

제 19 회
서울특별시 정보올림피아드 경시부문

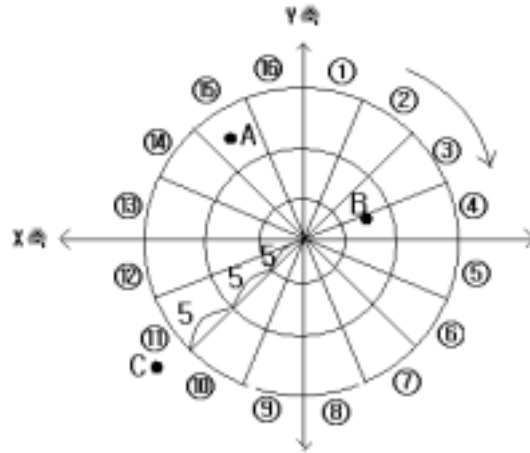
본 선 문 제

고 등 학 생 부

수험번호() 성명()

서 울 특 별 시 교 육 청

【문제 1】 다트 경기에서 한 판이 끝나고 난 후에 과녁에 N개의 화살이 꽂혀 있다. 이 화살들의 점수 합을 구하는 프로그램을 작성하고자 한다. 과녁은 다음의 그림과 같이 3개의 동심원으로 나뉘어져 있다. 과녁 중심의 좌표가 (0,0)이고 화살이 꽂혀 있는 위치는 좌표로 주어진다. 처리 조건에 따라 프로그램을 작성하시오.



(A,B,C는 화살이 꽂힌 자리임)

<처리 조건>

- (1) 화살의 개수 N은 10개 이하로 한다.
- (2) 동심원과 동심원 간의 거리는 5이다.
- (3) 과녁을 이루고 있는 16개의 부채꼴 모양은 모두 서로 합동이다.
- (4) 점수 계산 방법은 다음과 같다.

- ① 시계 방향으로 돌아가면서 각 구간의 기본점수는 번호와 같이 1~16점이다.
- ② 가장 바깥 동심원의 밖으로 벗어난 화살은 0점이다.
- ③ 과녁의 가운데 좌표(0,0)에 꽂힌 화살은 200점이다.
- ④ 좌표(0,0)을 제외하고, 제일 안쪽 동심원 안에 꽂힌 화살의 점수는 기본점수×10, 제일 안쪽 동심원과 두 번째 동심원 사이에 꽂힌 화살의 점수는 기본점수×5, 두 번째 동심원과 제일 바깥 동심원 사이에 꽂힌 화살의 점수는 기본점수 그대로이다.
- ⑤ 구간의 경계선에 꽂힌 화살은 높은 점수를 받는 쪽으로 처리한다.

- (5) 화살의 개수를 입력한다.

화살의 개수 ? 3

- (6) 화살의 X,Y 좌표를 입력한다.(단, X, Y는 정수이다.)

1번째 화살의 좌표 ? -9,12
 2번째 화살의 좌표 ? 8,3
 3번째 화살의 좌표 ? -14,-14

- (7) 같은 좌표가 중복 입력되면 “입력오류!” 메시지를 출력하고 다시 입력받는다.

1번째 화살의 좌표 ? -9,12
 2번째 화살의 좌표 ? 8,3
 3번째 화살의 좌표 ? -9,12
 입력오류!

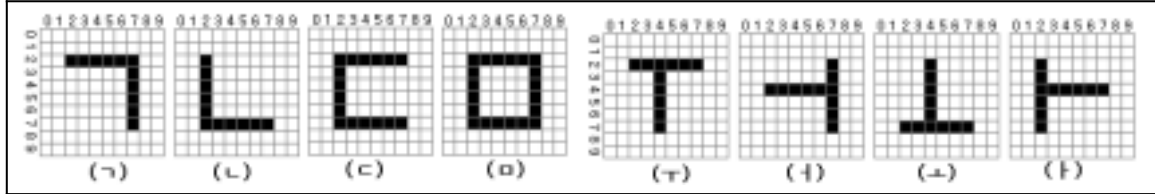
- (8) 최종 점수를 계산해서 출력한다.

최종 점수 : 35

고등학교부 2 쪽

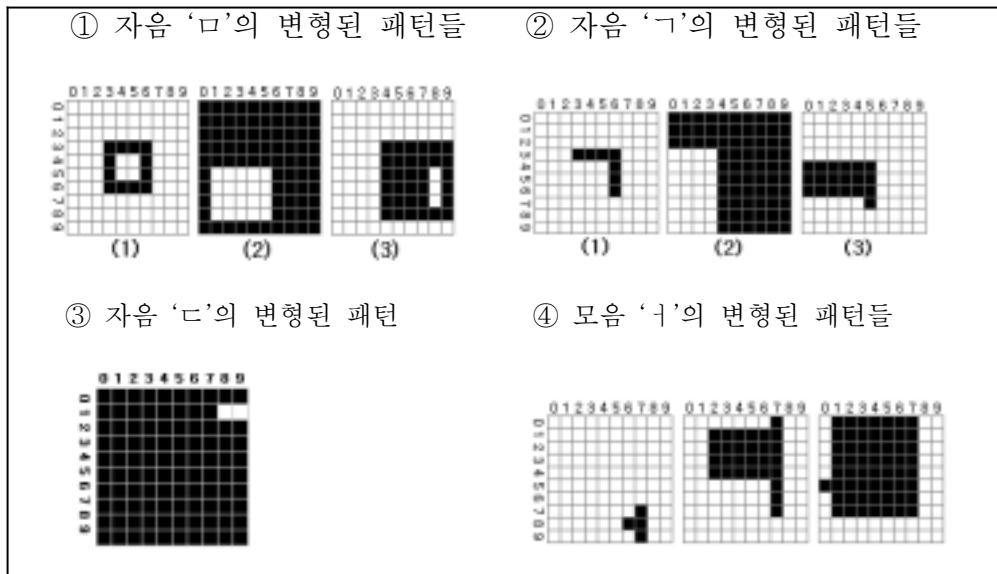
【문제 2】 다음은 패턴을 인식하고, 인식된 패턴에 해당하는 문자를 출력하는 문제이다. 다음 <보기1>은 한글의 자음 ‘ㄱ’, ‘ㄴ’, ‘ㄷ’, ‘ㄹ’과 모음 ‘ㅏ’, ‘ㅑ’, ‘ㅓ’, ‘ㅕ’에 대한 기본 패턴이다. 가로와 세로의 해상도는 10×10이다.

<보기1>



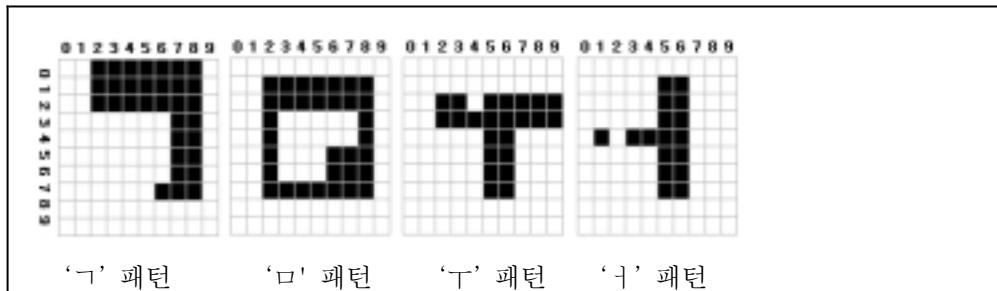
다음 <보기2>는 <보기1>에서 제시한 자음과 모음 패턴의 변형된 형태이다. 지정된 매트릭스 안에서 패턴의 위치는 어디에 있어도 관계없으며, 패턴의 크기도 관계없다. <보기1>에서 제시한 기본 패턴의 윤곽을 분명하게 갖추면 된다.

<보기2>



다음 <보기3>은 기본 패턴에서 벗어난 잘못된 형태의 예이다.

<보기3>



다음 <처리 조건>에 따라서 패턴을 인식하여 그 결과를 출력하는 프로그램을 작성하시오.

<처리 조건>

- (1) 매트릭스내의 패턴은 2진 데이터로 바꾸어 컴퓨터가 인식할 수 있도록 해야한다.

10×10 매트릭스에서 패턴문자로 표시된 화소(Cell)는 1로 표시하고, 공백인 화소는 0으로 표시한다. 따라서 <보기1>의 ‘ㄱ’은 다음과 같은 형식으로 표시할 수 있다.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

- (2) 위의 패턴데이터는 다음과 같은 형식으로 표현한다.

```
0A 0A 021602 071102 071102 071102 071102 071102 0A 0A
```

매트릭스의 각 줄의 패턴데이터는 공백으로 구분한다. 첫 번째 패턴데이터는 매트릭스의 첫 줄의 데이터로써, 0A는 0이 연속해서 A번(16진, 10진수로는 10번) 나왔다는 뜻이다. 세 번째 패턴데이터 021602는 먼저 0이 2번, 1이 6번 그리고 0이 2번 연속적으로 나왔다는 의미이다.

- (3) 글자 패턴의 가로×세로의 해상도는 각각 4×4이상, 10×10이하로 제한한다.
 (4) 프로그램을 실행하면 패턴데이터를 읽어들이는 파일명을 입력받는다. 파일명에 아무 것도 입력하지 않고 엔터를 누르면 프로그램을 종료한다.
 (5) 패턴데이터 자료 파일은 다음과 같은 형태의 텍스트 파일이다. 다음의 예는 4개의 패턴데이터 자료이다. 이 자료를 넣고 실행하면 (6)과 같은 결과가 나와야 한다.

```
0A 0A 021602 071102 071102 071102 071102 071102 0A 0A
0A 0A 021107 021107 021107 021107 021107 021602 0A 0A
0A 0A 021602 021107 021107 021107 021107 021602 0A 0A
0A 0A 021602 0211041102 0211041102 0211041102 0211041102 021602 0A 0A
```

- (6) 패턴데이터를 처리한 결과를 아래와 같이 출력한다. 입력받은 패턴데이터를 처리하여 일치하는 기본 패턴을 찾을 수 없으면 “패턴 오류” 메시지를 출력한다.

```
입력파일명 : a:\h21.txt
문자 : ㄱㄴㄷㄹ
```

- (7) 처리 결과를 출력한 다음에 자동으로 (4) 단계로 돌아가 파일명 입력을 기다린다.

고등학교부 4 쪽

【문제 3】 새로 입고되는 물건들을 창고 안에 놓으려 한다. 창고는 여러 칸으로 나뉘어 있으며, 각 칸마다 물건을 옮기는데 사용되는 노력값이 다르다. 물건을 옮기는데 사용된 누적 노력값이 가장 적게드는 방법을 찾아내어 그때의 누적 노력값을 출력하는 프로그램을 작성하시오.

열

		0	1	0	1
		4	0	4	1
		4	1	-1	1
		1	-1	1	1
		1	1	1	1

행

<창고 상태 예>

모든 좌표 표시는 (열,행)으로 한다. 위 <창고 상태 예>에서 표시된 부분의 좌표는 (4,1)이다.

위 표의 각 칸에 들어있는 값은 물건을 들고 그 칸을 지나갈 때에 사용되는 노력값이다. 물건은 한 번에 하나씩 이동시키는 것으로 한다. 노력값이 -1로 표시된 칸은 지나갈 수도 없고 물건을 놓을 수도 없다. 노력값이 -1이 아닌 칸은 모두 지나갈 수 있다.

노력값이 4인 칸에는 이미 다른 물건이 놓여 있다. 노력값이 1과 0으로 표시된 칸들은 현재 비어있으며, 새로 들어오는 물건은 노력값이 0인 칸에만 놓일 수 있다.

<처리 조건>

- (1) 열의 수와 행의 수는 각각 2 이상이고 50 이하이어야 한다. 또한 “열의 수 * 행의 수”는 4 이상이고 100 이하이어야 한다.
- (2) 열의 수 및 행의 수와 관계없이 새로 들어온 물건은 3개이며 물건이 놓일 칸도 3개로 한다. 만약 물건이 놓일 칸이 3개가 아니면 “Place Number Error!!!”를 출력한다.
- (3) 새로 들어온 물건은 노력값이 0인 칸이면 어디든지 들어갈 수 있다. 즉, 위의 <창고 상태 예>에서 첫 번째 물건은 (1,1) 또는 (3,1), (2,2) 어느 곳에나 놓을 수 있다. 만일 첫번째 물건을 (1,1)에 놓았다면 두 번째 물건은 (3,1) 또는 (2,2) 중 하나에 놓아야 하며, 세 번째 물건은 남은 칸에 놓아야 한다.
- (4) 새로 들어온 물건을 놓으면 그 칸의 노력값은 4가 된다. 즉 첫 번째 물건을 (3,1)에 놓았다면 다음부터 (3,1)을 지날 때에는 노력값이 4으로 계산되어야 한다.
- (5) 현재 있는 곳에서 다른 칸으로 이동할 때에는 현재 있는 칸의 노력값을 적용한다. 즉, (4,1) →(4,2)→(3,2)→(2,2)로 이동하면 $1 + 1 + 4 = 6$ 만큼의 노력을 한 것이다.
- (6) 빈 몸으로 돌아올 때의 노력값은 누적 노력값에 포함시키지 않는다.
- (7) (1,1), (3,1), (2,2)의 노력값을 0으로 지정한 이유는 그 칸이 물건이 놓일 자리임을 알리기 위한 것이다. 그러나, 물건을 운반하는 경로로 그 칸을 지날 때에는 노력값을 1로 계산하여야 한다.
- (8) 물건의 이동은 상, 하, 좌, 우 어느 쪽으로든 가능하지만 대각선으로는 불가능하다.
- (9) 세 개의 물건을 다 옮길 수 있어야 한다. 하나라도 옮길 수 없으면 “Fail to put!!!”을 출력한다.
- (10) 프로그램을 실행하면 누적 노력값 자료를 읽어들이는 파일명(a:\h31.txt)을 입력받는다. 파일명에 아무 것도 입력하지 않고 엔터를 누르면 프로그램을 종료한다.
- (11) 다음은 앞의 <창고 상태 예>에 해당되는 자료를 텍스트 파일 형태로 나타낸 것이다.

내용	설명
4 5	← 열의 수, 행의 수
4 1	← 창고 입구의 위치 좌표 (열, 행)
0 1 0 1	← 각 칸에서 물건을 이동할 때에 드는 노력값
4 0 4 1	
4 1 -1 1	
1 -1 1 1	
1 1 1 1	

(12) 위의 자료를 읽어들이 물건을 옮기는데 사용한 최소 누적 노력값을 찾아내어 아래와 같이 출력한다.

파일명 입력 : a\h31.txt
최소 누적 노력값 = 7

(13) 처리 결과를 출력한 다음에 자동으로 (10) 단계로 돌아가 파일명 입력을 기다린다.

고등학교부 6 쪽

【문제 4】 if 문과 두 개의 변수 Pan과 Lee 등을 이용하여 작성한 프로그램을 번역하여 실행한 후, 변수 Lee의 값을 출력하는 간단한 언어 번역기를 만들고자 한다. 예를 들어 다음 <보기>와 같은 프로그램을 처리하면 변수 Lee의 값이 9가 되어야 한다. 이렇게 처리하는 번역기를 <처리 조건>에 따라 만드시오.

<보기>

```
Pan=1
if (Pan>0) begin
  Lee=9
end
else begin
  Lee=0
end
```

<처리 조건>

- (1) 다음에 소개되는 명령어와 연산자만을 사용한다.
- (2) 다음은 if문의 형식으로, if문 다음의 조건이 참일 경우에만 ‘첫 번째 문장’을 수행한다.

```
if ( 조건 ) begin
  첫 번째 문장
end
```

- (3) 다음 if문의 조건이 참이면 ‘첫 번째 문장’을 수행하고, 조건이 거짓이면 ‘두 번째 문장’을 수행한다. ‘세 번째 문장’은 if문의 조건에 관계없이 수행한다.

```
if ( 조건 ) begin
  첫 번째 문장
end
else begin
  두 번째 문장
end
세 번째 문장
```

- (4) 다음 if문에서, 조건1과 조건2가 모두 참이면 ‘첫 번째 문장’을 수행한 다음 ‘두 번째 문장’을 수행한다. 조건1은 참이고 조건2가 거짓이면 ‘두 번째 문장’만을 수행한다. 조건1이 거짓이면 ‘세 번째 문장’만을 수행한다.

```
if ( 조건1 ) begin
  if ( 조건2 ) begin
    첫 번째 문장
  end
  두 번째 문장
end
else begin
  세 번째 문장
end
```

- (5) 다음은 이 프로그램에서 사용하는 논리 연산자들의 기능을 설명한 것이다.

연산자	사용 예	기능
>	Pan > Lee	변수 Pan의 값이 변수 Lee의 값보다 크면 참이다.
<	Pan < Lee	변수 Pan의 값이 변수 Lee의 값보다 작으면 참이다.
==	Pan == Lee	변수 Pan의 값과 변수 Lee의 값이 같으면 참이다.
=	Pan = 10	변수 Pan에 10을 저장한다.
+	Lee + 7	변수 Lee의 값과 7을 더한다.
-	Lee - 7	변수 Lee의 값에서 7을 뺀다.

- (6) 예를 들어 다음과 같은 문장을 입력하여 실행하면, 변수 Lee의 값은 97이 된다.

```
Pan=-1
if (Pan== -1) begin
    if (Pan<0) begin
        Lee=98
    end
    Lee=Lee+Pan
end
else begin
    Lee=Pan+1
end
```

- (7) 문법 상의 오류가 있으면 “데이터 오류”를 출력한다.
(8) 프로그램을 실행하면 자료를 읽어들이 파일명(a:\h41.txt)을 입력받는다. 파일명에 아무 것도 입력하지 않고 엔터를 누르면 프로그램을 종료한다.
(9) 다음은 입력할 내용을 담은 텍스트 파일의 예이다. 텍스트 파일을 만들 경우에는 모든 명령을 좌측으로 정렬하여 입력하도록 한다.

```
Pan=1
if (Pan>0) begin
    Lee=97
end
else begin
    Lee=0
end
```

- (10) 위의 텍스트 파일을 읽어들이 실행한 결과를 출력한다.

```
파일명 입력 : a:\h41.txt
Lee = 97
```

- (11) 처리 결과를 출력한 다음에 자동으로 (8) 단계로 돌아가 파일명 입력을 기다린다.

고등학교부 8 쪽

【문제 5】 다음은 주어진 문장에서 반복되는 문자열을 찾는 프로그램이다. <보기>와 같은 문장에서 반복되는 문자열은 "abc"이다. <처리 조건>에 따라서 주어진 문장에서 반복되는 문자열을 찾아내는 프로그램을 작성하시오.

<보기>

abckoabcbgcbcf

<처리 조건>

(1) 반복 문자열이란 같은 문자열이 두 번 이상 반복되는 문자열을 말한다. ①의 문장에서는 "abc"가 반복 문자열이지만, ②의 문장에서는 "abc"가 반복 문자열이 아니다.

① gabcckoabcbfg

② gabccklseuzmn

(2) 반복 문자열에서는 가장 긴 길이로 반복되는 문자열이다. 즉 다음과 같은 문장에서 반복 문자열은 "abcde"만이 반복 문자열이며, "ab", "abc", "bc", "abcd" 등은 반복 문자열이 아니다.

sabcdejkabcdeui

(3) 반복 문자열이 이미 찾은 반복 문자열의 일부분일 경우에는 반복 문자열로 간주하지 않는다. 다음에서 "abcde"는 반복 문자열이지만 "cde"는 반복 문자열이 아니다.

teabcdeuztcdescdejkabcdeu

(4) 반복 문자열의 특정 위치에 있는 문자 하나가 틀린 경우에도 반복 문자열로 간주한다. 예를 들면 다음 보기에서 "abc~~v~~xyz"과 "abc~~w~~xyz", "abc~~e~~xyz"은 같은 위치에 있는 'v', 'w', 'e' 만 다를 뿐 나머지 내용들은 완전히 일치한다. 이 세 개의 문자열은 반복 문자열로 간주하며, "abc?xyz"와 같이 표시한다.

abcvxyzrocqabcwxyzfeoabcexyzqp

'?' 표시는 하나의 반복 문자열에 한 개까지만 허용된다. 다음의 예에서 "ab?cd?ef"는 '?'가 두 개 들어가므로 반복 문자열이 아니고, "ab?cd"와 "ef"가 반복 문자열이 된다.

abxcdyefvgoxabpcdef

(5) 반복 문자열이 두 문자 이상을 건너서 반복될 경우에는 반복 문자열로 간주하지 않는다. 즉, 다음에서 "abc??def"는 한 개의 반복 문자열이 아니고, "abc"와 "def"가 각기 다른 두 개의 반복 문자열이다.

fabcswdefkopabczodefr

(6) 입력과 출력은 다음과 같은 형식으로 키보드로부터 입력받아 결과를 화면에 출력한다. 반복 문자열이 여러 개일 경우에는 쉼표로 구분하여 출력한다.

문자열은? **gabcrdefkoabcdefop**
반복 데이터 : abc?def

문자열은? **abtydkxegfaberbkxe**
반복 데이터 : ab, kxe

(7) 입력받은 데이터에 반복 문자열이 없으면 다음과 같이 출력한다.

문자열은? **knetfrs**
반복 데이터 : 없음