYAN, DHAK, DHAL, DIAL, DICL, DICK, DICY, DIDO, DIDY, DIEL, DIRM, DIRL, DISH, DISK, DIVI, DIXI, DIXY, DOBY, DOCK, DODO, DOEK, DOEN, DOGY, DORK, DORM, DORP, DORY, DOSH, DOTH, DOTY, DOWL, DOWN, DOWP, DOXY, DUAL, DUAN, DUCI, DUCK, DUEL, DURN, DURO, DUSH, DUSK, DUTY, ELAN, EMEU, ENVY, EOAN, EORL, EUGH, EURO, EYEN, EYRY, FICO, FIGO, FIGR, FIRK, FIRM, FIRN, FISH, FISK, FLAK, FLAM, FLAN, FLAP, FLAY, FLEY, FOAL, FOAM, FOCI, FOEN, FOGY, FORK, FORM, FOWL, FOXY, FOZY, FUCI, FUCK, FUEL, FUGU, FURL, FURO, FURY, GIEN, GIRL, GIRN, GIRO, GISM, GLAM, GLEI, GLEN, GLEY, GOAL, GOBO, GOBY, GOEL, GOEY, GOGO, GORK, GORM, GОРР, GORY, GOSH, GOTH, GOWK, GOWL, GOWN, GUAN, GUCK, GURL, GURN, GURU, GUSH, GYAL, GYRI, GYRO, QUAI, QUAY, QUEP, QUEY, RIAL, RICH, RICK, RICY, RIEL, RIEM, RISK, RISP, RIVO, ROAM, ROAN, ROCH, ROCK, RORY, ROSY, ROTI, ROTL, ROTO, RUBY, RUCK, RURP, RURU, RUSH, RUSK, RUTH, RYAL, SHAH, SHAM, SHAN, SHAY, SHRI, SIAL, SICH, SICK, SIDH, SIEN, SIGH, SIGN, SIRI, SITH, SIZY, SKEO, SKEP, SKRY, SLAM, SLAP, SLAY, SLEY, SNAP, SOAK, SOAP, SOCK, SORI, SORN, SOTH, SOWL, SOWM, SOWN, SOWP, SPAM, SPAN, SPAY, SPEK, SPRY, SUCH, SUCK, SUGH, SUSU, SYBO, SYEN, THAN, THEM, THEN, THEY, THRO, THRU, TICH, TICK, TIDY, TIROL, TIRO, TITI, TIVY, TOBY, TOCK, TOCO, TODY, TOEY, TOFU, TORI, TORN, TORO, TORY, TOSH, TOWN, TOWY, TUAN, TUCK, TURK, TURM, TURN, TUSH, TUSK, TUTU, TYRO, VIAL, VIRL, VIVO, VIZY, VLEI, VUGH, WHAM, WHAP, WHEN, WHEY, WICH, WICK, WIEL, WIRY, WISH, WISP, WITH, WOCK, WORK, WORM, WORN, WYCH, ZITI, ZOBO, ZORI

2. HILI, HILL, HOIK, HOKI, HOLK, HOLM, HOLP, HOLY, HOMO, HOMY, HONK, HOOK, HOON, HOOP, HULK, HULL, HUMP, HUNH, HUNK, HYMN, HYPO, IKON, ILLY, IMMY, IMPI, INKY, INLY, JILL, JIMP, JINK, JINN, JOHN, JOIN, JOKY, JOLL, JOMO, JOOK, JOUK, JUJU, JUKU, JUMP, JUNK, KILL, KILN, KILO, KILP, KINK, KINO, KIPP, KNOP, KOHL, KOJI, KOLO, KONK, KOOK, KOPH, KUKU, LILL, LILO, LILY, LIMN, LIMO, LIMP, LIMY, LINK, LINN, LINO, LINY, LION, LOIN, LOL, LOOK, LOOM, LOOP, LOUN, LOUP, LULL, LULU, LUMP, LUNK, LUNY, MILK, MILL, MILO, MINI, MINK, MINO, MINY, MOIL, MOJO, MOKI, MOKO, MOLL, MOLY, MOMI, MONK, MONO, MONY, MOOI, MOOK, MOOL, MOON, MOOP, MOPY, MOUP, MOYL, MUIL, MULL, MUMM, MUMP, MUMU, MUNI, MUON, NILL, NOIL, NOLL, NOLO, NOOK, NOON, NOOP, NOUL, NOUN, NOUP, NULL, OILY, OINK, OLIO, ONLY, OULK, OUPH, PHOH, PHON, PIKI, PILI, PILL, PILY, PIMP, PINK, PINY, PION, PIOY, PIPI, PIPY, PLIM, PLOP, PLOY, PLUM, POIL, POKY, POLK, POLL, POLO, POLY, POMO, POMP, PONK, PONY, POOH, POOK, POOL, POON, POOP, POUK, PUKU, PULI, PULK, PULL, PULP, PULU, PULY, PUMP, PUMY, PUNK, PUNY, PYIN, UMPH, UPON, YILL, YOLK, YOMP, YONI, YOOP, YOUK, YUKO, YUKY, YUMP

3. ABAC, ABAS, ABBA, ABBE, ABBS, ABED, ABET, ACED, ACER, ACES, ACRE, ACTA, ACTS, ADAW, ADDS, ADZE, AESC, AFAR, AGAR, AGAS, AGED, AGEE, AGER, AGES, ARAR, ARBA, ARBS, ARCS, ARDS, AREA, ARED, AREG, ARES, ARET, AREW, ARFS, ARSE, ARTS, ASAR, ASEA, ATAR, ATES, AVAS, AVER, AVES, AWED, AWEE, AWES, AXED, AXES, BAAS, BABA, BABE, BADE, BADS, BAFF, BAFT, BAGS, BARB, BARD, BARE, BARF, BARS, BASE, BASS, BAST, BATE, BATS, BATT, BAWD, BAWR, BEAD, BEAR, BEAT, BEDE, BEDS, BEEF, BEER, BEES, BEET, BEGS, BERE, BERG, BEST, BETA, BETE, BETS, BRAD, BRAE, BRAG, BRAS, BRAT, BRAW, BRED, BREE, BRER, BREW, BRRR, CABAB, CABS, CACA, CADE, CADS, CAFE, CAFF, CAGE, CARB, CARD, CARE, CARR, CART, CASA, CASE, CAST, CATE, CATS, CAVE, CAWS, CEAS, CECA, CEDE, CEES, CERE, CERT, CESS, CETE, CRAB, CRAG, CRAW, CRED, CREE, CREW, CZAR, DABS, DACE, DADA, DADS, DAES, DAFF, DAFT, DAGS, DARB, DARE, DARG, DART, DAST, DATA, DATE, DAWD, DAWS, DAWT, DAZE, DEAD, DEAF, DEAR, DEAW, DEBS, DEBT, DEED, DEER, DEES, DEET, DEEV, DEFT, DERE, DERV, DEVA, DEVS, DEWS, DRAB, DRAD, DRAG, DRAT, DRAW, DREE, DREG, DREW, EARD, EARS, EASE, EAST, EATS, EAWE, EBBS, ECAD, ECCE, EDGE, EFFS, EFTS, EGAD, EGER, EGGS, ERAS, ERED, ERES, ERGS, ERRS, ERST, ESES, ESSE, ESTS, ETAS, ETAT, EVER, EVES, EVET, EWER, EWES, EWTs, EXEC, EXES, FABS, FACE, FACT, FADE, FADS, FAFF, FAGS, FARD, FARE, FARS, FART, FAST, FATE, FATS, FAVA, FAVE, FAWS, FAZE, FEAR, FEAT, FEATS, FEED, FEER, FEES, FEET, FEGS, FERE, FESS, FEST, FETA, FETE, FETS, FETT, FRAB, FRAE, FRAG, FRAS, FRAT, FREE, FRET, GABS, GADE, GADS, GAED, GAES, GAFF, GAGA, GAGE, GAGS, GARB, GARE, GARS, GART, GAST, GATE, GATS, GAVE, GAWD, GAZE, GEAR, GEAT, GEDS, GEED, GEES, GEEE, GERE, GEST, GETA, GETS, GRAB, GRAD, GRAT, GRAV, GREE, GREW, GREX, QATS, RACA, RACE, RADE, RADs, RAFF, RAFT, RAGA, RAGE, RAGG, RAGS, RARE, RASE, RAST, RATA, RATE, RATS, RAVE, RAWs, RAZE, RAZZ, READ, REAR, REBS, RECS, REDD, REDE, REDS, REED, REEF, REES, REFT, REGS, REST, RETE, RETS, REVS, REWS, SABE, SABS, SACS, SADE, SAFE, SAGA, SAGE, SAGS, SARD, SARS, SASS, SATE, SAVE, SAWS, SCAB, SCAD, SCAG, SCAR, SCAT, SCAW, SEAR, SEAS, SEAT, SECS, SECT, SEED, SEER, SEES, SEGS, SERA, SERE, SERF, SERR, SERS, SESE, SESS, SETA, SETS, SETT, SEWS, SEXT, STAB, STAG, STAR, STAT, STAW, STED, STET, STEW, SWAB, SWAD, SWAG, SWAT, SWEE, TABS, TACE, TACT, TADS, TAED, TAES, TAGS, TARA, TARE, TARS, TART, TASS, TATE, TATS, TATT, TAVA, TAVS, TAWA, TAWT, TAXA, TEAD, TEAR, TEAS, TEAT, TEDS, TEED, TEER, TEES, TEFF, TEFS, TEGG, TEGS, TERF, TERR, TEST, TETE, TETS, TEWS, TEXT, TRAD, TRAT, TREE, TREF, TRES, TRET, TREW, TREZ, TSAR, TWAE, TWAS, TWAT, TWEE, TZAR, VACS, VADA, VAES, VARA, VARE, VARS, VASA, VASE, VAST, VATS, VAVS, VAWS, VEER, VEES, VEGA, VEGS, VERA, VERB, VERS, VERT, VEST, VETS, VEXT, WABS, WADA, WADD, WADE, WADS, WADT, WAES, WAFF, WAGE, WAGS, WAQF, WARD, WARE, WARS, WART, WASE, WAST, WATE, WATS, WATT, WAVE, WAWA, WAWE, WAWS, WEAR, WEBS, WEDS, WEED, WEER, WEES, WEET, WEFT, WERE, WERT, WEST, WETA, WETS, WEXE, ZAGS, ZARF, ZEAS, ZEDS, ZEES, ZEST, ZETA, ZEZE

Reminder

- Turn in your lab assignment to our Linux server.

- Make sure to only have one (1) .cpp file with the main() function in your working directory, otherwise your program will fail the grading script.
- Create a folder under your root directory, name the folder hw4 (case sensitive), copy all your .cpp and .h files to the folder (ArgumentManager.h is also needed)
- Only include the necessary files (.cpp and .h files) in your working directory in your final submission.
- To test your program, you may wish to copy the input files and answer files onto the server and run your program. Do not include any outputs files and after verifying that the code passes the tests, **delete any output*.txt files**.

Please reach out to myself or the TAs for any clarifications or typos.