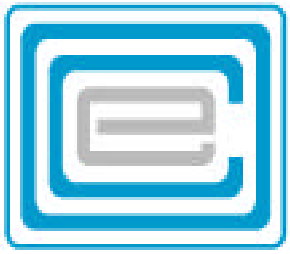


ECOO 2002
Programming Contest

Board wide Contest

to be written
after March 3rd

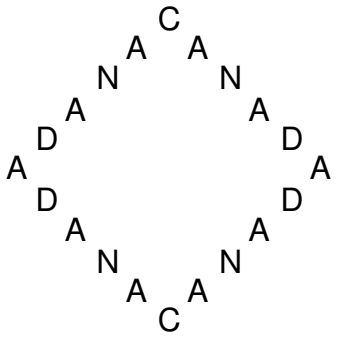
Problem 1: Diamonds

Write a program that will read from the file DATA11 (DATA12 for the second try) 5 words of length 3 to 10 letters and that will print out a diamond shape formed from each of these 5 words.
The program should pause between each diamond, and let you press a key to continue and present the next diamond.

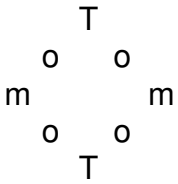
Sample input: (2 of 5 data only)

```
CANADA
Tom
```

Sample output:



Press any key to continue:



Press any key to continue:

Problem 2: Pairwise Cipher

In war time, messages sent by radio can be heard by the enemy. It is therefore important that the message is in a secret code, so that no one but friends, who have the **key**, can decode the message. The key is a secret phrase that only you and your friends know: It is used to scramble the alphabet. The scrambled alphabet is next used to find the letters to substitute in the message. There are therefore two steps in coding the message: first creating the scambled alphabet, and second, replacing the letters of the message, using the scrambled alphabet.

Suppose the key is **"Mon Oncle et ma tante"**
and the message is: **"Fish are birds without wings and birds are fish without fins"**

First, the alphabet is scrambled by means of a key as follows:
Line up the letters of the key together with the alphabet:
MONONCLEETMATANTEABCDEFGHIJKLMNOPQRSTUVWXYZ
Now pick out all the letters, one at a time, and if a letter occurs a second time, it is deleted: We now have the scrambled alphabet: **MONCLETABDFGHIJKPQRSUVWXYZ**

Next the message itself is prepared:

1. The letter "x" (and "X") is replaced by "ks"
2. The spaces are replaced by the letter X
3. All letters are capitalized.
4. All other characters are ignored and all accented letters are replaced by their regular counterparts

FISHXAREXBIRDSXWITHOUTXWINGSXANDXBIRDSXAREXFISHXWITHOUTXFINS

Encoding the message is done next

Group all the letters in the message by twos, adding the letter X at the end of the message if necessary to make a final pair.

FI SH XA RE XB IR DS XW IT HO UT XW IN GS XA ND XB IR DS XA RE XF IS HX WI TH OU TX FI NS

Now think of each letter as having a left and a right mate, according to the scrambled alphabet. The letter L has a left mate: (C) and a right mate: (E). Even the letter M has a right mate, (O) and a left mate: (Z). In the same way, Z has a left mate (Y) and a right mate (M).

Now substitute each letter of the pairs by the following rule,
using the left and right mates in: **MONCLETABDFGHIJKPQRSUVWXYZ**

Translating **FI**: take the right mate of **I (J)**, followed by the left mate of **F (D)**: **JD**
Translating **SH**: take the right mate of **H (I)**, followed by the left mate of **S (R)**: **IR**
Translating XA: take the right mate of A (B), followed by the left mate of X (W): BW
and so on, to give you these new pairs:

JD IR BW TQ DW SH UB XW AH NG AS XW CH UF BW FO DW SH UB BW TQ GW UH YG JV IE VM YE JD UO

And so the final message reads:

JDIRBWTQDWSHUBXWAHNGASXWCHUFBWFODWSHUBBWTQGWUHYGJVIEVMYEJDUO

You must write a program that will decode the messages that are received from your friends.
Your program must read 5 sets of data (a total of 10 lines) from DATA21 (DATA22 for the second try). Each set of data is made up of two lines: a key phrase and an encoded message. The lines are never larger than 80 characters. You must print out both the scrambled alphabet and the decoded message. You may expect only capital letters in the encoded message (line two of each pair) and no spaces. However, the first line in each pair may contain spaces, special characters and lower case letters.

Sample input: (only 2 of 5 sets of data)

nothing is new in this world
GCIGBVWWCVLHELRVHHTTHQRVOHEIBVAZCVLHELBVWWJVEHYTGEIOVNYOGCEZ
Zorro strikes again!
BZABVWRMYGYEKSALKWYGALTIDTYZRGYSRRMWBZYANEYZ

sample output:

NOTHIGSEWRLDABCFJKMPQUVXYZ
FISH ARE BIRDS WITHOUT WINGS AND BIRDS ARE FISH WITHOUT FINS

ZORSTIKEAGNBCDFHJLMPQUVWXY
ONCE UPON A TIME IN MEXICO NOT SO LONG AGO

Problem 3: Serial Number Dates

Every spreadsheet uses a special number to record dates and time.
For example, the number **35835.69412** represents 10 Feb 2002, 4:39 pm.
the integer part of the serial number, **35835**, stands for the number of days since 1 Jan 1904 and the fractional part, stands for the fraction of the current day that has passed: **.69412** =999/1440. In other words, 35835 days and 999 minutes have passed since midnight, 1 Jan 1904.

Note that 12 Nov 2002 24:00:00 o'clock is an invalid result,
and must be rewritten as: 13 Nov 2002, 0:00:00 o'clock.

Write a program that lets you read 5 serial numbers from the file DATA31 (DATA32 for the second try) and that will print out the exact date and time that is represented. As in the example below, you must include seconds.

Dates will vary between 1 Jan 1904 and 31 Dec 2099

Note that the leap years are every year that is divisible by 4, including the year 2000:
The serial number for 29 Feb 2000 is 35123

Sample input:

```
35835.69412
71582.86421
1.2345
4444.4444
1234.5678
```

Sample output:

```
year = 2002  month = 2  day = 10
time is: 16:39:32

year = 2099  month = 12  day = 25
time is: 20:44:28

year = 1904  month = 1  day = 2
time is:  5:37:41

year = 1916  month = 3  day = 2
time is: 10:39:57

year = 1907  month = 5  day = 19
time is: 13:37:38
```

Problem 4: Count Shapes

The input file contains a rectangular arrangement of dots and X's. The X's form shapes that are separated by space. The dots (periods) represent empty space which separate one shape from another. It is your task to count the number of shapes in the rectangle. For the purpose of defining a shape, please note that any given X belongs to the same shape as any other X that is its neighbour above, below, to its left and to its right. Any two X's on a diagonal are not connected. In the rectangle below there are 6 discrete shapes: two of these are touching diagonally and two shapes are positioned one inside the other.

```
.......xxxxxx..............................
.....xxxxxxxxxxx...........x................
....xxxxxxxxxxxxxx.........xxxxxxxxxx......
...xxxx...xxxxxxxxxxxx........xxxxxx.......
...xx....xxxxxxxxxxxxx...........x..........
........xxxxxxxxxxxxx........................
.......xxxxxxxxxxxxxx........................
...............xxxxxx........................
...............xxxx.......xxxxxxx...........
...........xxxx..........xxx..xxxx..........
...........xxxx..........xxx..xxx...........
........xxxxxxxxxx........xx..xx.............
........xxxxxxxxxx.........x.................
......xxxxxxxxxxxxxx.........................
.......xxxxxxxxxxxx..........................
............................................
............................................
............xxxxxxxxxxxxxxxxxxxxx...........
............xx..................xxx..........
............xx...xxxxxxxx......xxx..........
............xx...xxxxxxxx......xxx..........
............xx...xxxxxxxx......xxx..........
............xx...xxxxxxxx......xxx..........
............xx...xxxxxxxx......xxx..........
............xx..................xxx..........
............xxxxxxxxxxxxxxxxxxxxx...........
```

DATA41 (DATA42 for the second try) contains 5 sets of data. Each set of data will be made up of two lines containing the width or the rectangle (length or each line) and the second number the height of the rectangle to be examined (number of lines)

Sample input: (only 3 of a set of 5)

```
44
31
.........................................
.........................................
.......xxxxxx.............................
.....xxxxxxxxxxx...........x..............
....xxxxxxxxxxxxxx.........xxxxxxxxxx......
.......xxxxxxxxxxxxxx.....................
..............xxxxxx.....................
..............xxxx......xxxxxxx..........
...........xxxx.........xxx..xxxx.........
......xxxxxxxxxxxxxx......................
.......xxxxxxxxxxxx.......................
.........................................
............xxxxxxxxxxxxxxxxxxxxx.........
............xx.................xxx.........
............xx...xxxxxxxx......xxx.........
............xx...xxxxxxxx......xxx.........
............xx.................xxx.........
............xxxxxxxxxxxxxxxxxxxxx.........
5
3
.x.x.
x.x.x
.x.x.
50
5
.xxxxxxxx.x.............x.....................x
xx.......x.............xx.....................
.xx..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx.........xx...........x.............x.........
.xxxxx......xx.........xxx............x........x
```

Sample Output:

```
In rectangle #1 are 6 shapes

In rectangle #2 are 7 shapes

In rectangle #3 are 4 shapes
```

Problem 5: STACK print server

In a computer science lab, somewhere in Ontario, print jobs are sent to a print server. It is standard practice, that all jobs are placed in a queue, and they are printed in first-come-first-served order.

However, in this case, print jobs are placed on a stack, where last one in is first one out. When a job gets pushed onto the stack at the same time as the server "pops" a job off the stack, popping takes place before pushing.

The data file (DATA51 for the first try and DATA52 for the second) will contain 5 sets of data: The first line of each data set contains the number of lines that follow: Each of these lines contains three data, separated by a space: a 6-digit number representing the time the job is pushed onto the stack of the form: HHMMSS; a 3 digit number, representing the number of seconds it takes to print the job; and the name of the person who is sending the job (no more than 10 characters). The list is in the order in which jobs are requested.

It is your task to print out the statistics on the 5th and the last person printed:

Sample input: (3 of 5 sets only: no blank lines between sets)

```
12
110246 006 Alyssa
110332 189 Juan
110338 124 Jack
110422 074 Tanner
110608 130 Mary
110628 117 Seth
110915 003 Joshua
110947 007 Elizabeth
110957 023 Shelby
111048 133 Dalton
111123 095 Makayla
111205 193 Kayla
```

```
10
111316 065 Dylan
111357 189 David
111411 050 Miguel
111449 048 Cameron
111553 047 Monica
111647 173 Sydney
111813 172 Maria
111832 040 Patrick
111836 032 Melissa
111956 098 Ryan
```

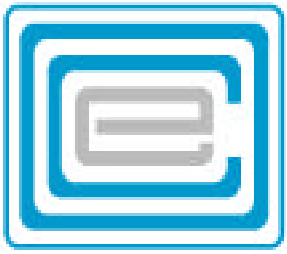
```
13
112030 172 Miranda
112045 062 Katherine
112229 003 Kaylee
112310 086 Alexander
112440 151 Jesse
112810 066 Anthony
112915 023 Stephanie
113109 170 Sierra
113208 115 Gabriel
113254 174 Mark
113352 078 Caroline
113540 171 Brooke
113552 178 Steven
```

Output:

```
job  5 completed at 11:11:11 for Shelby
job 12 completed at 11:21:40 for Jack

job  5 completed at 11:19:55 for David
job 10 completed at 11:28:30 for Sydney

job  5 completed at 11:28:24 for Katherine
job 13 completed at 11:45:55 for Gabriel
```

Detailed analysis for the first list:

```
11:02:46 Alyssa      arrives: busy till: 11:02:52
11:02:52 Alyssa      is finished: Job #1
11:03:32 Juan        arrives: busy till: 11:06:41
11:03:38 Jack        on the stack. Number of people waiting: 1
11:04:22 Tanner      on the stack. Number of people waiting: 2
11:06:08 Mary        on the stack. Number of people waiting: 3
11:06:28 Seth        on the stack. Number of people waiting: 4
11:06:41 Juan        is finished: Job #2
11:06:41 Seth        from the stack: busy till: 11:08:38
11:08:38 Seth        is finished: Job #3
11:08:38 Mary        from the stack: busy till: 11:10:48
11:09:15 Joshua      on the stack. Number of people waiting: 3
11:09:47 Elizabeth   on the stack. Number of people waiting: 4
11:09:57 Shelby      on the stack. Number of people waiting: 5
11:10:48 Mary        is finished: Job #4
11:10:48 Shelby      from the stack: busy till: 11:11:11
11:10:48 Dalton      on the stack. Number of people waiting: 5
11:11:11 Shelby      is finished: Job #5
11:11:11 Dalton      from the stack: busy till: 11:13:24
11:11:23 Makayla     on the stack. Number of people waiting: 5
11:12:05 Kayla       on the stack. Number of people waiting: 6
11:13:24 Dalton      is finished: Job #6
11:13:24 Kayla       from the stack: busy till: 11:16:37
11:16:37 Kayla       is finished: Job #7
11:16:37 Makayla     from the stack: busy till: 11:18:12
11:18:12 Makayla     is finished: Job #8
11:18:12 Elizabeth   from the stack: busy till: 11:18:19
11:18:19 Elizabeth   is finished: Job #9
11:18:19 Joshua      from the stack: busy till: 11:18:22
11:18:22 Joshua      is finished: Job #10
11:18:22 Tanner      from the stack: busy till: 11:19:36
11:19:36 Tanner      is finished: Job #11
11:19:36 Jack        from the stack: busy till: 11:21:40
11:21:40 Jack        is finished: Job #12
```



ECOO 2002

Eighteenth Annual Programming Contest

Regional

April 13, 2002

Problem 1: Halimate Coordinates

Captain John Archer encountered an intelligent species that has a unique way of describing 2 dimensions: The Halimates have no notion of negative numbers or indeed of ordered pairs to designate two dimensional concepts. Instead the Halimates have developed their own way of describing the locations of a two dimensional grid.

The centre of the grid is called 0. Along the main axis, the axis we would call the positive x-axis, lay the numbers 1,9, 25, 49, etc., the odd squares. All the natural numbers are arranged from there, along increasingly larger squares, in a counterclockwise direction as shown. And so, for example, the numbers 0, 30, 46, 81 correspond to the ordered pairs (0,0), (1,3), (3,-3) and (5,0).

Halimate Coordinates

		34	33	32	31	30	29	28		
		35	15	14	13	12	11	27		
		36	16	4	3	2	10	26		
		37	17	5	0	1	9	25	49	81
		38	18	6	7	8	24	48		
		39	19	20	21	22	23	47		
		40	41	42	43	44	45	46		

Human Coordinates

						1,3				
					0,0					5,0
								3,-3		

The text files DATA11 and DATA12 (no extension) will contain 5 numbers representing Halimate coordinates, which you must translate into Human coordinates as shown in the example below. Your output should appear on a cleared screen or window. The first text file will be used for the first try, the second file for the second try.

Sample Input:

```
121212
4600
123456
9997
10001
```

Sample Output:

```
121212   (-174,-107)
  4600   (-34,25)
123456   (97,176)
  9997   (-50,4)
 10001   (-50,0)
```

Problem 2: Snake Word Puzzle

In a certain variation of a word-find puzzle, words may be hidden not just in a horizontal, vertical or diagonal line. Word may change direction by moving left, right, up and down.

						O		
		S			O	N		
	A	N			I	T		
E	K	A	S	I	R	A	P	
	E							
		S			M	O		
		O	T				U	
	R					N	T	A
E							I	

The only direction words may not turn is in a diagonal direction . In the example on the left, the words SNAKE, ONTARIO and PARIS are valid words hidden in the puzzle, but STORE and MOUNTAIN contain letters that are connected diagonally and are therefore invalid words. Notice also, that ONTARIO occurs twice, and SNAKE occurs 4 times.

Your input file (DATA21 for the first try, DATA22 for the second) contains 5 words followed by 10 lines of 10 letters, representing a 10x10 word puzzle. Consider the upper left character to be (1,1), the upper right character to be (10,1) it is your task to locate the 5 words in the puzzle, by indicating the start of the word: If the word cannot be found, simply leave the starting ordered pair blank.
Note that none of the words have more than 8 letters.

Sample Input:

```
ONE
TWO
THREE
FOUR
FIVE
JDJKQIXWFK
UBLXVFBCUE
AENOWVTYAY
SKEFTHHVEW
YWBIRERUEL
YJIKHEFOUO
CJZMTBWQUA
UXGUZFWBNM
PEUMXIRYAJ
CWXGXCGSOO
```

Sample Output:

```
ONE is located at  (4,3) (4,3)
TWO is located at  (5,4)
THREE is located at  (5,7) (7,3)
FOUR is located at  (7,6)
FIVE is located at
```

Note: The two words **"ONE"** have a common start, but different endings.
There is no word **"FIVE"** in the 10x10 square.

Problem 3: Collect the Dots

You are given from 3 to 30 ordered pairs representing dots on graph paper, such that no set of 3 are collinear. You are then required to draw a polygon around all of them, so that no dot falls outside the polygon. Furthermore, the polygon must have a minimum amount of sides and its vertices must be a subset (possibly all) of the dots themselves.

The text file, DATA31 (DATA32 for the second try) contains 5 sets of data. The first line of each set contains a number, n, representing the total number of ordered pairs in the set. The next n lines contain the ordered pairs of the set, in no particular order. The coordinates of each dot are integers between 0 and 1000.

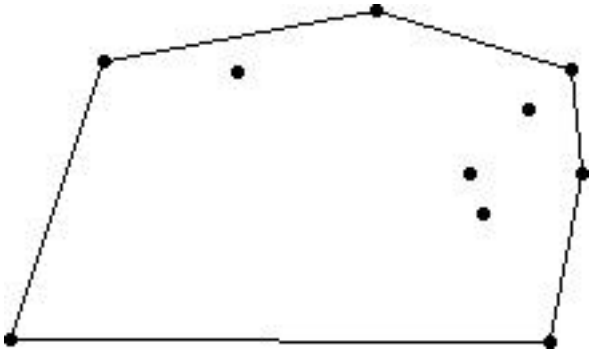
Your output should display all the dots of the set, as well as the polygon, in such a way that the polygon takes up most of the display screen or window. Each new display should appear after pressing any key.

Sample input:

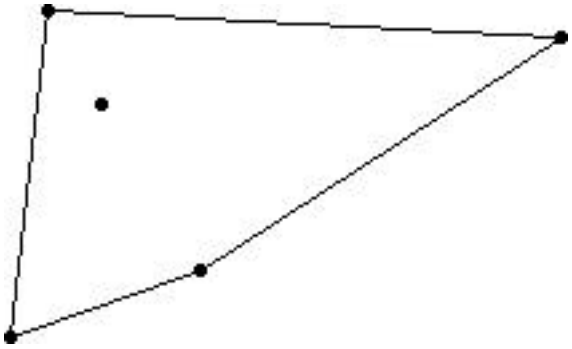
```
10
86 2
76 40
91 51
60 99
39 81
83 70
89 82
5 3
74 51
19 84
5
44 27
84 88
23 9
27 95
33 70
20
35 29
95 16
88 36
82 60
58 23
63 68
32 13
54 17
65 61
89 59
99 56
69 85
10 20
61 8
50 26
54 96
44 97
70 92
47 3
27 65
7
100 28
20 20
12 27
53 58
14 25
47 49
59 35
8
69 74
22 22
80 32
55 38
30 25
22 33
11 76
35 9
```

turn over

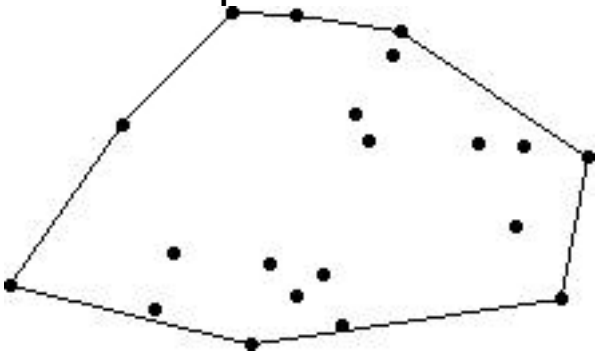
output 1



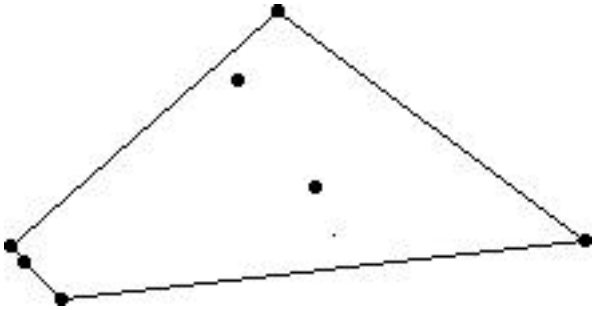
output 2



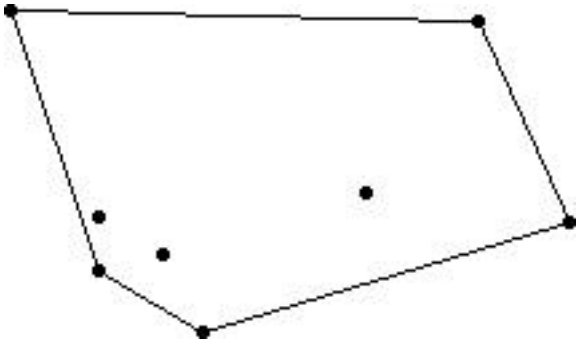
output 3



output 4



output 5



Problem 4: Code 29

A certain code cipher encodes messages as follows:

1. All letters will be capitalized
2. All characters that are not alphabetic will be deleted
3. All spaces will be replaced by the digits 1 or 2 in random order.
4. All letters are next replaced by their corresponding number:
   1=1, 2=2, A=3, B=4, ... , Z=28.
5. Next, each number is multiplied by its neighbour to the right, and replaced by the product
6. The last number is multiplied by a key: a secret number between 1 to 28 inclusive.
7. Since these numbers no longer can represent characters, the remainder is found after dividing by 29, which gives you numbers in the range 1,...,28 (the product will never produce a zero: this is a property of prime numbers such as 29)
8. Finally each number is replaced by its alphabetic value:
   1=1, 2=2, A=3, B=4, ... , Z=28.

Let us encode the phrase:
`For centuries the sundial was the predominant way of telling time.`

After instruction 4 we have:
8, 17, 20, 1, 5, 7, 16, 22, 23, 20, 11, 7, 21, 2, 22, 10, 7, 1, 21, 23, 16, 6, 11, 3, 14, 2, 25, 3, 21, 2, 22, 10, 7, 1, 18, 20, 7, 6, 17, 15, 11, 16, 3, 16, 22, 2, 25, 3, 27, 2, 17, 8, 1, 22, 7, 14, 14, 11, 16, 9, 1, 22, 11, 15, 7

None that the first space became 1. It could just as easily have become 2.

Since 8x17 = 136, which, when divided by 29 becomes 4 with remainder 20, we take the letter associated with 20, the letter R.
Next, 17*20 = 340. Quotient 11, remainder 21, letter S
Next, 20*1 = 20. Quotient 0, remainder 20, letter R
and so on:
We will use 19 for this occasion as the secret key. and so the last number will be multiplied by the number 19:
7*19 = 133. Quotient 4, remainder 17, letter O
and the secret code becomes:
RSRCDWBKWOQ2KMOJESQRGFBKZSOCKMOJEPJVKMUR2QQBMSOUWCRFTGITG2ZGTHRPO

Write a program that will take lines such as
`RSRCDWBKWOQ2KMOJESQRGFBKZSOCKMOJEPJVKMUR2QQBMSOUWCRFTGITG2ZGTHRPO`
and convert it to its original:
`FOR CENTURIES THE SUNDIAL WAS THE PREDOMINANT WAY OF TELLING TIME`

turn over

The text file DATA41 (DATA42 for the second try) contains 5 sets of 2 lines. The first line contains the key number (see step 6 above) and the second line contains a string of no more than 80 characters.
For each set find the original statement, as in the example below.

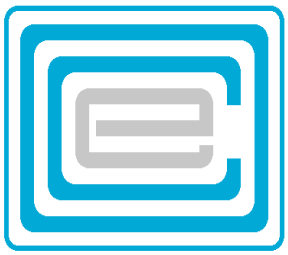
Note: In step 7, 8*17 = 136, which, when divided by 29 becomes 4 with remainder 20.
In shorthand notation, we say that 8*17 = 20 mod 29.
It is a property of prime numbers such as 29, that you can always go backwards, which of course you must do in order to decode the secret message:
Can you write a routine, that will find the 'x' in: x*17 = 20 mod 29 ?

Sample Input

```
19
RSRCDWBKWOQ2KMOJESQRGFBKZSOCKMOJEPJVKMUR2QQBMSOUWCRFTGITG2ZGTHRPO
10
RSCLYPHVKDWBMCRFAKTZHMQQPFBQOKZGAUEI2ACIBF2OKW
11
OJEWQSBJ1NFOJPNFHXPSCYUP2KNMSU1MOJEGDVRBXSQSBJNRSRLI2DGI2
28
C2QRG2JQHE
8
CFKWBPUYIZKMOJESZWSORJPUWCRFTOJLS1RBO
```

Sample Output

```
FOR CENTURIES THE SUNDIAL WAS THE PREDOMINANT WAY OF TELLING TIME
FORTY PERCENT OF ALL CANADIANS LIVE IN ONTARIO
THE WORD MATHEMATICS COMES FROM THE GREEK WORD FOR LESSON
GESUNDHEIT
SATURDAY IS THE SIXTH DAY OF THE WEEK
```



ECOO 2002

Eighteenth Annual Programming Contest

Finals

May 1, 2002

Problem 1: Fibonacci Rabbits

The Fibonacci numbers 1,1,2,3,5,8,... are a sequence of numbers where the next number is formed from the sum of the previous two. They represent the number of rabbit pairs there are from month to month, if each pair of rabbits were to generate one new pair each month, except in the first month of their lives.

in the following example, the numbers stand for the age of each rabbit pair

In month 0, there is one newborn pair, aged:	0	total number: 1
in month 1, the same pair:	1	total number: 1
in month 2, the first pair has offspring:	2,0	total number: 2
in month 3, again the first pair has offspring:	3,1,0	total number: 3
in month 4, the first two pairs have offspring:	4,2,1,0,0	total number: 5

and so on.

The assumption is, of course, that the rabbits live forever. What will happen to the numbers, when rabbits die after giving birth to 5 new pairs of rabbits (i.e. at age 6 months)? After giving birth to 10 pairs of rabbits (i.e. at age 11)?

Assume, that in each month, births happen first, then deaths, and finally you take count.

The input file will contain two numbers: the life expectancy, e, in terms of months, (less than 15) and the month, when you are to take count. Each pair of rabbits will have (e-1) rabbit pairs for offspring. Because the numbers increase very fast, the data will always result in a number of 8 digits or less.

DATA11 (no extension, DATA12 for the second try) will contain 10 numbers on 10 lines, representing 5 groups of 2 numbers. The first number in each group represents the life expectancy of each rabit pair, and the second number represents the month number, when you are to take count of how many rabbit pairs there are. Your task is to write a program that will print out the correct number of rabbit pairs there are, in the manner shown in the example below:

Sample Input

```
5
20
2
30
9
4
3
60
7
40
```

Sample Output:

```
 Life expectancy   number of months   number of rabbits
               5                 20               4420
               2                 30                  1
               9                  4                  5
               3                 60           20330163
               7                 40           89300197
```

Problem 2: Mountain Rescue

AABBC
ABBCC
AAAAD
DDDDD
BBEBB

To the left is a 5x5 array of letter representing a map of a mountain range. Each letter represents a square kilometer of the area. Different letters reprsent different plateaux: A being the lowest level and E the top of the mountain.

Someone is stuck on level E and your rescue party is on the upper left corner of the 5x5 grid. It is your task to find the quickest way up the mountain to level E to rescue the individual.

To travel from one area on the grid to another can be done in 4 different directions, North, South, East or West. It takes 20 mins to travel from one to the other at the same level but one hour to climb to the next level, 3 hours to climb 2 levels at the same time (e.g, from B up to D or from C down to A), 5 hours to climb 3 levels at one time and 7 hours to climb or descend 4 levels at a time.

You are on a rescue mission, and must reach the top of the mountain as quickly as possible. The quickes way will always be 10 moves or less.

Note: An exhaustive search will take 4^{10} = 1 048 576 steps, regardless of whether the field contains 25 squares or 2500 squares. Most of these paths will be such useless moves as going back and forth five times from (1,1) to (1,2). Clearly, for the sake of time, you must find a way to elliminate most of such useless paths.
When judging your output, you have a limit of 1 minute, in which your program must finish.

DATA21 (DATA22 for the second try) contains 25 lines of 5 letters, representing five 5x5 arrays. Write a program that will find the shortest path, and the lenght of time it is going to take to reach the summit. Print your result as shown in the example below:

Sample Input

```
ABBBB          AABBC          AEDDC          AABBC          AAAAA
BBAAA          ABBCC          ABBDC          CCCCC          ADDDA
AABBC          AAAAD          BBBBB          DCCEC          BDEDB
CCDDD          DDDDD          CCCCB          DDDDD          CDDDC
EDDDD          BBEBB          DDDBB          DDDDC          BBBBB
```

Sample Output:

```
it takes 440 minutes to take the path: ABBACDE
it takes 360 minutes to take the path: AABBCCDDDDE
it takes 340 minutes to take the path: AABBDE
it takes 300 minutes to take the path: AABCCDDE
it takes 280 minutes to take the path: AABCDDE
```

Note: Often the path is not unique: in which case, any correct path will do.

Problem 3: Genealogy

Data31 and Data32 are text files containing, among other things, a database with the records of several generations of a family.
Each record is saved as one line, where

1 characters 1-5 contains the individual's ID number. (a 5-digit number)
2 characters 6-29 contains the person's name
3 character 30 contains the person's sex (f or m, always in lower case.)
3 characters 31-35 contains the ID number of the individual's father
4 characters 36-40 contains the ID number of the individual's mother
5 characters 41-45 contains the year that the individual was born

All numbers are positive integers that have exactly 5 digits, with leading zeroes, and the name may contain several spaces:

Sample Input:

```
58
00055Amy Sue              H   f000540000001963
00039Anthony Dwayne       H   m000320003301961
00023Anthony Lee          C   m000000001301955
00037Barbara Jean         H   f000320003301956
00047Beverly Jane         O   f000000000001937
00002Blanche Ellen        S   f000000000001901
00036Brenda Sue           H   f000320003301954
00008Bruce Wayne          S   m000000000601970
00056Cathy Jo             H   f000540000001965
00013Charlotte Bernice    H   f000010000201924
00049Charlotte Rose       H   f000460004701960
00038Christopher Bryan    H   m000320003301959
00048Cynthia Jean         H   f000460004701958
00053Daniel James         A   m000000005001964
00017Dennis Duane         C   m000000001301947
00020Donald Ray           C   m000000001301951
00015Edward Duane         C   m000000001301945
00041Edward Max           H   m000000000001934
00016Ellen Bernice        C   f000000001301945
00004Frank A. (Jr.)       L   m000000000001920
00012Gary Wayne           L   m000040000301950
00003Geneva Elizabeth     H   f000010000201923
00033Gladys Lee           S   f000000000001933
00019Harald Louis         C   m000000001301950
00027Imogene Stover       H   f000010000201926
00058James Alan           H   m000540000001969
00031John David           L   m000000002701956
00001John James           H   m000000000001900
00032John William         H   m000010000201928
00018Joseph Edward        C   m000000001301948
00006Karen Kay            L   f000040000301944
00050Kathryn Ann          H   f000010000201937
00010Laurie Ellen         L   f000090000001969
00034Lawrence William     H   m000320003301951
00007Lisa Marie           S   f000000000601968
00021Lloyd Neal           C   m000000001301952
00035Loretta Jo           H   f000320003301953
00042Marcia Lynn          H   f000410004001954
00051Mark David           A   m000000005001959
00045Marlon Edward        H   m000410004001967
00024Martin Earl          C   m000000001301957
00044Marty Gene           H   m000410004001959
00040Mary Lou             H   f000010000201931
00026Matthew Charles      C   m000000001301960
00043Melodie Lou          H   f000410004001957
00014Pat                  C   m000000001301944
00030Paula Berdeen        L   f000000002701953
00054Phillip Lee          H   m000010000201943
00057Phillip Lee (II)     H   m000540000001967
00028Randall Gene         L   m000000002701952
00046Richard Daniel       H   m000010000201933
00005Robert Max           D   m000000000001923
00029Ryan Randall         L   m000280000001962
00022Samuel Lee           C   m000000001301954
00009Thomas Dean          L   m000040000301948
00011Thomas Lee           L   m000090000001970
00052Timothy Donald       A   m000000005001960
00025Walter Vincent       C   m000000001301959
32
1
29
20
3
```

The first line of the database is an integer, which denotes the number of records that follow (less than 500). Any ID that is cannot be located in the current database will designate an unknown person. and so some individual may have parents who are "unknown".
Also, of course, individuals, who are childless have an "unknown" spouse. Even if one has an oldest child, that child's other parent may be unknown.
You may assume that there are always less than 20 children in any given family.

turn over

After the database are 5 special integers, representing the ID numbers of individuals for which you must print out the following information: (**exactly as in the sample output below**)

1 both the father and mother (if known, otherwise print "unknown")
2 all the children (unless no children can be found)
3 the spouse (Here you must assume that the other parent of one's oldest child is the legal spouse of the individual). If there is no spouse, do not print this line.

Each of the 5 sets of data must appear on a cleared screen/window, and the program must be waiting for any character to be pressed before displaying the next set of data.

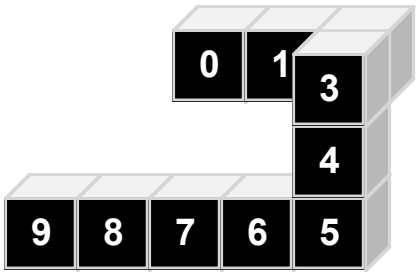
Sample output:

```
subject:  John William        H    born: 1928     male
-------------------------------------------
father:   John James          H    born: 1900
mother:   Blanche Ellen       S    born: 1901
wife:     Gladys Lee          S    born: 1933
children :
          Lawrence William    H    born: 1951     male
          Loretta Jo          H    born: 1953     female
          Brenda Sue          H    born: 1954     female
          Barbara Jean        H    born: 1956     female
          Christopher Bryan   H    born: 1959     male
          Anthony Dwayne      H    born: 1961     male
Press any key to continue..

subject:  John James          H    born: 1900     male
-------------------------------------------
father:   unknown                  born: unknown
mother:   unknown                  born: unknown
wife:     Blanche Ellen       S    born: 1901
children :
          Geneva Elizabeth    H    born: 1923     female
          Charlotte Bernice   H    born: 1924     female
          Imogene Stover      H    born: 1926     female
          John William        H    born: 1928     male
          Mary Lou            H    born: 1931     female
          Richard Daniel      H    born: 1933     male
          Kathryn Ann         H    born: 1937     female
          Phillip Lee         H    born: 1943     male
Press any key to continue..

subject:  Ryan Randall        L    born: 1962     male
-------------------------------------------
father:   Randall Gene        L    born: 1952
mother:   unknown                  born: unknown
no children
Press any key to continue..

subject:  Donald Ray          C    born: 1951     male
-------------------------------------------
father:   unknown                  born: unknown
mother:   Charlotte Bernice   H    born: 1924
no children
Press any key to continue..

subject:  Geneva Elizabeth    H    born: 1923     female
-------------------------------------------
father:   John James          H    born: 1900
mother:   Blanche Ellen       S    born: 1901
husband:  Frank A. (Jr.)      L    born: 1920
children :
          Karen Kay           L    born: 1944     female
          Thomas Dean         L    born: 1948     male
          Gary Wayne          L    born: 1950     male
press any key to continue..
```

Problem 4: Tunneling

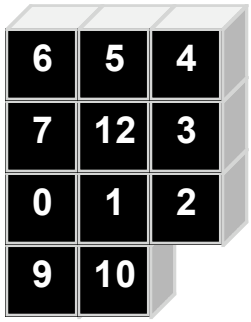
An interesting mining robot of the 24th century is the tunneling machine. It will remove a cubic meter of dirt from where it is operating and beam it to its destination. It will then reinforce the sides of the cube with special walls on all 6 sides, except where there is access to previously mined sections. The machine will in fact remove existing walls, if the newly excavated cubic space is adjacent to an older space.

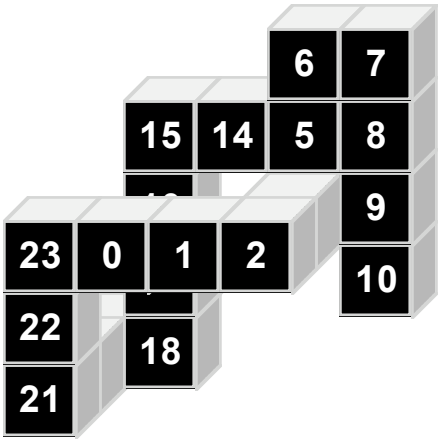
The machine can move in any one of 6 directions: East=1, Up=2, South=3, West=4, Down=5 and North=6. The robot must keep track of how many walls it has installed. The following are some examples:



In the figure to the left, the robot received the following 9 instructions: 1, 1, 3, 5, 5, 4, 4, 4, 4
For convenience, the cubes are numbered in the order in which they have been excavated, with 0 the original reference cube, and therefore the robot started out with the 6 walls of cube #0
Next it excavated cube 1, removing 1 wall of cube 0, adding 5 walls of cube 1
Then it excavated cube 2, removing 1 wall of cube 1, adding 5 walls of cube 2, etc.
After excavating the 9 cubes, the total number of walls is 42

In the figure to the right, the robot received the following 12 instructions: 1, 1, 2, 2, 4, 4, 5, 5, 5, 1, 2, 2.
Again, the cubes are numbered in the order in which they have been excavated, with 0 the original reference cube, and therefore the robot started out with the 6 walls of cube #0
When cube 7 was excavated, not only was one wall removed from cube 6, one wall also was removed from cube 0, and 4 walls added for cube 7
Cube 8 occupies the same space as cube 0, and therefore no walls were removed or added.
With cube 12, one wall was removed from cube 11=cube 1, one wall from each of cubes 3, 5, and 7, and 2 walls were added from cube 12.
There are a total of 36 walls.





In the figure to the left, the robot received the following 23 instructions: 1, 1, 6, 6, 2, 2, 1, 5, 5, 5, 2, 2, 4, 4, 4, 5, 5, 5, 4, 3, 3, 2, 2

Notice that after cube 10, the robot backs up and retraces former cubes: 11=10, 12=8, 13=5.

This figure has a total of 80 walls.

It is your task to write a program that will keep track of the number of walls that are in the tunnel that the robot has excavated.

DATA41 (DATA42 for the second try) contains 5 sets of data: The first line of each set contains the number n of instructions for the robot. The next n lines contain the moves the robot must make: numbers from 1 to 6. The 5 sets of data form one continuous set of numbers not broken up by any blank lines.

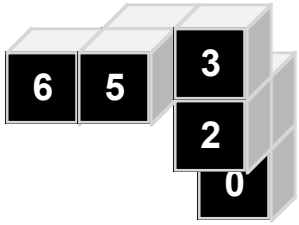
please turn

There are never more than 99 moves, and no square will be excavated that is more than 10 meters removed from the starting position in any of the 6 directions.

Sample Input:

```
5       4       6
1       4       2
1       4       2
1       12      1
1       1       5
1       1       5
6       2       5
2       2       2
3       4       2
2       4       4
4       5       4
3       5       4
4       5       5
9       1       5
1       2       5
1       2       4
3       23      3
2       1       3
2       1       2
4       6       2
```

For your convenience, here are the pictures of the first two tunnels. The last three appear on the previous page.



Sample Output

```
after 5 moves, the tunnel has 26 walls.
after 6 moves, the tunnel has 30 walls.
after 9 moves, the tunnel has 42 walls.
after 12 moves, the tunnel has 36 walls.
after 23 moves, the tunnel has 80 walls.
```

Score Sheet

ECOO Programming Contest

April 2002

Team (School) Name: _____

1

	Time handed in	Judge's initials	Score
11			
12			

2

	Time handed in	Judge's initials	Score
21			
22			

3

	Time handed in	Judge's initials	Score
31			
32			

4

	Time handed in	Judge's initials	Score
41			
42			