제13회 서울시 정보올림피아드 본선대회 문제

(고등학생부) 수험번호( ) 이름( )

[문제 1] OMR판독

OMR 답안카드를 판독기로 읽은 자료가 주어졌을 때, 읽혀진 자료를 처리 조건에 따라 번역하여 화면에 출력하는 프로그램을 작성하시오.

학생번호	문항	답 란	문항	답 란
⓪ ⓪	1	● ② ③	11	① ② ③
● ①	2	● ② ③	12	① ② ③
② ●	3	● ② ③	13	① ② ③
③ ③	4	● ② ③	14	① ② ③
④ ④	5	● ② ③	15	① ② ③
⑤ ⑤	6	● ② ③	16	① ● ③
⑥ ⑥	7	● ② ③	17	① ● ③
⑦ ⑦	8	● ② ③	18	① ● ●
⑧ ⑧	9	● ② ③	19	① ② ●
⑨ ⑨	10	● ② ③	20	① ② ●
1 2 열 열		3 4 5 열 열 열		6 7 8 열 열 열

<그림1> OMR 답안 카드

J
I
H
G
F
E
D
C
B
A

<그림2> 판독 자료

<처리 조건>

(1) OMR 카드 자료는 <그림1>의 각 열에 표시된 세로 줄을 읽어 문자열 배열 A$에 순서대로 입력되며, 표기된 위치에 따라 <그림2>의 판독 자료와 같은 문자가 다음과 같이 배정된다.

① 열에 표기가 없으면, 그 열의 내용은 1개의 공백만 존재한다.

② 열에 표기가 있으면, 그 위치에 해당되는 문자들을 오름차순으로 정리된 상태로 존재한다.

예) 2열의 ⑥, ⑧ 에 표기되어 있으면 ---> A$(2)="BD"

(2) <그림1>의 OMR 답안카드를 판독기로부터 읽은 자료는 배열 A$에 <보기1>과 같이 저장되어 있다.

<보기1> 판독기로부터 읽은 자료

```
A$(1)="I"                    A$(5)=" "
A$(2)="H"                    A$(6)=" "
A$(3)="ABCDEFGHIJ"           A$(7)="CDE"
A$(4)=" "                    A$(8)="ABC"
```

(3) 프로그램이 실행되면 배열에 저장된 자료를 번역하여 <보기2>와 같이 초기화면을 출력한다.

<보기2> 초기화면

```
   번호:  12
========================================
   답란:  1  2  3  4  5  6  7  8  9  10
========================================
   0-10:  1  1  1  1  1  1  1  1  1  1
  11-20:  B  B  B  B  B  2  2  *  3  3
========================================

     열 번호: ?
     열의 내용:
```

(4) <보기2>에서 '열 번호'와 '열의 내용'을 입력하면 변경된 결과를 <보기3>과 같이 출력한다. 단, '학생번호' 표기의 세로란이나 '답란' 표기의 가로란이 공백(무기입)이면 'B'를 출력하고 2개 이상 중복 표기하였으면 '*'를 출력한다.

<보기3> 열 번호 1번의 내용을 "AB"로 입력한경우

```
   번호:  *2
========================================
   답란:  1  2  3  4  5  6  7  8  9  10
========================================
   0-10:  1  1  1  1  1  1  1  1  1  1
  11-20:  B  B  B  B  B  2  2  *  3  3
========================================

     열 번호: 1
     열의 내용: AB
```

(5) '열 번호'의 범위는 1부터 8까지이며, '열의 내용'은 <그림2>의 판독 자료 중 일부가 오름차순으로 입력된다.

(6) 열 번호에 0을 입력하면 프로그램을 종료한다.

[문제 2] 연산

두 수를 입력하여, 처리 조건에 따라 연산을 실행한 후 결과를 화면에 출력하는 프로그램을 작성하시오.

<처리 조건>

(1) 프로그램을 실행하면 다음과 같이 두 수 a,b를 입력한다.

```
두 수를 입력하시오. ? 19, 450 <Enter>
```

단, 입력되는 수는 모두 1에서 9999사이의 정수이다.

(2) 입력된 각각의 수를 소인수 분해하여 <보기1>과 같이 출력한다.

<보기1> 소인수 분해

```
입력된 수(a,b) : 19 , 450
소인수 분해 :   19 ==> 19
               450 ==> 2 * 3^2 * 5^2

< 스페이스바> 키를 누르시오 !!!
```

(3) <보기1>에서 <스페이스바> 키를 누른 다음, 아래와 같이 문자열을 입력한다.

```
문자열 입력 : 1234567 <Enter>
```

단, 문자열은 숫자만으로 구성되며, 최대 12자리까지 입력이 가능하다.

(4) 입력된 문자열을 수치로 변환한 후, 처음에 입력한 두 수 a, b에 각각 더하여 새로운 수 a1, b1을 만들고, 두 수를 덧셈한 결과를 <보기2>와 같이 출력한다.

<보기2> 큰 정수 더하기

```
***** 큰 정수의 더하기 *****
a1:      1234586
a2:    + 1235017
     ---------------
         2469603

<스페이스바> 키를 누르시오 !!!
```

(5) <보기2>에서 <스페이스바> 키를 누르면, a를 b로 나눈 값의 나머지를 <보기3>과 같이 분자가 1인 진분수의 합으로 출력한다.

<보기3> 분자가 1인 진분수로 변환

```
***** 분자가 1인 진분수 출력 *****

19 / 450 ==> 1 / 24 + 1 / 1800

<스페이스바> 키를 누르시오 !!!
```

(6) <보기3>에서 <스페이스바> 키를 누르면, a를 b로 나눈 값을 <보기4>와 같이 순환소수로 출력한다.

<보기4> 순환소수 출력

```
***** 순환소수 출력 *****

                    .
19 / 450 ==>       .042
순환부분 :    2

<스페이스바> 키를 누르시오 !!!
```

(7) <보기4>에서 <스페이스바> 키를 누르면 "Try again (y/n) ?"를 출력한 후 'y'를 입력하면 다시 실행하고, 'n'을 입력하면 실행을 종료한다.

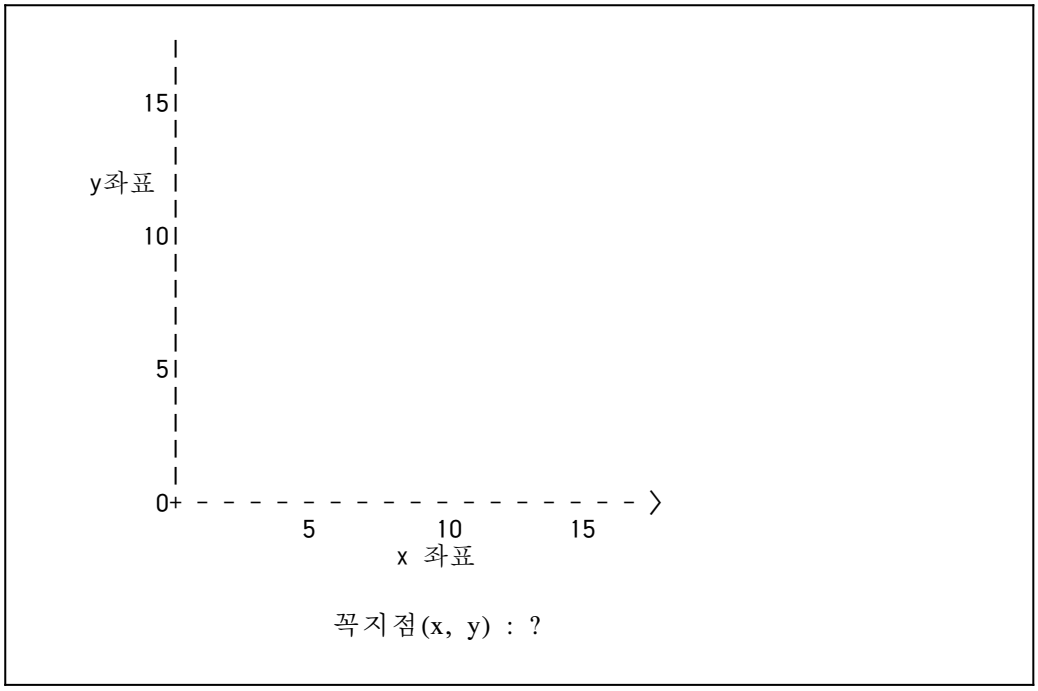
[문제 3] 다각형 면적

다각형의 꼭지점을 모두 입력하였을 때, 처리 조건에 따라 다각형의 모양을 그리고, 면적을 계산하여 출력하는 프로그램을 작성하시오.

<처리조건>

(1) 프로그램을 실행하면 <보기1>과 같은 초기 화면을 출력하고 꼭지점의 좌표 x,y를 입력한다.

<보기1> 초기 화면



(2) <보기1>에서 꼭지점의 입력순서는 다각형이 그려지는 순서를 결정하며, 꼭지점의 수는 최대 7개까지 가능하다.

(3) 입력된 꼭지점에 따라 선분을 그릴 때, 다각형이 되지 않는 경우는 '경고음'을 울리고 좌표의 값을 다시 입력받는다.

(4) 좌표가 입력될 때마다 맨 처음에 입력한 꼭지점의 좌표와 비교하여, 두 좌표가 동일하면 <보기2>와 같이 출력한다.

(5) 다각형을 그리는 기준은 다음과 같다.

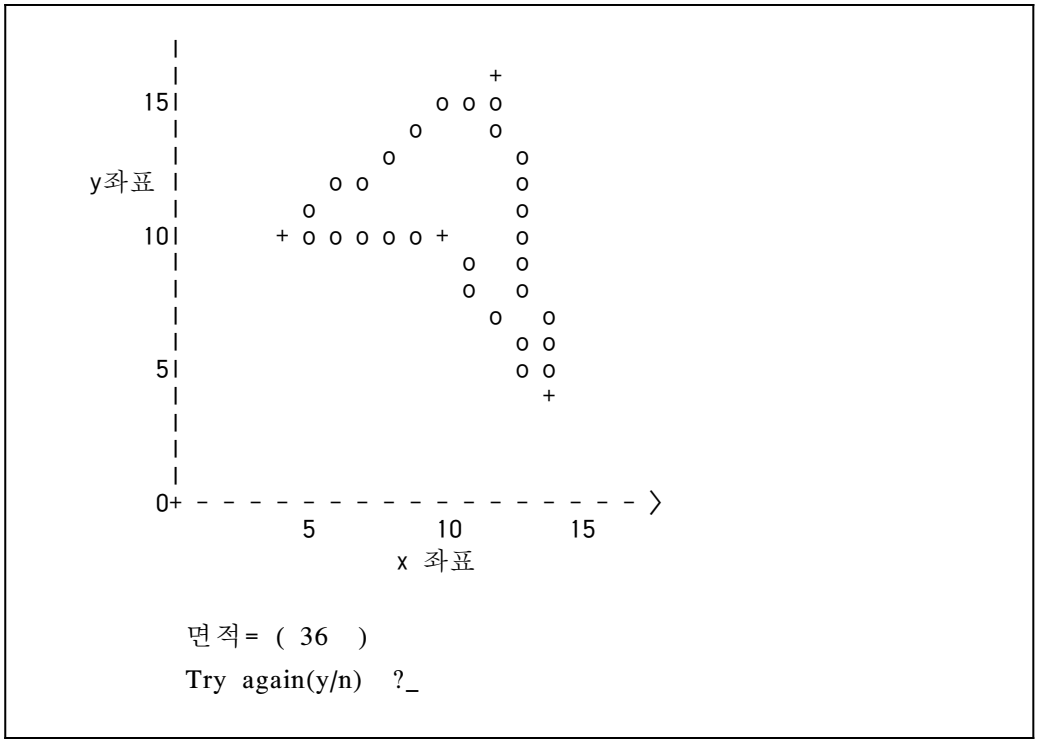
① 다각형을 그리는 좌표의 크기는 17×17 이고, 간격은 가로, 세로 모두 1이다.

② 꼭지점 좌표 위치는 '+'를 출력한다.

③ 꼭지점 사이의 선분은 'o'을 사용하여 출력하며, 출력할 선분의 좌표값이 정수가 아닌 경우는 반올림하여 정수로 처리한다. 단, 면적을 계산할 때는 선분이 모두 직선으로 이어진 것으로 계산한다.

(6) 꼭지점 좌표를 (4,10), (12,16), (14,4), (10,10), (4,10) 순서로 입력하였을 때의 출력화면은 <보기2>와 같다.

<보기2> 다각형 출력



(7) <보기2>에서 'y'를 입력하면 프로그램을 다시 실행하고, 'n'을 입력하면 실행을 종료한다.

[문제 4] 미로탐색

10 ×10의 크기를 갖는 미로를 탐색하는 프로그램을 처리 조건에 따라 작성하시오.

<처리조건>

(1) 프로그램을 실행하면 <보기1>과 같이 초기 화면을 출력한다.

<보기1> 초기 화면

```
                < 미 로 탐 색 >

                  * * * * * * * * * * * *
                9 * 0 0 1 1 0 1 1 1 0 0 *
                8 * 1 0 1 1 0 0 0 0 0 1 *
                7 * 0 0 1 1 0 1 0 1 1 1 *
                6 * 0 1 1 0 0 1 0 0 0 1 *
              Y 5 * 0 0 1 1 0 0 1 1 0 0 *
                4 * 1 0 0 0 0 0 1 1 0 0 1 *
                3 * 1 1 1 1 0 1 1 1 0 1 1 *
                2 * 0 0 1 0 0 0 0 0 0 1 1 *
                1 * 0 1 1 1 1 1 0 1 1 1 1 *
                0 * 0 0 0 0 1 1 0 0 0 0 0 *
                  * * * * * * * * * * * *
                    0 1 2 3 4 5 6 7 8 9
                              X
                  미로수정(+, -) ?_
                  수정좌표(x, y)
```

(2) '미로수정(+,-) ?'에서 '+' 또는 '-'를 입력하면 '수정좌표'를 입력받는다.

① '+'가 입력된 경우는 수정좌표와 인접한 '상', '하', '좌', '우'의 4곳을 모두 '0'으로 변경한다.

② '-'가 입력된 경우는 수정좌표와 인접한 4곳의 값을 '1'로 변경한다.

(3) '미로수정(+,-) ?'에서 'Esc'키를 누르면, '미로수정'을 마치고 다음과 같이 '출발점'과 '도착점'의 좌표를 입력한다.

```
출발점(x1,y1) ? 0, 9 <Enter>
도착점(x2,y2) ? 9, 0 <Enter>
```

(4) 출발점과 도착점 입력 후, 이동 가능한 통로를 <보기2>와 같이 출력한다.

여기서, 미로를 이동하는 규칙은 다음과 같다.

① <보기1>에서 '*'로 표시된 울타리 안의 '0'은 통로이며, '1'은 벽을 표시한다.

② 이동 방향은 '상','하','좌','우' 4방향만 가능하며, 벽이나 울타리가 없어야 한다.

<보기2> 이동 가능한 통로

```
                < 미 로 탐 색 >

                  * * * * * * * * * * * * *
                9 * 0 0     0         0 0 *
                8 *   0     0 0 0 0 0     *
                7 * 0 0     0   0         *
                6 * 0     0 0   0 0 0     *
              Y 5 * 0 0     0 0       0 0 *
                4 *   0 0 0 0       0 0   *
                3 *       0         0     *
                2 *       0 0 0 0 0       *
                1 *           0           *
                0 *           0 0 0 0 0 * *
                  * * * * * * * * * * * * *
                    0 1 2 3 4 5 6 7 8 9
                          X

                출발점: 0, 9
                도착점: 9, 0

                <스페이스바> 키를 누르시오 !!!
```

(5) <보기2>에서 <스페이스바> 키를 누르면, '최단경로'와 '경로의 수'를 <보기3>과 같이 출력한다.

<보기3> 최단경로와 경로의 수

```
                < 미 로 탐 색 >

                  * * * * * * * * * * * * *
                9 * 0 0                     *
                8 *   0                     *
                7 * 0 0                     *
                6 * 0                       *
              Y 5 * 0 0                     *
                4 *   0 0 0                 *
                3 *       0                 *
                2 *       0 0 0             *
                1 *           0             *
                0 *           0 0 0 0 0 * *
                  * * * * * * * * * * * * *
                    0 1 2 3 4 5 6 7 8 9
                          X

                출발점: 0, 9
                도착점: 9, 0      경로의 수 : 2

                Try again(y/n) ?
```

(6) 경로가 없을 때는 <보기4>와 같이 출력한다.

<보기4> 이동 경로가 없을 때

```
    이동 불가능함 !


    <스페이스바> 키를 누르시오 !!!
```

(7) <보기3>에서 'y'를 입력하면 프로그램을 다시 실행하고, 'n'을 입력하면 실행을 종료한다.

[문제 5] 산술식을 문자열로

산술식을 문자열로 입력하여, 처리하는 프로그램을 주어진 처리 조건에 따라 작성하시오.

<처리조건>

(1) 프로그램을 실행하면, 다음과 같이 문자열을 입력한다.

```
문자열 입력: ?_
```

(2) 입력되는 문자열은 한 자리의 정수와 연산기호 +, −, *, ／, ^(거듭제곱), (,) 이다. 단, 이 밖의 문자가 입력되면 오류 메시지를 아래와 같이 출력한 후, <스페이스바> 키를 누르면 문자열을 처음부터 다시 입력한다.

```
입력오류 !
```

(3) 입력된 문자열은 연산 우선 순위에 따라서 연산 순서를 <보기1>과 같이 그림으로 출력한다.

<보기1> 연산 순서 출력

```
문자열 입력: ?  3 + ( 4 + 5 ) ^ 2 - 7 * 6 <Enter>

연산순서:  3  +  (  4  +  5  )  ^  2  -  7  *  6

          I        I     I        I     I     I
          I        +-( 1 )-+        I     +-( 3 )-+
          I           I             I        I
          I           +------( 2 )----+        I
          I                    I               I
          +----------( 4 )---------+             I
                      I                        I
                      +-------------( 5 )------------+

<스페이스바> 키를 누르시오 !!!
```

(4) <보기1>에서 <스페이스바> 키를 누르면, <보기2>와 같이 이진트리(tree) 구조로 출력한다.

<보기2> 이진트리 출력

```
문자열 입력: ?  3 + ( 4 + 5 ) ^ 2 - 7 * 6 <Enter>

이진트리:                    +----------(-)----------+
                             I                       I
                        +---(+)---+                +--(*)--+
                        I         I                I       I
                        3     +---(^)---+          7       6
                              I         I
                           +--(+)--+    2
                           I       I
                           4       5

<ESC>키를 누르시오 !!!
```

(5) <보기2>에서 <ESC> 키를 한 번 누를 때마다 연산되는 과정을 <보기3>과 같이 출력한다. 단, 나눗셈 연산의 경우 소수점 이하는 반올림하여 정수 처리한다.

<보기3> 연산과정 출력

```
(1) <ESC>키를 1회 누른 경우

                    +----------(-)----------+
                    I                       I
               +---(+)---+                +--(*)--+
               I         I                I       I
               3     +---(^)---+          7       6
                     I         I
                  ( 9 )       2

(2) <ESC>키를 2회 누른 경우

                    +----------(-)----------+
                    I                       I
               +---(+)---+                +--(*)--+
               I         I                I       I
               3      ( 81 )              7       6

(3) <ESC>키를 5회 누른 경우

                              ( 42 )

<스페이스바> 키를 누르시오 !!!
```

(6) <보기3>에서 <스페이스바> 키를 누르면 화면을 지우고, <보기4>와 같이 출력한다. 여기서, 스택(A)는 입력된 문자열을 저장한다.

<보기4> 스택을 이용한 연산과정

```
문자열 입력: ?  3 + ( 4 + 5 ) ^ 2 - 7 * 6 <Enter>

    I  3  I  <== 꼭대기(Top)
    I  +  I
    I  (  I
    I  4  I
    I  +  I
    I  5  I
    I  )  I
    I  ^  I
    I  2  I
    I  -  I
    I  7  I
    I  *  I
    I  6  I        I     I        I     I
    +------+       +------+       +------+   <== 바닥(Bottom)
    스택(A)        스택(B)        스택(C)

    <스페이스바> 키를 누르시오 !!!
```

(7) <보기4>에서 <스페이스바> 키를 한 번 누를 때마다 스택(A)의 꼭대기(top)에 있는 문자가 하나씩 처리되어 스택(A)에서 지워진다. 여기서, 꼭대기란 스택의 자료가 저장된 최상의 위치를 말한다.

(8) 스택(A)의 문자를 처리하는 절차는 다음과 같다.

① 꺼내올(pop) 문자가 숫자이면 스택(C)에 push한다.

② 꺼내올 문자가 ‘)’ 이외의 연산자이면 스택(B)의 꼭대기에 있는 연산자와 우선 순위를 비교한다.

③ 꺼내올 문자가 ‘)’ 이면, 스택(B)에서 ‘(’를 만날 때까지 ⑤과정을 반복한다. 단, ‘(’를 만나면 스택(B)에서 ‘(’를 pop하고 스택(A)에서 ‘)’를 pop하여 상쇄하고, ①부터 다시 시작한다.

④ 스택(A)의 연산자 우선 순위가 스택(B)의 연산자 우선 순위보다 크거나 같으면 스택(A)에서 하나의 문자를 pop하여 스택(B)에 push한다.

⑤ ④가 성립하지 않으면 스택(C)에서 숫자를 2개 pop하고, 스택(B)에서 연산자를 pop하여 연산을 수행한다. 연산된 결과는 다시 스택(C)에 push한다.

⑥ 위의 과정을 스택(A)에 있는 모든 문자가 처리될 때까지 반복한다.

(9) 스택(A)와, 스택(B)에서의 연산자 우선 순위는 <보기5>와 같다.

<보기5> 연산자의 우선순위

연산자 / 스택	+	—	*	/	^	(	)
스택(A)	1	1	2	2	3	4	
스택(B)	1	1	2	2	3	0	0

(10) <스페이스바> 키를 누를 때마다 연산되는 과정을 <보기6>과 같이 출력한다.

<보기6> 스택을 이용한 연산과정

```
(1) <스페이스바> 키를 3회 누른 경우

 I  4  I
 I  +  I
 I  5  I
 I  )  I
 I  ^  I
 I  2  I
 I  -  I
 I  7  I
 I  *  I          I  (  I
 I  6  I          I  +  I          I  3  I
 +-----+          +-----+          +-----+
 스택(A)          스택(B)          스택(C)


(2) <스페이스바> 키를 7회 누른 경우

 I  ^  I
 I  2  I
 I  -  I
 I  7  I
 I  *  I          I  (  I          I  9  I
 I  6  I          I  +  I          I  3  I
 +-----+          +-----+          +-----+
 스택(A)          스택(B)          스택(C)

(3) <스페이스바> 키를 13회 누른 경우


                                  I 42  I
 +-----+          +-----+          +-----+
 스택(A)          스택(B)          스택(C)


  Try again(y/n) ?_
```

(11) <보기6>에서 'y'를 입력하면 프로그램을 다시 실행하고, 'n'을 입력하면 실행을 종료한다.