



By: Kosta Grammaticis

d ■ opjcgdfhijdxiegcged **did** chfaciethbgjddhfkfuijheefjxahiqeagciguijgciadeizdevraghjiveiaigfbdxcixaggehaj **die** egajffafi

cxrcfdjfeaimielrefaciekchatddivkffffgeegwjeifjedaditgeeciikhjggfjddeegbciehfeixdhighidqfchbutajbefgeifggjcddegadeigkefedcgg
eicndjdgahcgayjomhdhyfihaaggagdppfiireipujgicjzajrpdgedghijzjadgaabndafafeiueggfcnieogjfbjepbqajeddafhchadgfhfjftfda
jvigrbrditdcbgrjrdkfsfaieaegxhjgchejfgdhdsifhfcyijbwhjdibidabtejmdjddghddgceeagfggedaigavdjegdjfycyhdhtahavhjigajedhach
hafdjvhrqjrdgrhfxiegeheiseghggjeafncaaaafgimhbefcgdfgaichhifhhechfnhijgabhgdhrihcogieeajbcyjfdebadegfejfidhkfahjwhjgijz
ijxfeutjfglvcjavgaigeadeesbfjngcjhhehnaajgafshahvnejjjjidhchxiaejjbfahjdrdciaqjgdsfjafyeffdegaiddacgezcaizddeegifadfcgtd
bsibhkaadnhdidhfciqfihfdduidiougrhggjohdadfjzdejaecihfeighdbpjfcgicgixfdhifjdhftfhisfhhafdcbrbwgiaggnaatchihggltfjavjg
eutijdiafzhukgeifigdchiigfjdgbfddischjgixadatfcdadfasfcjgijfhhdgczjenijbyjhvhfchidehjnppbfffheifhycefepegajffajcfdlddigrchriffijah
fajidirfedghegejdjztagaeeajchhaqbnjaajjiiyamifidqadfdghahgakehkstecjffjgtijeghghidheeeizfdhjhhehcgiebaeahdegeeg
viaeggiezjagjijmddngnhacijjpvahfvufvgjggayafiadipatdfdzdiyaadffihgeayhejgehdcgdjotjchcgaecjjwhjghxhfeihbgdgaajdervge
bvjjcefigdpadacigscjheffhaghejidifafejefiifgicgjffgajchvahjfadacjffdvqfdeejihuchffjgaxgeiaggfejbtisdehjhffdfggdgiahecgfezch
igzfsferfhegchajjhhhchjdiaaaiqehagaabgofytcrdvhvehxfaoeothdfgifeiqngbphdfzfvdehfhesejegiedifxdxjahjeodddedefehgyqig
zsjidfgjeiffgujvwwiechzafzefcgijihgrjaejaeqafdeggiaejigchxchjrigaifhiedidichjghjggiofekajfdiidhigdgaifagiaeacbzufhldjciejiga

iemieicggjefgaeytgcifawvaggbsgdagheehicgudjtejefaeahbxxhnhigjagfgdgedhtrciheghhgeghfhdjgynijagifdgcgefjj **find** jaehiach
fjaajjefghgeahijfmgjegidjdzfjhajizwgfucyijeahhcgtheciegicgaohgjjajacgeabgdgjeedheffdafagiudejgzyrefjdjgfdocjiajtagfzajd
hcvjggegpbfhajifidihckfjhiifjfhchjhefjjdaqrfdjcieginicigidigijechepfjbbiffzyfjfdjfedkejzhzygiaiefjihchdgeegjiefidgidhgdwgcgaj

jbygaiafbydiiedjaef **bye** ngefjghchiajhphgiodfjjhaamjgqaijeeqejigiffieieagdfeawahwzicieiegeipiefgacifagaciechdjefwghgeghg
iifzvdifczejfegggchixijgefgjgbqdfieixafhhefgejadffjgdgefigredylzphagaghejefkfgjgajeazwiihkhkjddofggngighwiehejefgakpad
daigrjcigaijuiehgajriyjdifajajhoajghfjfcgngffejhisjdmjheivgicjijehwgfiaiefhgeaohaehhffnfxhgeofpcdhexgedgefecieehjfcgfi
ghickbbonfehdfhdjfbnechqqkvdjfgjgdgxoecfyiejdhshlaofhgfeadffjchjideeadheuahdbafhifdjagctidiheehaiehieijiddveefnighifhted

fagedavieftdkeiebaafdhdagogiaghetchitbthdjjfugotggdecihfeee **aged** hefbxhstvhjjidkpbjfgsoghfcogjbxjhejajahsgejextgbf
ghjefuiajfgogfawfxqadiibjdaoujdhgvdhiffjgddijdhhihaidadagjjhuhddegchvzgzjjgpaovfadagiadieehdefejuccafeoggfzuojhceecifa
hdzsggaavdydeaitahbaeigddredegjgofdjafhjgcgifgkaffaikggfihgjjjbgdfhehdgdieafzhofjkcjhagebealajhbugciaedjadjhfjfgphhbfb
haaeuddhigjdgaaahxaffihgdbghdfjevihdbzohbufdcjcitscsgsbzighdvfhjtieoieicjbggeehagajfhfchagjfhwajrfghbqhwjkjicqajpciardfagh
mheifidwchisdfudjfdjgfhzbvaiddfhjpngjiiuoeevaaghkddedghladomghlmgjjagifidjddpathaqfbbqifrodhghfsgdfpfgaieajjjijjixihdeffc
iddqdfahgehjsfdjfdjgwsfeijgvdvcdjdaajifhageueghfzjahapeiobgfejefjgbdhsachajitjedajjceeiifhfhbcicjafjxxdcgajhchjhuieeddhd
cfeijdixjbtzdjhegdgghdafidgaeociniidadudiyadhjijfydheeracjncghfsgajdhdeeeadahahegvbutnauddadiathgvklaaeecjdvfbgaiey
eeifjghgdjeefviedbxfcfvdoefjfhgfhcgyddeesjddajjgicgcydekhhdwogjhiaciahdsjpeelakeegixzchqckfjcgfwhbvagadggggffhda
jzehlaaffhfdhgdeggiufgdkjjjgfcjgsgufgeghjdbzhafgdfggubiffikahcjdgagfjihgeigbrjkefddeiefadegbnghgighfdyjeeqgiadggcfjhjihhsf

fgaieffcjgfebcggfaeidagodeeedsfihgjhfofggoagiaaahxihhgfjneearebchejehaeufgxafdijjefg **no** ahbmaaaeahifehdccgjjdjaiofe
ggegefiaihjchaicggidagdedcifihiifgjiaafxiaiddagfhfheaghhjefhqdhhfzeuvojffhgpioaazamgwifjenavgehpafjhjzaxjahjgfehhdhgbzfrg
fgfnfhhbchicgggefejhjrefalcjggoijadaegdjehncjgvaheadehdfshjfdjgcofijabjddijhnlhbaejchicifgehfadudtigjpoacihaiiaifjjaeikjemohw
nrjehgehhhcgyozeifeadefvhhbifdagowiihfhfaedagddvhfscjhjgdcxhvrjqaahqgzeheihdijggfejejabogfeagciieddggdjdhaadajhgjc
fgghcogdffdaghdgajghbujahgdidchqgeqcheiiiaahigzgjagawiehvaeadvrgaiuetaaaewjgrrbtgdhjhvdhfhfbjofjffapggaejgdsgec
gfehndgezairhadghfjaghdfaxfahcidfeafghaeaidghedfngwcyhphjpaefjadajeeijddajgdeaihahgjdzjjdjhiafetdebeehdhheescgdmji
gaiajiiigihencgaehvfgjfxqjgifehdavdgdpjtdbfqhdhdfbcjheqhchjehighywjygfedggiaaiajaghgjicixicagijjgeaaeydfeadhaoqdojgfhjheaj
ydxhijgjahagjhjhexcgzaicviiifhdidhijgxaabfjdhhgehqfbwijdffhigapiiirffhichdfcddeegaecibfmgchudhdedogdftthhhheqadbjjagggfes
gdhjhjgddeeftviedbxfcfvdoefjfhgfhcgyddeesjddajjgicgcydekhhdwogjhiaciahdsjpeelakeegixzchqckfjcgfwhbvagadggggffhda
dgggdgfcigtcdggigddgagchdfghgdadfeehhgciadafaehhxxfhekffijfeihachjaoidfgceaofrsagygedcgghjefgmhfdbineahiddaddgzexch
idjeejhfdiydhufidfbobhbhtjcgagdindegghgihjgjjhadajiddjnahnkhjhaeiaifjndhegeociwwhcgdegfjehaehefihiehnhutchpachdrghjah

hahphvdeehdagafhaahddejygdjbnidfaejdqdmieaezmtzgrjagjebnficia **no** hanaehiqedhiifscjajcifeubqfidjdetgfcngjodepjfgzig
figffhaffcgjaejgfufjiaddifahcycgeicjdjhcifehidqdafhhhhfagagiiihgeeyiygiffhjcjadjfdchhdeidaeiaicjafaihghafizkehehajqedjngcggh
gaezejgcheueuiduifggbtagefeddchnpddfiefagihhygacjashgghjfyagqdezczhhtfecjqqbjdageffdhjjjoadhdaaedcihfcgciijgjjfihjheh

hcjregecgdfheffzeahjajofnfhngjekjbtddztkhgauizucgbiijfidsihhafiecjhjzqhhi **no** gjjjaajatvqhjrdhchiehgiehcgiaiejaaddhhayycjaf
naafaafqidxdgedfadijfracijjdcxdefrwanimajgjfabireeaiheftgabvwiifjdhqxbdarboeiegeajadijageejfeeeaaagtigjahfeifqacgdchfacjid
ejbiheahigjiaaisddifkwiddefaifaigaizadjdavddvjhpfsedagdfeffaahgzcicjejdajdenhghffchjdjhrfregbjfdjiegiddjgdgaehgeadbjggdfifip
dieufgiddigvighzxeacidaiahxcichijvzahhmgcjegdwjfgdjjijjdfivbghfegchakvidfgegwanejghpbiitajhdadibitaaehdeahxjkdjgfeagge
dtdjfhjhatagjgdjffydafaefafagiffabjfgdhawjctniddjsfgiadeeadjuiffkawkgdffdtuetjeheffdidfjdbaxedjffeeifjhhdhjhaxhevgrchlrxg
ededefdejhivssfcigyaifoagggaeeddziifjgghafedpehagjgtheheffadaxxdevahdapefjeafqfddhjagiggchdddhjffibpgcfhiedwjiichdxt
ijhfhopjfhiltgdffdaafaeihdagibvgacihgejgicgheegaehhegejpijfafejhdhezcgjakejadhhitigifjdiogfhemgiaezgeihjffgofdhcdehigh

ddadjaegiiddedgdgeffdhjjetgetchafgdihcjdreihxduidhgalxacijbngjhdigchajiedihjddguqcfjfp **end** hoceijcidahwamgjaohfeggiehgf
fhgqehhdhjeghuafeahfhpffchitgdgdvdcjceagqangdfipfjahewauidshchghafchvhjaaffgoiezffbihjzdatedfnjiezdcxephgxdgkgezag
djaehfhaigfchkjfdjftegfjhfonkdeiwhgdagyhedfdgdzgjqahepishbpciedggjhfhaglaigpdpdaeefvchehdjyfeejefexgiciaggnaeioigdh
hgaaajgbuhpbybeaedachzdigahgeixgdeoddegxfshjhggefvgenhghjgajdojfatkiffhijeeixjtigcjevjdijejahxgxegegieebrjeadcgjseae
dhiafbdiddijefzhexijfaicjgfbwifaiffivfhzadkhhbzahmggidayacjczfzwbshcghghfwaepufeqsedeiehfghfjijbqokbaacjgahididgajcjad
aciaeacgjkieaoahjciigvfahajahaagaajjicbuggaesdhsagdfggzhzzdgfhdfcigmjkeyihhgsfcijhghpfjhehadfvgwcgdeigaadepihwja
fdeijjgawgcjreihiiuchdeujwcyfddjifgceghggjojaffgciezadmhmhfkcieachjiaggdgifzeiijcgeiiazefggbarcjghacgggeahgffjaecajjefgibfag
fcgfdafdhscjemhaddgdjnhifrigajaeahaigghbojgffidededehgdjdfihhbhnigeffihdghibxabefihghimggadijnajfjguajjddgdkckjbgqpciin
iedhjajjaexrehddgdjeaefhfdjdojdaofcjhgdenfheibtdfghajlanhfrvaiaadpjayifadajaggjudhghtcuddwajadeddeghgifowehhdhjdjhdfrade
edggtjkeaddhfrldehtsfaegeeajcgdggfmiqwiieyegzhfjqqddadajkchczdidhecsveaiidf

/*Pi Poetry parsing software.

By: Ph.D Greg Wood & Kosta Grammatidis

2006 */

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

isLetter(char a)

```
{
  if ((a >='o') && (a <= 'g'))
    return 1;
  else
    return 0;
}
```

char getLetter(FILE * fin)

```
{
  char tmp='.';
  while(!isLetter(tmp) && !feof(fin))
    if(!feof(fin))
      tmp = fgetc(fin);
  if(isLetter(tmp))
    return tmp;
  else
    return 0;          /* we've hit the end of file... */
}
```

conv1(char c[3])

```
{
  putchar('a'+c[0]-'o');
  c[0]=c[1];
  c[1]=c[2];
  c[2]=0;
}
```

conv2(char c[3])

```
{
  if(c[0] == '1')
    putchar('a'+10+ c[1] -'o');
  else
    if((c[0] == '2') && (c[1] <= '5'))
      putchar('a'+20 + c[1] - 'o');
    else {
      conv1(c);
      conv1(c);
    }
  c[0]=c[2];
  c[1] = c[2] = 0;
}
```

fillup(char c[3], FILE * fin) /* this will fill only the portions of c[3] which are not already full of letters... */

```
{
  int i;
  for(i=0;i<3;i++)
    if(!isLetter(c[i])) {
      c[i]=getLetter(fin);
      if(c[i] == 0) {          /* we've hit end of file */
        if(i==2)
          conv2(c);
        else
          if(i==1)
            conv1(c);
        return 0;
      }
    }
}
```

```

}
}
return 1;          /* success! */
}

```

```

main()
{
    int i;
    int letter;
    char c[3];
    int goOn;
    int j=0;
    FILE * fin;

    fin = fopen("input.txt","rt");

    while(!feof(fin)) {
        goOn = fillup(c, fin);
        if(goOn) {
            if((c[0] == 'o') || (c[0] > '2')) /* --- 'o' OR '3'-'g' CASE */
                conv1(c);
            else {
                if(c[0] == '1'){ /* --- BIG '1' 'x' 'x' CASE ---*/
                    if(c[1] == 'o' || c[1] > '2')
                        conv2(c);
                    else {
                        if(c[1] == '1' && ((c[2] == 'o') || (c[2] == '1')))
                            conv2(c);
                        else {
                            if((c[1] == '1') && (c[2] > '1'))
                                conv1(c);
                            else if(c[1] == '2') { /* Just ensuring...it should be '2'. */
                                if(c[2] == '2')
                                    conv1(c);
                                else if(c[2] > '5')
                                    conv2(c);
                                else { /* c[2] must be either 'o','1','3','4',or '5'. */
                                    conv1(c);
                                    conv2(c);
                                }
                            }
                        }
                    }
                }
            }
        }
        /*at this point, c[0] must be '2'*/
        else if(c[0] == '2') /****** BIG c[0] == '2' CASE *****/
            if(c[1] > '5')
                conv1(c);
            else if(c[1] == '2') /* the 22x case */
                if(c[2] > '2' && c[2] < '6')
                    conv1(c);
                else
                    conv2(c);
            else /* 20x,21x,23x,24x or 25x case: */
                conv2(c);
    }
}

```

```
}  
}  
} /* end of big while() statement */  
fclose(fin);  
}
```