제12회 서울시 정보올림피아드 본선대회 문제

(중학생부) 수험번호( ) 이름( )

[문제 1] 단어 종류 분류

여러 개의 단어로 구성된 문자열 하나를 입력하여, 처리조건에 따라 단어를 종류별로 분류하는 프로그램을 작성하시오.

<처리조건>

(1) 문자열은 프로그램 실행 중에 입력한다.

(2) 문자열을 구성하는 요소는 <보기 1>과 같은 문자만을 사용한다. 그 이외의 문자를 입력하면 경고음(Beep음)을 출력하고, 잘못 입력한 문자는 받아들이지 않는다.

<보기 1> 입력 가능한 문자

① 영문자: A~Z, a~z ② 숫자: 0~9 ③ 특수문자: %,$,#,*,! ④ 엔터키: <Enter>

(3) 엔터(Enter)키를 치면 문자열 입력을 종료한다.

(4) 영문자의 소문자와 대문자는 동일 문자로 처리하며, 소문자가 입력되면 대문자로 변환하여 출력한다.

(5) 입력되는 문자열의 길이는 50자 이내이다.

(6) 각 단어의 길이는 15자 이내이다.

(7) 문자열을 분류하는 기준은 다음과 같다.

① 단어와 단어 사이는 하나 이상의 공백으로 구분한다.

② 단어의 종류는 “명령어”, “변수”, “상수”, “기타”로 한다.

③ 명령어는 “read”와 “print”이다.

④ 변수는 영문자와 숫자를 혼용할 수 있으나, 첫 글자는 반드시 영문자라야 한다.

⑤ 상수는 0에서 9까지의 정수만을 사용한다.

⑥ “명령어”, “변수”, “상수” 이외의 단어는 “기타”로 한다.

⑦ 같은 단어는 하나만 출력한다.

(8) 프로그램을 실행하면 <보기 2>와 같이 메시지를 출력한 후, 문자열을 입력한다.

<보기 2> 문자열의 입력

문자열을 입력하세요?_ PRINT (korea) $## 2002 KO+REA Worldcup

(9) 문자열의 입력이 종료되면, 분류된 단어와 단어의 종류를 <보기 3>과 같이 출력한다.

<보기 3> 분류된 단어와 단어의 종류 출력

```
입력 문자열 : PRINT korea $## 2002 KOREA Worldcup
분류된 단어 : PRINT      ==>  명령어
              KOREA      ==>  변수
              $##        ==>  기타
              2002       ==>  상수
              WORLDCUP   ==>  변수

      Try again(Y/N) ?_
```

(10) "Try again(Y/N)?"를 출력한 후 'Y'를 입력하면 다시 실행하고, 'N'을 입력하면 실행을 종료한다.

[문제 2] 반편성

어느 중학교에서 반편성을 하기 위하여 모의고사를 보았다. 모의고사 총점을 가지고 다음 처리조건에 따라 반편성(반배정)을 하는 프로그램을 작성하시오.

<처리조건>

(1) 입력되는 데이터는 다음과 같으며, 프로그램 내부에 직접 기술하도록 한다.

학번	이름	성별	총점
1101	AAA	M	290
1102	BBB	M	285
1103	CCC	F	260
1104	DDD	M	282
1105	EEE	F	279
1106	FFF	F	268
1107	GGG	M	250
1108	HHH	M	218

학번	이름	성별	총점
1109	III	F	235
1110	JJJ	M	255
1111	KKK	F	287
1112	LLL	F	278
1113	MMM	M	275

* 성별은 남자는 ' M'
* 여자는 ' F' 로 표기함

(2) 프로그램을 실행하면 <보기1>과 같이 학번을 출력한 후 2명의 데이터를 이름, 성별, 총점순으로 입력받는다.

<보기 1> 데이터의 추가 입력

```
# 1114번 ?_ NNN,F,288
# 1115번 ?_ OOO,M,259
```

(3) <보기2>와 같이 편성할 반의 수를 입력받는다.

<보기 2> 반의 수 입력 ('5'를 입력한 경우)

```
# 편성할 반의 수 ?_ 5
```

편성할 반의 수는 2~9까지의 정수로 입력하며, 입력이 잘못되면 'Input error'를 출력한 후 다시 입력받는다.

(4) 반편성을 하는 절차는 다음과 같다.

① 총점이 높은 학생부터 석차를 부여한다.(단, 총점이 같은 학생은 없는 것으로 가정한다.)

② 남학생은 홀수반에, 여학생은 짝수반에 배정을 한다.

③ 석차가 1등인 학생부터 반을 배정하며 <보기 3>과 같이 'ㄹ'자 배정을 한다.

④ 남학생은 앞반에서 뒤쪽으로 배정을 시작하고, 여학생은 뒷반에서 앞쪽으로 배정을 시작한다.

<보기 3> 반배정의 보기 (편성할 반의 수가 '5'인 경우)

```
          1반   3반   5반                          2반   4반
남자반 ================                  여자반  ===========
          1     2     3                            2     1
          6     5     4                            3     4
          7     8     9                            6     5
               ....   10                            7     ...
```

(5) 배정이 완료되면, <보기 4>와 같이 남자반의 배정 결과를 출력한다.

<보기 4> 반배정 석차순 출력(남자반)

```
 <<  남 자 반  출 력 >>
 학번  이름  성별  총점  석차   배정된 반
=========================================
1101  AAA    M    290     1        1
1102  BBB    M    285     4        3
1104  DDD    M    282     5        5
1113  MMM    M    275     8        5
1115  OOO    M    259    11        3
1110  JJJ    M    255    12        1
1107  GGG    M    250    13        1
1108  HHH    M    218    15        3
```

(6) 임의의 키를 누르면, <보기 5>와 같이 여자반의 배정 결과를 출력한다.

<보기 5> 반배정 석차순 출력(여자반)

```
 <<  여 자 반  출 력 >>
 학번  이름  성별  총점  석차   배정된 반
=========================================
 1114  NNN    F    288    2        4
 1111  KKK    F    287    3        2
 1105  EEE    F    279    6        2
 1112  LLL    F    278    7        4
 1106  FFF    F    268    9        4
 1103  CCC    F    260   10        2
 1109  III    F    235   14        2
```

(7) <ESC>키를 누르면 프로그램의 실행을 종료하고, 그 이외의 키가 입력되면 (3)항의 '편성할 반수' 입력부터 다시 반복 실행한다.

[문제 3] 형태판별

가로, 세로 10×10의 크기를 갖는 문자 배열 안에 5개의 'X'와 5개의 'O'를 사용하여 두개의 모양을 만들었다. 두 개의 모양이 같은 형태를 이루는가를 판별하는 프로그램을 다음의 처리조건에 따라 작성하시오.

<처리조건>

(1) 모양을 이루는 5개의 문자 'X'는 가로, 세로 또는 대각선 방향으로 인접해 있고, 마찬가지로 5개의 문자 'O'도 서로 인접해 있다.

(2) 다음의 방법을 적용하여 두 개의 모양이 완전히 겹쳐질 수 있다면 두 모양은 같은 형태를 이루는 것으로 판별한다.

① 둘 중 하나를 선택하여 90도씩 시계방향(또는 반시계방향)으로 회전시켜 본다.

② 상, 하 또는 좌, 우로 이동시켜 본다.

(3) 프로그램을 실행하면 <보기 1>과 같이 모양을 입력받는다. (공백은 '.'으로 입력한다.)

(4) 'O', 'X', '.' 이외의 문자가 입력되거나, 또는 'O'와 'X'의 수가 5개를 초과하면 'Input error'를 출력하고 다시 입력받는다.

(5) 줄의 맨 처음에 엔터(Enter)키를 치면 그 줄은 'O','X'가 전혀 없는 것으로 하고, 다음 줄의 자료를 입력받는다.

<보기 1> 모양의 입력

```
[입력]       1       5         10
  1번 줄 --> <Enter>
  2번 줄 --> <Enter>
  3번 줄 --> <Enter>
  4번 줄 --> . . . . . O . . . <Enter>
  5번 줄 --> . . . . . O O . . <Enter>
  6번 줄 --> . . . . . O . . . <Enter>
  7번 줄 --> . . X . . . O . . . <Enter>
  8번 줄 --> X X X X . . . . . <Enter>
  9번 줄 --> <Enter>
 10번 줄 --> <Enter>        * <Enter>는 키보드의 엔터키를 의미함
```

(6) 모양의 입력이 끝났을 때, 두 개의 모양이 일치하면 '모양 일치'를 출력하고, 불일치하면 '모양 불일

치'를 출력한다. 몇 가지 모양 입력 후 결과를 <보기 2>와 <보기 3>에 나타내었다.

<보기 2> 불일치하는 모양의 2가지 예

```
[입력]      1       5       10      [입력]      1       5       10
 1번 줄 --> . . . . . . . . . .       1번 줄 --> . . . . . . . . . .
 2번 줄 --> . . . . . . . . . .       2번 줄 --> . . . . . . . . . .
 3번 줄 --> . . . . . . . . . .       3번 줄 --> . . . . . . . . . .
 4번 줄 --> . . . . . 0 . . .         4번 줄 --> . . X X . . 0 0 . .
 5번 줄 --> . . . . . 0 0 . .         5번 줄 --> . . X . . . 0 . . .
 6번 줄 --> . . . . . 0 . . .         6번 줄 --> . . X . . . 0 . . .
 7번 줄 --> . . X . . . 0 . . .       7번 줄 --> . X . . . . . 0 . .
 8번 줄 --> X X X X . . . . . .       8번 줄 --> . . . . . . . . . .
 9번 줄 --> . . . . . . . . . .       9번 줄 --> . . . . . . . . . .
 10번 줄 --> . . . . . . . . . .      10번 줄 --> . . . . . . . . . .
[출력]                               [출력]
 모양 불일치                          모양 불일치
```

<보기 3> 일치하는 모양의 2가지 예

```
[입력]      1       5       10      [입력]      1       5       10
 1번 줄 --> . . . . . . . . . .       1번 줄 --> . . . . . . . . . .
 2번 줄 --> . . . . . 0 . . . .       2번 줄 --> . . . . X . . . . .
 3번 줄 --> . . . . . . 0 . . .       3번 줄 --> . . . X . . . . . .
 4번 줄 --> . . . . . . . 0 . .       4번 줄 --> . . X X X . . . . .
 5번 줄 --> . . . . X X 0 0 . .       5번 줄 --> . . . . 0 . 0 . . .
 6번 줄 --> . . . X . X . . . .       6번 줄 --> . . . . . 0 0 . . .
 7번 줄 --> . . X . . . . . . .       7번 줄 --> . . . . . . 0 . . .
 8번 줄 --> . . . . . . . . . .       8번 줄 --> . . . . . . . . . .
 9번 줄 --> . . . . . . . . . .       9번 줄 --> . . . . . . . . . .
 10번 줄 --> . . . . . . . . . .      10번 줄 --> . . . . . . . . . .
[출력]                               [출력]
 모양 일치                            모양 일치
```

(7) 'Try again(Y/N)'을 출력한 후 'Y'를 입력하면 처음부터 다시 실행하고, 'N'을 입력하면 실행을 종료한다.

[문제 4] 해쉬함수

해시함수(Hash function)를 이용한 정렬(Sort) 프로그램을 작성하고자 한다. 정렬하고자 하는 자료의 최대값을 m이라 하고 자료의 총 갯수를 n이라 할 때, 해시함수 h는 다음과 같이 정의된다고 하자.

h = int(m/n) + 1

다음 처리조건에 따라 수치 자료를 오름차순으로 정렬하는 프로그램을 작성하시오.

<처리조건>

(1) 정렬할 자료는 다음과 같이 모두 양의 정수이며, 1~99 사이의 수이다.

5, 2, 3, 4, 19, 66, 5, 27, 1, 26, 31, 16, 21

(2) 자료는 프로그램 내부에 직접 기술한다.

(3) 프로그램을 실행하면, <보기 1>과 같이 정렬할 자료의 수를 입력받는다.

<보기 1> 정렬할 자료의 수 입력

```
# 자료의 수 ?_
```

자료의 수는 0~13까지로 제한하며, 그 이외의 값이 입력되면 'Input error'를 출력하고 다 시 입력받는다. (단, '0'을 입력하면 프로그램의 실행을 종료한다.)

(4) 자료의 수가 입력되면, <보기 2>와 같이 초기화면을 출력한다.

<보기 2> 초기화면(자료의 수가 10인 경우)

```
# 자료의 수 ? 10
정렬할 자료 :  5  2  3  4  19  66  5  27  1  26
          +---+---+---+---+---+---+---+---+---+---+
방번호(Key) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
          +---+---+---+---+---+---+---+---+---+---+
배정된자료  |   |   |   |   |   |   |   |   |   |   |
          +---+---+---+---+---+---+---+---+---+---+
순번: 1          Key: 0     현재 배정할 자료: 5
```

(5) <Space bar>키를 한번 누를 때마다 하나씩 빈 방에 자료를 배정하며, 배정된 방의 상태를 화면에 출력한다. 배정하는 순서와 방법은 다음과 같다.

① 자료의 배정 순번은 입력된 순서대로 앞에서부터 차례대로 배정한다.

② 배정할 방번호(Key)는 다음의 공식에 의하여 계산한다.

Key = int(현재 배정할 자료/해시함수(h))

③ 만일 Key에 해당되는 방에 이미 다른 자료가 배정되어 있으면 ④~⑧의 방법으로 처리한다.

④ 현재 배정할 자료를 A, Key의 방에 배정된 자료를 B라고 할 때, A와 B를 비교한다.

⑤ A>=B 이면 Key를 1 증가시킨다.

⑥ ⑤의 결과, 만일 Key에 해당되는 방이 비어 있으면 그 곳에 자료를 배정한다.

⑦ ⑤의 결과, 만일 Key에 해당되는 방이 비어 있지 않으면 ⑤의 과정을 반복한다.

⑧ A<B 이면 Key 오른쪽의 빈방을 찾고, Key부터 빈방까지의 자료를 한 칸씩 오른쪽으로 이동시킨 후 자료를 배정한다. (단, 빈방이 없으면 화면 하단에 '정렬 실패'를 출력하고 다시 실행한다.)

(6) 자료를 방에 배정한 보기를 <보기 3>~ <보기 6>에 나타내었다.

<보기 3> <보기 2>의 화면에서 <Space bar>를 1회 누른 경우

```
# 자료의 수 ? 10
정렬할 자료 :  5  2  3  4  19  66  5  27  1  26
          +---+---+---+---+---+---+---+---+---+---+
방번호(Key) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
          +---+---+---+---+---+---+---+---+---+---+
배정된자료  | 5 |   |   |   |   |   |   |   |   |   |
          +---+---+---+---+---+---+---+---+---+---+
순번: 2          Key: 0     현재 배정할 자료: 2
```

<보기 4> <보기 2>의 화면에서 <Space bar>를 6회 누른 경우

```
# 자료의 수 ? 10
정렬할 자료 :  5  2  3  4  19  66  5  27  1  26
          +---+---+---+---+---+---+---+---+---+---+
방번호(Key) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
          +---+---+---+---+---+---+---+---+---+---+
배정된자료  | 2 | 3 | 4 | 5 | 19|   |   |   |   | 66|
          +---+---+---+---+---+---+---+---+---+---+
순번: 7          Key: 4     현재 배정할 자료: 5
```

<보기 5> <보기 2>의 화면에서 <Space bar>를 10회 누른 경우

```
# 자료의 수 ? 10
정렬할 자료 :  5  2  3  4  19  66  5  27  1  26
          +---+---+---+---+---+---+---+---+---+---+
방번호(Key) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
          +---+---+---+---+---+---+---+---+---+---+
배정된자료  | 1 | 2 | 3 | 4 | 5 | 5 | 19| 26| 27| 66|
          +---+---+---+---+---+---+---+---+---+---+
정렬결과 --> 1  2  3  4  5  5 19 26 27 66
```

<보기 6> 정렬 실패의 보기 (자료의 수 '4'를 입력하고 <Space bar>키를 4회 누른 경우)

```
# 자료의 수 ? 4
정렬할 자료 :  5  2  3  4
          +---+---+---+---+
방번호(Key) | 0 | 1 | 2 | 3 |
          +---+---+---+---+
배정된자료  |   | 2 | 3 | 5 |
          +---+---+---+---+
정렬실패 !
```

(7) 자료 배정이 모두 끝난 후 <Space bar>키를 누르면 처음부터 다시 실행한다.

[문제 5] 최단경로

도로가 N×N 격자 모양(바둑판 모양)으로 된 도시가 있다. 다만, 일부 교차로의 공사 관계로 통행이 제한되는 경우가 있다. 통행이 제한되는 교차로의 좌표가 주어졌을 때 두 지점간의 최단거리와 최단경로의 수를 구하는 프로그램을 처리조건에 따라 작성하시오.

<처리조건>

(1) 좌표의 입력은 왼쪽 위를 (1,1), 오른쪽 아래를 (N,N)으로 정한다.

(2) 프로그램을 실행하면, <보기 1>과 같이 격자의 크기 'N'과 통행이 제한되는 교차로의 위치(좌표)를 입력받는다.

<보기 1> 격자의 수와 통행이 제한되는 교차로의 수 입력

```
# 격자의 수 ?_
# 통행제한 교차로 수 ?_
```

격자의 수는 2~7까지, 통행제한 교차로 수는 0~5 까지로 제한하며, 그 이외의 값이 입력되면 'Input error'를 출력하고 다시 입력받는다.

(3) <보기 2>와 같이 도로의 모양을 화면에 출력한 후 통행제한 교차로의 좌표를 입력받고, 입력된 통행제한 교차로의 위치에는 'A'를 표시한다.

<보기 2> 통행제한 교차로의 좌표 입력

```
# 격자의 수 ? 4
# 통행제한 교차로 수 ? 3
     1    2    3    4  <-- X
   1 +----A----+----+
     |    |    |    |
   2 +----A----+----+
Y    |    |    |    |
   3 +----A----+----+
     |    |    |    |
   4 +----+----+----+
# 통행제한 교차로(X,Y) ? 2,1
# 통행제한 교차로(X,Y) ? 2,2
# 통행제한 교차로(X,Y) ? 2,3
```

(4) 이동할 두 지점의 좌표 (X1,Y1)과 (X2,Y2)를 입력받는다. 단, X1≦X2 이고, Y1≦Y2이다.

<보기 3> 이동할 두 지점 입력 ((1,1)에서 (4,4)까지 이동할 경우)

```
# 출발 위치(X1,Y1) ? 1,1
# 도착 위치(X2,Y2) ? 4,4
```

(5) 두 지점간의 최단거리와 최단경로의 수를 <보기 4>와 같이 출력한다. 단, 교차로 사이의 거리(간격)은 모두 동일하며 1Km이다.

<보기 4> 결과 출력

```
최단거리 --->  6 Km
최단경로의 수 ---> 1
```

(6) <Space bar>키를 누르면 이동할 두 지점을 다시 입력 받고 결과를 출력한다. 이 때 도로의 상태를 출력한 <보기 2>의 화면(도로의 모양이 출력된 화면)이 지워지거나 또는 위로 올라가지 않도록 처리한다.

```
# 출발 위치(X1,Y1) ? 1,1
# 도착 위치(X2,Y2) ? 4,1

최단거리 --->  9 Km
최단경로의 수 ---> 4
```

(7) 이동할 경로의 수가 없는 경우에는 '이동 불가능'을 출력하고, 이동할 지점을 다시 입력받는다.
(8) '출발 위치(X1,Y1) ?'의 물음에 '0, 0' 을 입력하면 프로그램의 실행을 종료한다.