Shool ID:
Name:

ACM on-line Judge 程式上傳範例

ACM problem Set 網址 HTTP://ACM.UVA.ES/PROBLEMSET/

Submit your Code 上傳程式碼

Register 註冊

ID:
PASSWORD:

Case 1: 以特定值當作輸入結尾。

例如：ACM Q579 :ClockHands

Input

The input is a list of times in the form *H*:*M*,… **The input is terminated with the time 0:00**. Note that *H* may be represented with 1 or 2 digits (for 1-9 or 10-12, respectively); *M* is always represented with 2 digits…

Output

The output displays the smallest positive angle in degrees between the hands for each time. The answer should between 0 degrees and 180 degrees for all input times. Display each angle on a line by itself in the same order as the input. The output should be rounded to the nearest 1/1000, i.e., **three places after the decimal point should be printed.**

輸入輸出	一般程式只處理一筆的寫法	ACM 的寫法

Sample Input
12:00
9:00
8:10
0:00
 	
Sample Output
0.000
90.000
175.000
	#include <stdio.h>
	#include <stdlib.h>
	
	int main(void)
	{
	int H,M,m1,m2;
	float o,i,k;
	
	printf("What time is it?(x:xx)(12hr)");
	scanf("%d:%d",&H,&M);
	
	m1=M/10;
	m2=M%10;
	……
	
	system("PAUSE");
	return 0;
	
	}
		//ACM Q579
		//ID-xxxxxx
		#include <stdio.h>
		#include <stdlib.h>
		
		int main(void)
		{
		int H,M,m1,m2;
		float o,i,k;
		
		~~printf("What time is it?(x:xx)(12hr)");~~
		
		while (scanf("%d:%d",&H,&M) == 2)
		{
		if (H==0 && M==0)
		break;
		
		m1=M/10;
		m2=M%10;
		……
		
		}
		
		~~system("PAUSE");~~
		return 0;
		
		}
		
		這個不用寫
		這個不用寫

Case 2: 未給任意值當作輸入值結尾。

例如：**ACM Q100: The 3n + 1 problem**

The Input

The input will consist of a series of pairs of integers *i* and *j*, one pair of integers per line. … You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including *i* and *j*. You can assume that no opperation overflows a 32-bit integer.

The Output

For each pair of input integers *i* and *j* you should output *i*, *j*, and the maximum cycle length for integers between and including *i* and *j*. These three numbers **should be separated by at least one space** with all three numbers on one line and with one line of output for each line of input. The integers *i* and *j* must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

輸入輸出	一般程式只處理一筆的寫法	ACM 的寫法

Sample Input
1 10
10 1
100 200
201 210
900 1000
 	
Sample Output
1 10 20
10 1 20
100 200 125
201 210 89
900 1000
174
	#include <stdio.h>
	#include <stdlib.h>
	
	int main(void)
	{
	int i,j;
	int length;
	
	printf("Input 2 數，空白隔開:)");
	scanf("%d %d",&i,&j);
	
	……
	
	system("PAUSE");
	return 0;
	
	}
		//ACM Q100
		//ID-xxxxxx
		#include <stdio.h>
		#include <stdlib.h>
		
		int main(void)
		{
		int H,M,m1,m2;
		float o,i,k;
		
		~~printf("Input 2 數，空白隔開:)");~~
		while (1)
		{
		scanf("%d %d",&i,&j);
		……
		
		}
		
		~~system("PAUSE");~~
		return 0;
		
		}

這個不用寫

這個不用寫

Case 3: 要求以 EOF 為輸入結束值。

例如：**ACM Q10432: Polygon Inside A Circle**

Time limit: 1 second Memory: 16 MB

The Problem

Consider a polygon of equal sides inside a circle as shown in the figure below.

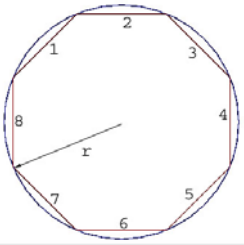

Figure: The regular polygon inside a circle

Given the radius of the circle and number of sides. You have to find the area of the polygon.

The Input

In each line there will be two numbers indicating the radius **`r'** (0<**r**<20000) and the number of sides of the polygon **`n'** (2<**n**<20000) respectively. **Input is terminated by `EOF'**.

The Output

Output the area in each line. The number must be rounded to the **third digit** after the decimal point.

輸入輸出	一般程式只處理一筆的寫法	ACM 的寫法

Sample Input
2 2000
10 3000
Sample Output
12.566
314.159
	#include <stdio.h>
	#include <stdlib.h>
	#define PI 3.141592653589
	
	int main(void)
	{
	double r, n;
	
	printf("Input radius and side, 空白隔開:)");
	scanf("%lf %lf",&r,&n);
	
	……
	
	system("PAUSE");
	return 0;
	}
		//ACM Q10432
		//ID-xxxxxx
		#include <stdio.h>
		#include <stdlib.h>
		
		int main(void)
		{
		double r, n;
		
		~~printf("Input radius and side, 空白隔開:)");~~
		while (scanf("%lf %lf",&r,&n) != EOF)
		{
		scanf("%d %d",&i,&j);
		……
		
		}
		
		~~system("PAUSE");~~
		return 0;
		
		}

這個不用寫

這個不用寫

台大 NPSC 上傳程式範例

npsc 網址： http://contest.cc.ntu.edu.tw/npsc2004/

題目(1) 羅馬數字

問題

羅馬數字經常用於書本章節及頁碼的編號。在這一題工作是讀入兩個正整數，然後輸出兩個數字差的絕對值。所有的數字都必須以羅馬數字來表示。而連續四個相同符號出現時，必須用減法規則來化簡之。

輸入檔說明

每個輸入檔中會有一個或以上的測試資料。每一行由兩個數字組成一筆測試資料，且所有數字將會小於 4,000。檔案最後會以符號 # 表示結束。

輸出檔說明

每筆測試資料的答案必須輸出到檔案中，並且換行。如果答案爲零，則須輸出字串 ZERO。

輸入輸出	一般程式只處理一筆的寫法	NPSC 的寫法

ROMAN.IN
I I
MM II
#
 	
ROMAN.OUT
ZERO
MCMXCVIII
	#include<stdio.h>
	#include<stdlib.h>
	#include<math.h>
	
	int main(void)
	{
	char r[14],l[14],w;
	int i,n,t,k[14],m[14],p,z,s;
	float b;
	
	printf("請輸入羅馬數字 x2 來計算數差(用空格隔開):");
	
	scanf("%s %s",r,l);
	
	……
	printf("%s %s",k,m);
	system("PAUSE");
	return 0;
	}
		//題目：xxxx
		//ID-xxxxxx
		#include<stdio.h>
		#include<stdlib.h>
		#include<math.h>
		
		int main(void)
		{
		char r[14],l[14],w;
		int i,n,t,k[14],m[14],p,z,s;
		float b;
		FILE *in, *out;
		~~printf("請輸入羅馬數字 x2 來計算數差(用空格隔開):");~~
		
		in = fopen ("roman.in","r+");
		out = fopen("roman.out","w+");
		
		fscanf(in, "%s %s",r,l**);**
		……
		
		fprintf(out, "%s %s",k,m**);**
		
		fclose(in);
		fclose(out);
		
		~~system("PAUSE");~~
		return 0;
		
		}

這個不用寫

這個不用寫