

제 16회 대전시 정보올림피아드 본선대회 문제

(중등학생부) 수험번호() 이름()

[문제 1] 장미꽃 구입(최대효과)

한 화원에서 장미꽃을 파는데 많이 사면 값을 깎아 주기 때문에 장미꽃 수에 따라 가격이 다르다고 한다. 가격은 한송이를 사면 a_1 원, 2송이를 사면 a_2 원, k 송이를 사면 a_k 원이라고 한다. 대량 유통의 목적으로 포장 단위가 매송이마다 정해져 있지 않다.

필요한 장미꽃의 수가 j 개라 할 때 j 개 이상을 가장 저렴하게 살 수 있는 비용 a 원을 계산하는 프로그램을 완성하여라. j 의 범위는 $[1, 100]$ 으로 한다. 예로 사용될 in.txt 와 out.txt 는 다음과 같다.

in.txt

```
j
1 a1
2 a2
...
k ak
```

예

```
11
1 400
2 700
4 1000
6 1600
```

out.txt

a

예

3000

[문제 2] 카드섹션

운동장에 가로 20명, 세로 20명으로 정렬한 사람들이 검은색 또는 흰색 카드를 들어서 다각형을 만드는 카드 섹션을 하고 있다. 다음 [처리 조건]에 따라 카드 섹션 프로그램을 작성하시오.

<처리 조건>

- ① 행(20명), 열(20명)로 정렬한 사람들의 기준은 왼쪽 상단 모서리에 위치한 사람이다(즉, 왼쪽 상단 모서리의 위치는 (행:1, 열:1)이고, 오른쪽 하단 모서리의 위치는 (행:20, 열:20)이다).
- ② 만들고자하는 다각형의 꼭지점을 다음과 같이 입력한다.
 - 꼭지점 입력은 행, 열의 숫자로 1~20 사이의 정수로 입력한다.

예) 1 행 10 열의 점을 입력한 경우 다음과 같이 입력한다.

row of Vertex 1 : 1

column of Vertex 1 : 10

- 첫 번째로 입력하는 꼭지점은 행의 위치가 가장 작은 꼭지점을 입력한다. 만일 행의 위치가 같은 경우에는 열의 위치가 가장 작은 꼭지점을 입력한다.
 - 나머지 꼭지점들은 다각형에서 첫 번째로 입력한 꼭지점으로부터 오른쪽 방향으로 차례대로 입력한다.
 - 꼭지점의 입력을 마치려면 마침표(.)를 입력한다.
- ③ 다각형 내부에 위치한 사람들은 검은 색 카드를 들고, 다각형 외부에 위치한 사람들은 흰색 카드를 든다. 그리고 다각형의 외각 선에 걸친 사람들은 검은색을 든다(흰색 카드는 '0'으로 검은색 카드는 '#'으로 나타낸다).
 - ④ 출력 결과는 다음 [프로그램 수행 결과]와 같이 한다.
 - ⑤ 프로그램을 수행한 다음에 프로그램 종료 여부를 확인한다.

exit program (y/n) ?

'Y' 또는 'y' 문자를 입력하면 'good-bye!' 메시지 출력과 함께 프로그램을 종료하고, 'N' 또는 'n' 문자를 입력하면 프로그램을 재 수행한다.

<프로그램 수행 결과>

```
row of Vertex 1 : 1
column of Vertex 1 : 1
row of Vertex 2 : 1
column of Vertex 2 : 3
row of Vertex 3 : 10
column of Vertex 3 : 3
row of Vertex 4 : 10
column of Vertex 4 : 1
row of Vertex 5 : .
```

```
###00000000000000000000
###00000000000000000000
###00000000000000000000
###00000000000000000000
###00000000000000000000
###00000000000000000000
###00000000000000000000
###00000000000000000000
###00000000000000000000
###00000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
0000000000000000000000
```

```
exit program (y/n) ? y
```

```
good-bye !
```

[별 점 1-1]

```
# # # # # 0 0 0 (생략)
# # # # # 0 0 0 (생략)
# # # # # 0 0 0 (생략)
# # # # # 0 0 0 (생략)
# # # # # 0 0 0 (생략)
0 0 0 0 0 0 0 0 (생략)
```

[별첨 1-2]

```
0 0 0 0 # 0 0 0 0 0 0 0 (생략)
0 0 0 # # # 0 0 0 0 0 0 (생략)
0 0 # # # # # 0 0 0 0 0 (생략)
0 # # # # # # # 0 0 0 0 (생략)
# # # # # # # # 0 0 0 (생략)
0 0 0 0 0 0 0 0 0 0 0 (생략)
```

[별첨 1-3]

```
# # # # # # # # # 0 0 0 (생략)
0 # # # # # # # 0 0 0 0 (생략)
0 0 # # # # # 0 0 0 0 0 (생략)
0 0 0 # # # 0 0 0 0 0 0 (생략)
0 0 0 0 # 0 0 0 0 0 0 0 (생략)
0 0 0 0 0 0 0 0 0 0 0 (생략)
```

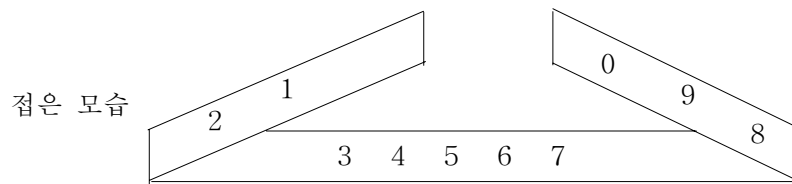
[별첨 1-4]

```
0 0 0 0 # 0 0 0 0 0 0 0 # 0 0 0 0 0 0 0 (생략)
0 0 0 # # # 0 0 0 0 0 # # # 0 0 0 0 0 0 (생략)
0 0 # # # # # 0 0 0 # # # # # 0 0 0 0 0 (생략)
0 # # # # # # # 0 # # # # # # # 0 0 0 0 (생략)
# # # # # # # # # # # # # # 0 0 0 (생략)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (생략)
```

[문제 3] 해싱(주소결정)

Folding은 해싱(hashing) 기법에서 키 값을 가지고 주소를 결정하는 방법중의 한가지이다. Folding은 키(key) 값을 다음과 같은 방법으로 처리하여 주소를 결정한다.

1	2	3	4	5	6	7	8	9	0
---	---	---	---	---	---	---	---	---	---



계산 방식

				2	1	left value	
		3	4	5	6	7	mid value
+				0	9	8	right value
		3	4	6	8	6	=> 계산된 주소

입력 형식 :

1	12
* * * \$ * * * * \$ * *	

\$를 포함한 숫자

12자리의 데이터 입력

출력 형식 :

DATA	ADRESS
14\$63568\$902	73818
513\$27658\$13	28004
....

<처리조건>

- ① 입력 데이터의 '\$' 표시는 접힐 부분을 표시한다.
- ② '\$' 표시는 제일 첫 부분과 제일 끝 부분에 나오지 않으며, 연속되어 나오지 않는다.
- ③ '\$' 표시와 '\$' 표시 사이의 자릿수가 제일 긴 것으로 가정한다.

위 조건에 맞춰 프로그램 하시오.

<프로그램 수행 결과>

Input Key Value : 12\$34567\$890

left : 21
mid : 34567
right : 98

Enter any key

Key Value	Address
12\$34567\$890	34686

Continue ?(Key Value / stop)
513\$27658\$13

left : 315
mid : 27658
right : 31

Enter any key

Key Value	Address
513\$27658\$13	28004

Continue ?(Key Value / stop)
stop

[문제 4] 우유 저장

미국의 텍사스에서 목장을 경영하는 농부 스미스씨는 다음과 같은 크기를 갖는 우유 통을 매우 많이 가지고 있다.

Can	10 gallons
Pail	2 gallons
Gallon	
Quart	1/4 gallon
Pint	1/8 gallon
Cup	1/16 gallon

스미스씨가 X gallon의 우유를 위의 우유 통으로 저장할 수 있는 가지의 수를 계산하는 프로그램을 작성하시오. 예를 들어 스미스씨가 1 Quart를 저장하고자 할 때 4 가지의 방법이 있게 된다.

- 1: 1 quart
- 2: 2 pints
- 3: 1 pint + 2 cups
- 4: 4 cups

1 gallon일 경우는 26 가지의 저장 방법이 있게된다.

<처리조건>

- 1) 입력은 1 - 50 ($1 \leq X \text{ gallons} \leq 50$)이 되도록 한다.
- 2) 프로그램은 입력된 gallon의 값에 대하여 저장할 수 있는 방법의 가지 수를 출력한다.
- 3) 각 값에 대하여 5초 이내로 출력하도록 한다.

<실행 예>

Gallon 수를 입력하세요(끝내려면 아무 문자나 입력하세요) : 1
1 Gallon으로 저장할 수 있는 방법의 수는 26 가지입니다.

Gallon 수를 입력하세요(끝내려면 아무 문자나 입력하세요) : 3
3 Gallon으로 저장할 수 있는 방법의 수는 302 가지입니다.

Gallon 수를 입력하세요(끝내려면 아무 문자나 입력하세요) : q

[문제 5] 로마자와 10진법 대응

여러 분은 로마 왕이었던 마에다스(Midas)의 황금이라는 이야기를 들어보았을 것이다. 우리가 전설로 들어왔던 것과는 다르게 마에다스는 현명하였으며 많은 황금을 팔아 엄청난 부를 축적하였었다. 그런데 전통적인 로마자로는 그 많은 양의 부를 수로서 측정하기가 매우 불편하여 그는 초 로마자(Super Roman Numerals)를 개발하였다.

초기 로마자는 전통적인 로마자의 숫자 표기법을 따르지만 훨씬 더 많은 단위의 값을 각 한자가 갖도록 한 것이다. 로마자의 숫자 표기법은 아래와 같이 각 한자의 로마자에 대하여 10진법의 값으로 대응되도록 한 것이다.

I	1	L	50	M	1000
V	5	C	100		
X	10	D	500		

각 표기자는 최대 값을 나타내는 자를 제외하고 계속 반복해서 3번을 최대한도로 나타낼 수 있다. 따라서 III는 3이며 CCC는 300이 된다.

일반적으로 표기는 내림차 순으로 다음과 같이 연결하여 쓰게된다.

$$CCLXVIII = 100+100+50+10+5+1+1+1 = 268$$

하지만 가끔 이 표기법은 한 자가 그 자보다 한 단위 혹은 두 단위 상위 자 앞에 놓여지기도 한다. 예를 들어 I는 V나 X 앞에, X는 L이나 C 앞에, 등으로 쓸 수 있으며, 이 경우 앞의 자의 값을 뒤의 자의 값에서 뺀다. 예를 들면 다음과 같다.

$$\begin{aligned} IV &= 4 \\ IX &= 9 \\ XL &= 40 \end{aligned}$$

그러나 XD, IC, XM과 같은 결합은 뒤에 있는 상위 단위의 표기자가 앞의 표기자에 대하여 너무 크므로 허락되지 않는다. 예를 들어 490을 표기할 경우 XD는 안되며 CDXC는 되고, 99에 대하여 IC는 안되며 XCIX는 된다. 또한 990을 표기할 경우 XM은 안되며 CMXC는 되게 된다. 따라서 표기를 내림차 순으로 하지 않고 단위가 낮은 표기자를 앞에 놓을 경우 뒤의 표기자는 앞의 표기자에 대하여 10배 보다 크게되는 경우가 있어서는 안 된다는 것이다.

불행하게도 일반적인 로마자에서는 10,000을 표기할 경우 MMMMMMMMMM과 같이 나타내야만 한다. 반면 초 로마자의 경우는 다음과 같은 표기 단위를 정하여 사용하게 된다.

I	1	L	50	M	1,000	R	50,000	U	1,000,000	N	50,000,000
V	5	C	100	P	5,000	S	100,000	B	5,000,000	Y	100,000,000
X	10	D	500	Q	10,000	T	500,000	W	10,000,000	Z	500,000,000

<입력>

입력 파일(roman.txt)로부터 각 행에 대하여 하나 씩 양의 10진 값을 읽어 들이며 입력 값이 0일 경우 프로그램을 종료한다.

<출력>

출력 파일(roman.out)은 각 라인 별로 읽어들인 10진 값에 대응하는 초 로마자 표기를 하는 것이다.

<입.출력 예>

Sample Input

18

1997

12345678

0

Sample Output

18 XVIII

1997 MCMXCVII

12345678 WUUSSSQRPDCLXXVIII