C-Programming Final

Department of Computer Science and Engineering, National Sun Yat-sen University.

2019/01/03

Note:

|  |
| --- |
| **(1) 程式碼全部撰寫在同一個** .c**檔中。**  **(2) 請存成** .c**檔，檔名用自己的學號命名。**  **(3) 請勿更改給定的main function，若更動main function導致程式無法執行，後果自負。**  **(4) 輸出格式須與範例〝一模一樣〞，勿多寫跟少寫，否則會斟酌扣分。**  **(5) 助教批改時會使用新安裝的Devcpp進行編譯，若在考試時對編譯器進行更動或使用非預設之標頭檔，後果自負。**  **(6) 對1題20分，對2題40分，對3題60分，對4題80分，全對100分。** |

1. 請設計一程式印出你的學號及姓名

Please design a program to print your student ID and name.

|  |
| --- |
| Name: 王小明  Student ID: M073040000  請按任意鍵繼續 . . . |

1. 3025這個4位數相當特別。如果你把他從中間切開你會得到2個長度相同的數30,25。而且此2數和的平方恰等於原來的數，我們稱這種數為quirksome number。

(30+25)2=3025

這個問題是：給你N位數（N＝2,4,6,8），要找出符合以上性質的所有數。  
For example：4位數就是從0000到9999。注意：數字前方的0也要算在內。也就是說0001等於 (00+01)2，是一個4位數的quirksome number。

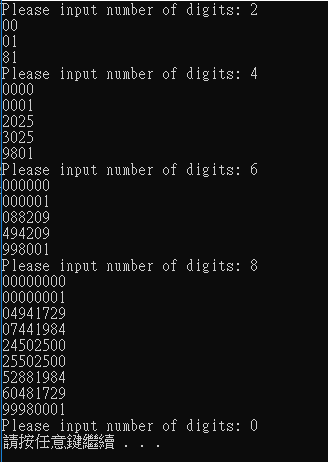
Hint: 有可能是2進位、4進位、6進位跟8進位，所以我就分別判斷(如果2進位就把它拆成兩個1進位去計算；4進位拆成兩個2進位去算...以此類推)

輸入說明

輸入可能包含了好幾列測試資料，每一列只有一個數字N（ N=2,4,6,8），代表要求的是幾位數的quirksome number。N=0實結束程式

輸出說明

對每一個輸入N，你應該要輸出所有N位數的quirksome number，每個一行，由小到大排列。注意：數字前方的0不可忽略，例如2位數中的quirksome number 01不可只輸出1。



The number 3025 has a remarkable quirk: if you split its decimal representation in two strings of equal length (30 and 25) and square the sum of the numbers so obtained, you obtain the original number:

(30+25)2=3025

The problem is to determine all numbers with this property having a given even number of digits.

For example, 4-digit numbers run from 0000 to 9999. Note that leading zeroes should be taken into account. This means that 0001 which is equal to (00 + 01)2 is a quirksome number of 4 digits. The number of digits may be 2,4,6 or 8. Although maxint is only 32767 and numbers of eight digits are asked for, a well-versed programmer can keep his numbers in the range of the integers. However efficiency should be given a thought.

Hint: Input maybe 2 digits, 4 digits….( if 2 digits, it will be split into two 1 digits to calculate; 4 digits split into two 2 digits to calculate.)

Input

Input maybe contain several lines of testing data, each line has only one number N(N=2,4,6,8), means number of digits of this quirksome number. When N=0, exit the program.

Output

For each input N, you should output all the n-digit quirksome numbers, one for each row, from small to large. Note: 0 in front of the number cannot be ignored. For example, quirksome number 01 in 2 digits cannot be output only 1.

3.

質數就是一個數字只有兩個因數：1和自己，例如：1,2,3,5,17,101和10007都是質數。

在這問題中你需要讀入一組文字，每個字是由 a-z 和 A-Z 組成，每個字母都有他的值，字母 a 值 1，字母 b 值 2 ... 到字母 z 值 26，同樣的字母 A 值 27，B 值 28， Z 值 52。

你應該要寫個程式來檢查這組字是不是prime word，如果這組文字的字母值總和為質數的話，他就是 prime word。

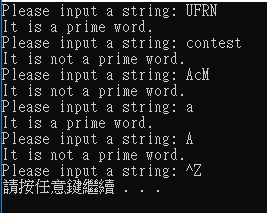
Hint: 如果n沒有任何小於等於http://calculus.nctu.edu.tw/upload/calculus_web/maple/Site/carnival/number/01.files/image006.gif的因數，那麼必定是質數。

輸入說明

輸入資料可以不斷輸入直到EOF，每組一列，且有 L （1 <= L <= 20）個字母。

輸出說明

對每一組字，如果它是 prime word 的話，輸出 "It is a prime word."。否則，請輸出 "It is not a prime word."。



A prime number is a number that has only two divisors: itself and the number one. Examples of prime numbers are: 1, 2, 3, 5, 17, 101 and 10007.

In this problem you should read a set of words, each word is composed only by letters in the range a-z and A-Z. Each letter has a specific value, the letter a is worth 1, letter b is worth 2 and so on until letter z that is worth 26. In the same way, letter A is worth 27, letter B is worth 28 and letter Z is worth 52.

You should write a program to determine if a word is a prime word or not. A word is a prime word if the sum of its letters is a prime number.

Hint: If n does not have any factor less than or equal tohttp://calculus.nctu.edu.tw/upload/calculus_web/maple/Site/carnival/number/01.files/image006.gif, then it must be a prime number.

Input

Input data can be entered continuously until EOF, one column per group, with L (1 <= L <= 20) letters.

Output

For each word you should print: ‘It is a prime word.’, if the sum of the letters of the word is a prime number, otherwise you should print: ‘It is not a prime word.’.

4.在早期人工智慧的領域中常常會用到機器人，在這個問題中有一支機器手臂接受指令來搬動積木，而你的任務就是輸出最後積木的情形。

一開始在一平坦的桌面上有n塊積木（編號從0到n-1）0號積木放在0號位置上，1號積木放在1號位置上，依此類推，如下圖。

https://zerojudge.tw/ShowImage?id=853

機器手臂有以下幾種合法搬積木的方式（a和b是積木的編號）：

• move a onto b

在將a搬到b上之前，先將a和b上的積木放回原來的位置（例如：1就放回1的最開始位罝）

• move a over b

在將a搬到b所在的那堆積木之上之前，先將a上的積木放回原來的位罝（b所在的那堆積木不動）

• pile a onto b

將a本身和其上的積木一起放到b上，在搬之前b上方的積木放回原位

• pile a over b

將a本身和其上的積木一起搬到到b所在的那堆積木之上

• quit

動作結束

前四個動作中若a=b，或者a, b在同一堆積木中，那麼這樣的動作算是不合法的。所有不合法的動作應該被忽略，也就是對各積木均無改變。

輸入說明

輸入含有多組測試資料，每組測試資料的第一列有一個正整數n（0 < n < 25），代表積木的數目（編號從0到n-1）。接下來為機器手臂的動作，每個動作一列。如果此動作為 quit，代表此組測試資料輸入結束。你可以假設所有的動作都符合上述的樣式。請參考Sample Input。

輸出說明

每組測試資料輸出桌面上各位置積木的情形（每個位置一列，也就是共有n列），格式請參考Sample Output。

範例輸入

10

move 9 onto 1

move 8 over 1

move 7 over 1

move 6 over 1

pile 8 over 6

pile 8 over 5

move 2 over 1

move 4 over 9

quit

範例輸出

0: 0

1: 1 9 2 4

2:

3: 3

4:

5: 5 8 7 6

6:

7:

8:

9:

Many areas of Computer Science use simple, abstract domains for both analytical and empirical studies.

For example, an early AI study of planning and robotics (STRIPS) used a block world in which a robot arm performed tasks involving the manipulation of blocks.

In this problem you will model a simple block world under certain rules and constraints. Rather than determine how to achieve a specified state, you will “program” a robotic arm to respond to a limited set of commands.

The problem is to parse a series of commands that instruct a robot arm in how to manipulate blocks that lie on a flat table. Initially there are n blocks on the table (numbered from 0 to n − 1) with block bi adjacent to block bi+1 for all 0 ≤ i < n − 1 as shown in the diagram below:

https://zerojudge.tw/ShowImage?id=853

The valid commands for the robot arm that manipulates blocks are:

• move a onto b

where a and b are block numbers, puts block a onto block b after returning any blocks that are stacked on top of blocks a and b to their initial positions.

• move a over b

where a and b are block numbers, puts block a onto the top of the stack containing block b, after returning any blocks that are stacked on top of block a to their initial positions.

• pile a onto b

where a and b are block numbers, moves the pile of blocks consisting of block a, and any blocks that are stacked above block a, onto block b. All blocks on top of block b are moved to their initial positions prior to the pile taking place. The blocks stacked above block a retain their order when moved.

• pile a over b

where a and b are block numbers, puts the pile of blocks consisting of block a, and any blocks that are stacked above block a, onto the top of the stack containing block b. The blocks stacked above block a retain their original order when moved.

• quit

terminates manipulations in the block world.

Any command in which a = b or in which a and b are in the same stack of blocks is an illegal command. All illegal commands should be ignored and should have no affect on the configuration of blocks.

Input

The input begins with an integer n on a line by itself representing the number of blocks in the block world. You may assume that 0 < n < 25.

The number of blocks is followed by a sequence of block commands, one command per line. Your program should process all commands until the quit command is encountered.

You may assume that all commands will be of the form specified above. There will be no syntactically incorrect commands.

Output

The output should consist of the final state of the blocks world. Each original block position numbered i (0 ≤ i < n where n is the number of blocks) should appear followed immediately by a colon. If there is at least a block on it, the colon must be followed by one space, followed by a list of blocks that appear stacked in that position with each block number separated from other block numbers by a space. Don’t put any trailing spaces on a line.

There should be one line of output for each block position (i.e., n lines of output where n is the integer on the first line of input).

Sample Input

10

move 9 onto 1

move 8 over 1

move 7 over 1

move 6 over 1

pile 8 over 6

pile 8 over 5

move 2 over 1

move 4 over 9

quit

Sample Output

0: 0

1: 1 9 2 4

2:

3: 3

4:

5: 5 8 7 6

6:

7:

8:

9:

5. 在本題中，題目會先給你一個包含小括號()及中括號[]的字串。當字串符合下列條件時我們稱他為正確的運算式：

1.該字串為一個空字串

2.如果Ａ和Ｂ都為正確的運算式，則ＡＢ也為正確的運算式，

3.如果Ａ為正確的運算式，則(A)及[A]都為正確的運算式。

現在，請你寫一支程式可以讀入這類字串並檢查它們是否為正確的運算式。字串的最大長度為128個字元。

輸入說明

輸入的第一列為正整數n，代表接下來有n列待測資料。

輸出說明

檢查每列待測資料，如果正確輸出Yes，否則輸出No。

範例輸入

3

([])

(([()])))

([()[]()])()

範例輸出

Yes

No

Yes

You are given a string consisting of parentheses () and []. A string of this type is said to be correct:

(a) if it is the empty string

(b) if A and B are correct, AB is correct,

(c) if A is correct, (A) and [A] is correct.

Write a program that takes a sequence of strings of this type and check their correctness. Your program can assume that the maximum string length is 128.

Input

The file contains a positive integer n and a sequence of n strings of parentheses ‘()’ and ‘[]’, one string

a line.

Output

A sequence of ‘Yes’ or ‘No’ on the output file.

Sample Input

3

([])

(([()])))

([()[]()])()

Sample Output

Yes

No

Yes

Main Function(請複製 Please Copy)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

void Question1();

void Question2();

void Question3();

void Question4();

void Question5();

int main(){

srand(time(NULL));

int i;

while(1){

printf("Please enter the number of question(0 for exit): ");

scanf(" %d",&i);

if (i==0) break;

switch(i){

case 1:

Question1();

break;

case 2:

Question2();

break;

case 3:

Question3();

break;

case 4:

Question4();

break;

case 5:

Question5();

break;

default:

printf("Undefined number!!");

}

}

}