

**제 15회 서울시 정보올림피아드 본선대회 문제**  
 (고등학생부)                      수험번호(                      )    이름(                      )

**[문제 1] 숫자바꾸기**

입력한 숫자들을 순서대로 배열하고, 배열된 숫자의 위치를 일정한 규칙에 따라 바꾸어 주는 프로그램을 처리 조건에 따라 작성하시오.

<처리조건>

(1) 프로그램을 실행하면 다음과 같이 2개 이상 20개 이하의 숫자들을 입력한다.

2개 이상, 20개 이하의 숫자들을 입력하시오 => 1,2,3,4,5,6

(2) 숫자들의 위치 이동 규칙은 다음과 같다.

① 처음 입력한 숫자

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|

② 첫 번째 이동

각각의 순서를 다음과 같이 1→6, 2→5, 3→4, 4→3, 5→2, 6→1로 이동 시킨다.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|

③ 두 번째 이동

앞의 결과에 이어서, 둘씩 묶어서 (6,5)→(2,1), (4,3)→(4,3), (2,1)→(6,5)로 이동 시킨다.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 2 | 1 | 4 | 3 | 6 | 5 |
|---|---|---|---|---|---|

④ 세 번째 이동

앞의 결과에 이어서, 셋씩 묶어서 (2,1,4)→(3,6,5), (3,6,5)→(2,1,4)로 이동 시킨다.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 3 | 6 | 5 | 2 | 1 | 4 |
|---|---|---|---|---|---|

⑤ 네 번째 이동

앞의 결과에 이어서, 넷씩 묶어서 (3,6,5,2)→(5,2,1,4), (1,4)→(3,6)로 이동 시킨다.

(단, 뒤의 둘로 묶은 (1,4)의 경우와 같이 짝이 모자라서 넷씩 묶을 수 없을 때는 나머지를 한 묶음으로 한다.)

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 4 | 3 | 6 | 5 | 2 |
|---|---|---|---|---|---|

⑥ 다섯 번째 이동

앞의 결과에 이어서, 다섯씩 묶어서 (1,4,3,6,5)→(4,3,6,5,2), (2)→(1)로 이동 시킨다.

(단, 뒤의 (2)의 경우와 같이 짝이 모자라서 다섯씩 묶을 수 없을 때는 나머지를 하나의 묶음으로 한다.)

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 2 | 1 | 4 | 3 | 6 | 5 |
|---|---|---|---|---|---|

⑦ 여섯 번째 이동

앞의 결과에 이어서, 각각의 순서를 다음과 같이 2→5, 1→6, 4→3, 3→4, 6→1, 5→2로 이동 시킨다.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 5 | 6 | 3 | 4 | 1 | 2 |
|---|---|---|---|---|---|

⑧ 일곱 번째 이동

앞의 결과에 이어서, 둘씩 묶어서 (5,6)→(1,2), (3,4)→(3,4), (1,2)→(5,6)로 이동 시킨다.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|

(3) 입력받은 숫자들의 자리 이동을 몇번 할 것인지 입력한다.

|                       |
|-----------------------|
| 몇 번째 이동 결과를 출력할까요 ? 6 |
|-----------------------|

(4) 정해진 규칙에 따라 이동을 반복 하다보면, 처음 입력된 순서와 같아질 때가 있다. 이 때의 이동 횟수보다 큰 수를 입력했을 때 다음과 같은 오류 메시지를 출력한다.

|  |
|--|
| 몇번째 이동 결과를 출력할까요 ? 9<br>오류 발생 !!<br>7번 자리 이동하면 처음 입력된 순서와 같아집니다. 다시 입력하십시오. => |
|--|

(5) 조건에 맞는 숫자가 입력되면, 입력한 숫자들을 정해진 규칙에 따라 입력된 수만큼 자리 이동시킨 결과를 출력한다.

|                            |
|----------------------------|
| 6 번째 이동한 결과 => 5,6,3,4,1,2 |
|----------------------------|

(6) 결과를 출력한 후 프로그램을 (1)번부터 다시 시작한다.

(7) 프로그램 진행도중에 <ESC> 키를 누르면 실행을 끝낸다.

## [문제 2] 체스(QUEEN)배치

[그림 1]과 같이 8×8인 체스판에 Queen('Q'로 표시)이 있으면 그 Queen을 중심으로 가로, 세로, 대각선 방향(음영으로 처리된 곳)에는 다른 Queen을 놓을 수 없다. 8 개의 Queen이 모두 이러한 규칙을 만족하면서 체스판에 배치되도록 다음의 처리 조건에 따라 프로그램을 작성하시오.

[그림 1] Queen의 공격 범위

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 |   |   |   |   |   |   |   |   |
| 2 |   |   |   |   |   |   |   |   |
| 3 |   |   |   |   |   |   |   |   |
| 4 |   |   |   | Q |   |   |   |   |
| 5 |   |   |   |   |   |   |   |   |
| 6 |   |   |   |   |   |   |   |   |
| 7 |   |   |   |   |   |   |   |   |
| 8 |   |   |   |   |   |   |   |   |

### <처리조건>

- (1) 프로그램을 실행하면 <보기 1>과 같이 가로, 세로가 8×8인 체스판을 출력하고 첫번째 Queen이 놓일 임의의 위치에 대한 좌표를 입력받는다.

<보기 1> 초기 입력 화면

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |                        |
|---|---|---|---|---|---|---|---|---|------------------------|
| 1 |   |   |   |   |   |   |   |   | 위치 입력(가로, 세로) : ( _, ) |
| 2 |   |   |   |   |   |   |   |   |                        |
| 3 |   |   |   |   |   |   |   |   |                        |
| 4 |   |   |   |   |   |   |   |   |                        |
| 5 |   |   |   |   |   |   |   |   |                        |
| 6 |   |   |   |   |   |   |   |   |                        |
| 7 |   |   |   |   |   |   |   |   |                        |
| 8 |   |   |   |   |   |   |   |   |                        |

- (2) 입력받은 위치에 첫번째 Queen(화면에 '<Q>'로 표시함)을 출력하고 문제의 규칙을 만족하도록 나머지 7 개의 Queen(화면에 'Q'로 표시함)을 배치하여 <보기 2>와 같이 출력한다.

<보기 2> 가로 4, 세로 2를 입력한 화면

|   | 1 | 2 | 3 | 4   | 5 | 6 | 7 | 8 |
|---|---|---|---|-----|---|---|---|---|
| 1 |   | Q |   |     |   |   |   |   |
| 2 |   |   |   | <Q> |   |   |   |   |
| 3 |   |   |   |     |   | Q |   |   |
| 4 |   |   |   |     |   |   |   | Q |
| 5 |   |   | Q |     |   |   |   |   |
| 6 | Q |   |   |     |   |   |   |   |
| 7 |   |   |   |     |   |   | Q |   |
| 8 |   |   |   |     | Q |   |   |   |

위치 입력(가로, 세로) : (4,2)

다음 경우는 스페이스바를 누름!!

(3) 첫번째 Queen이 고정된 경우 나머지 7 개의 Queen이 배치된 형태는 여러 가지가 존재한다. 따라서 스페이스바를 누르면 규칙을 만족하는 새로운 형태의 배치 화면을 <보기 3>과 같이 출력한다.

<보기 3> 스페이스바를 누른 후의 화면

|   | 1 | 2 | 3 | 4   | 5 | 6 | 7 | 8 |
|---|---|---|---|-----|---|---|---|---|
| 1 |   |   |   |     |   | Q |   |   |
| 2 |   |   |   | <Q> |   |   |   |   |
| 3 | Q |   |   |     |   |   |   |   |
| 4 |   |   |   |     | Q |   |   |   |
| 5 |   |   |   |     |   |   |   | Q |
| 6 |   | Q |   |     |   |   |   |   |
| 7 |   |   |   |     |   |   | Q |   |
| 8 |   |   | Q |     |   |   |   |   |

위치 입력(가로, 세로) : (4,2)

다음 경우는 스페이스바를 누름!!

(4) 스페이스바를 누를 때마다 새로운 배치 화면을 보여주며 더 이상 배치 화면이 없는 경우에는 <보기 4>와 같이 서로 다른 배치가 가능한 경우의 수를 출력한다.

<보기 4> 마지막 배치 화면

|   | 1 | 2 | 3 | 4   | 5 | 6 | 7 | 8 |
|---|---|---|---|-----|---|---|---|---|
| 1 |   |   |   |     |   |   |   | Q |
| 2 |   |   |   | <Q> |   |   |   |   |
| 3 | Q |   |   |     |   |   |   |   |
| 4 |   |   | Q |     |   |   |   |   |
| 5 |   |   |   |     |   | Q |   |   |
| 6 |   | Q |   |     |   |   |   |   |
| 7 |   |   |   |     |   |   | Q |   |
| 8 |   |   |   |     | Q |   |   |   |

위치 입력(가로, 세로) : (4,2)

가능한 경우는 8 가지임

종료?(y/n)

(5) '종료?(y/n)' 물음에 'y'를 입력하면 새로운 위치를 입력받기 위해 초기 입력 화면으로 되돌아가며, 'n'을 입력하면 프로그램을 종료한다.

### [문제 3] 다각형판별

XY-좌표평면 위에서 다각형의 꼭지점 좌표를 입력받아 볼록 다각형인지 오목 다각형인지 판별하는 프로그램을 처리 조건에 따라 작성하시오.

<처리조건>

(1) 프로그램을 실행하면 <보기1>과 같이 다각형의 꼭지점 좌표를 입력받는다.

(단, 꼭지점의 수는 10개 이하이고,  $-20 \leq x \leq 20$ ,  $-10 \leq y \leq 10$ 이다.)

<보기1> 다각형의 꼭지점 좌표 입력

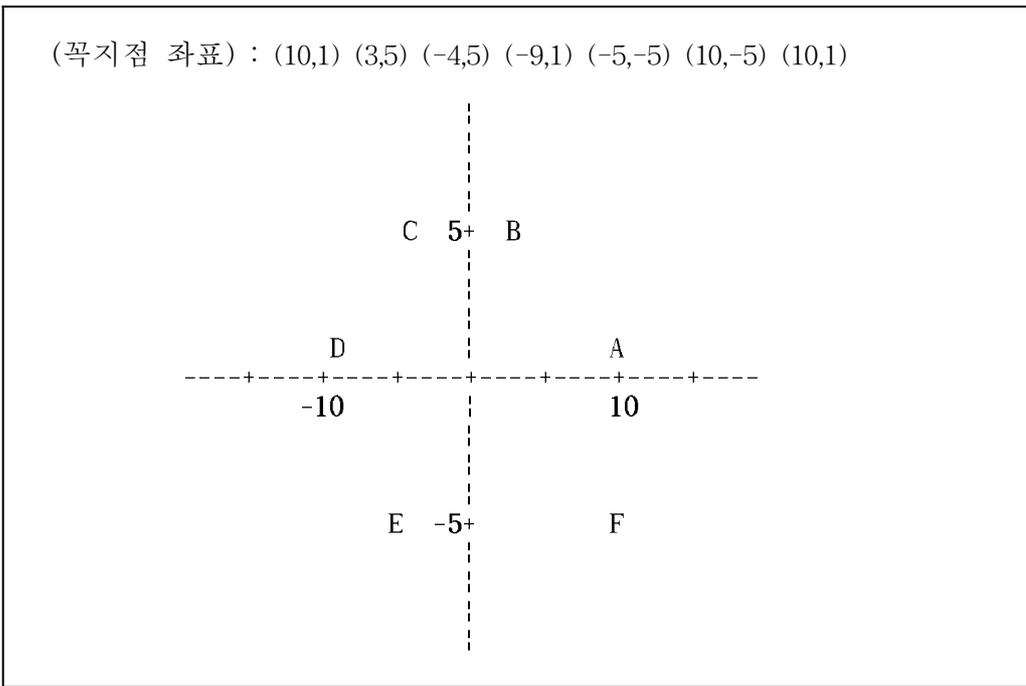
꼭지점(x,y)? 10,1  
 꼭지점(x,y)? 3,5  
 꼭지점(x,y)? -4,5  
 꼭지점(x,y)? -9,1  
 꼭지점(x,y)? -5,-5  
 꼭지점(x,y)? 10,-5  
 꼭지점(x,y)? 10,1

(2) 입력한 꼭지점의 좌표가 맨 처음 입력한 값과 일치하면 입력을 종료한다.

(3) 화면에 좌표 축을 그리고, 입력한 꼭지점의 좌표를 <보기2>와 같이 표시한다.

(단, <보기2>의 그림에서 다각형을 구성하는 선분은 실제 화면에 출력하지 않는다.)

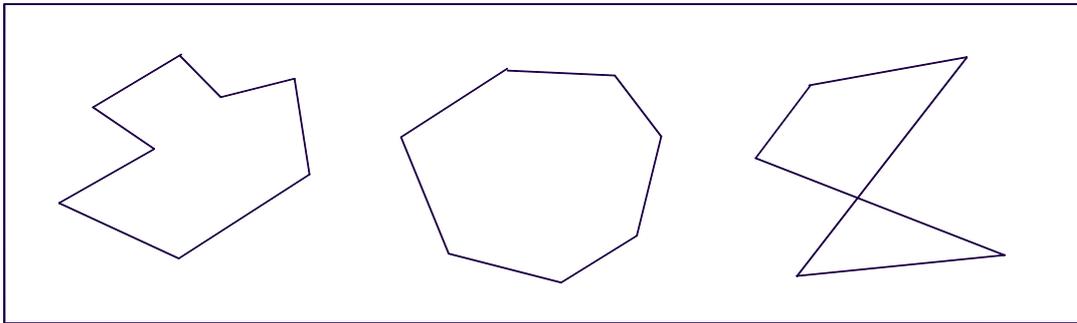
<보기2> 다각형의 꼭지점 출력(꼭지점의 순서는 알파벳 문자 'A'~'Z'을 사용함)



- (4) 꼭지점을 입력한 순서대로 연결하였을 때 선분이 서로 교차하는 경우 ‘다각형이 아닙니다’ 라는 메시지를 출력하고, 처음부터 다시 입력한다.
- (5) <스페이스바>를 누르면 다각형의 형태를 판단한 결과를 다음과 같이 출력한다.

결과 : 볼록 다각형입니다.

- (6) 볼록 다각형, 오목 다각형 및 다각형이 아닌 경우의 예를 <보기3>에 나타내었다.  
 <보기3> 오목 다각형, 볼록 다각형 및 다각형이 아닌 경우의 예



오목 다각형

볼록 다각형

다각형이 아닙니다.

- (7) 판단 결과가 오목 다각형인 경우 ‘오목다각형입니다.’를 출력하고, 볼록 다각형인 경우 ‘볼록다각형입니다.’를 출력한다.
- (8) ‘종료(y/n)?’ 물음에 ‘y’를 입력하면 다시 실행하고, ‘n’을 입력하면 프로그램을 종료한다.

## [문제 4] 멀티프로세싱

다수의 프로그램을 동시에 메모리(주기억장치)에 적재하고, CPU가 이들을 일정한 시간 단위로 순회하면서 처리하는 멀티 프로세싱 시스템이 있다. 이러한 멀티 프로세싱 방식을 모의 실험하는 프로그램을 처리 조건에 따라 작성하시오.

### <처리조건>

(1) 처리해야 하는 프로그램의 초기 입력 자료는 <표1>과 같으며 프로그램 상의 데이터로 구현한다.(프로그램에 직접 기술함)

<표1> 초기 입력 자료(대기 중인 프로그램 목록)

| 순번 | 프로그램 명칭 | 크기 | 처리 요구 시간 |
|----|---------|----|----------|
| 1  | A       | 2  | 3        |
| 2  | B       | 2  | 2        |
| 3  | C       | 3  | 5        |
| 4  | D       | 1  | 1        |
| 5  | E       | 3  | 4        |
| 6  | F       | 3  | 3        |
| 7  | G       | 7  | 6        |
| 8  | H       | 2  | 2        |
| 9  | I       | 1  | 3        |
| 10 | J       | 3  | 1        |
| 11 | K       | 2  | 2        |
| 12 | L       | 3  | 5        |

(2) <표1>의 내용은 다음과 같다.

- ① ‘순번’은 프로그램이 메모리에 적재되는 순서이며, 최대 12개까지 있을 수 있다.
- ② ‘프로그램명칭’은 프로그램을 구별하기 위하여 사용한다.
- ③ ‘크기’는 각 프로그램의 용량을 말하며, 1이상 10이하의 정수이다.
- ④ ‘요구시간’은 프로그램이 종료되기까지의 CPU 사용 시간(초)이다.

(3) 사용할 수 있는 메모리의 최대 크기는 10이다.

(4) 처리 방법은 다음과 같다.

- ① CPU는 1초씩 메모리에 적재된 프로그램들을 들어온 순서대로 순회 처리한다.
- ② 프로그램의 메모리는 한정되어 있으므로, 항상 처리 가능한 용량만큼 프로그램이 메모리에 적재 될 수 있다.
- ③ 적재된 프로그램 중 어느 하나가 종료하면, 그 프로그램이 적재 되었던 곳은 빈 공간이 된다.
- ④ 프로그램을 메모리에 적재할 때는 메모리의 상위번지부터 하위번지로 빈 공간을 순차 검색하여 적재 가능한 공간이 있으면 적재한다. (프로그램의 명칭을 메모리의 해당위치에 표시함)
- ⑤ 메모리에 적재된 순번이 낮을수록 CPU의 사용 순번이 낮으며, CPU 사용 순번을 알 수 있도록 화면에 준비 큐(Ready Que)를 사용한다.
- ⑥ 새로 적재된 프로그램의 경우 준비큐의 맨 뒤에 위치한다.

- ⑦ 준비큐의 중간에 있는 프로그램이 먼저 종료하는 경우 그 프로그램은 준비큐에서 삭제되고, 준비큐의 이 후의 목록은 왼쪽으로 한칸씩 이동한다.
  - ⑧ <스페이스바>를 1회 누를 때마다 1초의 시간이 흐른 것으로 처리하며, 작업 전환을 위하여 중간에 소요되는 시간은 없는 것으로 가정한다.
  - ⑨ 시간이 흐를 때마다 변화된 상황을 작업목록의 '남은시간'에 출력한다. 이 때, 남은 시간이 0이면 '종료'를 표시한다.
- (5) 프로그램을 실행하면 <보기1>과 같이 초기화면을 출력한다.  
 <보기1> 초기화면

| (준비 큐)  |   |         |    |    |      |      |
|---------|---|---------|----|----|------|------|
| A B C D |   |         |    |    |      |      |
| (메모리)   |   | (작업 목록) |    |    |      |      |
|         |   | 순번      | 명칭 | 크기 | 요구시간 | 남은시간 |
| 상위      | A | 1       | A  | 2  | 3    | 3    |
|         | A | 2       | B  | 2  | 2    | 2    |
|         | B | 3       | C  | 3  | 5    | 5    |
|         | B | 4       | D  | 1  | 1    | 1    |
|         | C | 5       | E  | 3  | 4    | 4    |
|         | C | 6       | F  | 3  | 3    | 3    |
|         | C | 7       | G  | 7  | 6    | 6    |
|         | D | 8       | H  | 2  | 2    | 2    |
|         | D | 9       | I  | 1  | 3    | 3    |
| 하위      |   | 10      | J  | 3  | 1    | 1    |
|         |   | 11      | K  | 2  | 2    | 2    |
|         |   | 12      | L  | 3  | 5    | 5    |

경과시간=0  
 계속하려면 <스페이스바>를 누르시오.!!!

- (6) <보기1>의 초기화면에서 <스페이스바>를 1회, 2회, 3회, 5회, 10회, 37회 누른 후의 변화된 상황은 다음과 같다.
- ① <스페이스바>를 1회 누른 경우

| (준비 큐)  |   |         |    |    |      |                         |
|---------|---|---------|----|----|------|-------------------------|
| A B C D |   |         |    |    |      |                         |
| (메모리)   |   | (작업 목록) |    |    |      |                         |
|         |   | 순번      | 명칭 | 크기 | 요구시간 | 남은시간                    |
| 상위      | A | 1       | A  | 2  | 3    | 2 <-- (화살표는 출력하지 않아도 됨) |
|         | A | 2       | B  | 2  | 2    | 2                       |
|         | B | 3       | C  | 3  | 5    | 5                       |
|         | B | 4       | D  | 1  | 1    | 1                       |
|         | C | 5       | E  | 3  | 4    | 4                       |
|         | C | 6       | F  | 3  | 3    | 3                       |
|         | C | 7       | G  | 7  | 6    | 6                       |
|         | D | 8       | H  | 2  | 2    | 2                       |
|         | D | 9       | I  | 1  | 3    | 3                       |
| 하위      |   | 10      | J  | 3  | 1    | 1                       |
|         |   | 11      | K  | 2  | 2    | 2                       |
|         |   | 12      | L  | 3  | 5    | 5                       |

경과시간=1

- ② <스페이스바>를 2회 누른 경우

| (준비 큐)  |   | (작업 목록) |    |    |      |       |
|---------|---|---------|----|----|------|-------|
| A B C D |   | 순번      | 명칭 | 크기 | 요구시간 | 남은시간  |
| 상위      | A | 1       | A  | 2  | 3    | 2     |
|         | A | 2       | B  | 2  | 2    | 1 <-- |
|         | B | 3       | C  | 3  | 5    | 5     |
|         | B | 4       | D  | 1  | 1    | 1     |
|         | C | 5       | E  | 3  | 4    | 4     |
|         | C | 6       | F  | 3  | 3    | 3     |
|         | C | 7       | G  | 7  | 6    | 6     |
|         | D | 8       | H  | 2  | 2    | 2     |
|         |   | 9       | I  | 1  | 3    | 3     |
| 하위      |   | 10      | J  | 3  | 1    | 1     |
|         |   | 11      | K  | 2  | 2    | 2     |
|         |   | 12      | L  | 3  | 5    | 5     |

경과시간=2

③ <스페이스바>를 3회 누른 경우

| (준비 큐)  |   | (작업 목록) |    |    |      |       |
|---------|---|---------|----|----|------|-------|
| A B C D |   | 순번      | 명칭 | 크기 | 요구시간 | 남은시간  |
| 상위      | A | 1       | A  | 2  | 3    | 2     |
|         | A | 2       | B  | 2  | 2    | 1 <-- |
|         | B | 3       | C  | 3  | 5    | 4 <-- |
|         | B | 4       | D  | 1  | 1    | 1     |
|         | C | 5       | E  | 3  | 4    | 4     |
|         | C | 6       | F  | 3  | 3    | 3     |
|         | C | 7       | G  | 7  | 6    | 6     |
|         | D | 8       | H  | 2  | 2    | 2     |
|         |   | 9       | I  | 1  | 3    | 3     |
| 하위      |   | 10      | J  | 3  | 1    | 1     |
|         |   | 11      | K  | 2  | 2    | 2     |
|         |   | 12      | L  | 3  | 5    | 5     |

경과시간=3

④ <스페이스바>를 5회 누른 경우

| (준비 큐)  |   | (작업 목록) |    |    |      |       |
|---------|---|---------|----|----|------|-------|
| A B C E |   | 순번      | 명칭 | 크기 | 요구시간 | 남은시간  |
| 상위      | A | 1       | A  | 2  | 3    | 1 <-- |
|         | A | 2       | B  | 2  | 2    | 1     |
|         | B | 3       | C  | 3  | 5    | 4     |
|         | B | 4       | D  | 1  | 1    | 종료    |
|         | C | 5       | E  | 3  | 4    | 4     |
|         | C | 6       | F  | 3  | 3    | 3     |
|         | C | 7       | G  | 7  | 6    | 6     |
|         | E | 8       | H  | 2  | 2    | 2     |
|         | E | 9       | I  | 1  | 3    | 3     |
| 하위      | E | 10      | J  | 3  | 1    | 1     |
|         |   | 11      | K  | 2  | 2    | 2     |
|         |   | 12      | L  | 3  | 5    | 5     |

경과시간=5

⑤ <스페이스바>를 10회 누른 경우

| (준비 큐) |   | (작업 목록) |    |    |      |      |
|--------|---|---------|----|----|------|------|
| C E F  |   | 순번      | 명칭 | 크기 | 요구시간 | 남은시간 |
| 상위     | F | 1       | A  | 2  | 3    | 종료   |
|        | F | 2       | B  | 2  | 2    | 종료   |
|        | F | 3       | C  | 3  | 5    | 2    |
|        |   | 4       | D  | 1  | 1    | 종료   |
|        | C | 5       | E  | 3  | 4    | 3    |
|        | C | 6       | F  | 3  | 3    | 3    |
|        | C | 7       | G  | 7  | 6    | 6    |
|        | E | 8       | H  | 2  | 2    | 2    |
|        | E | 9       | I  | 1  | 3    | 3    |
| 하위     | E | 10      | J  | 3  | 1    | 1    |
|        |   | 11      | K  | 2  | 2    | 2    |
|        |   | 12      | L  | 3  | 5    | 5    |

경과시간=10

⑥ <스페이스바>를 37회 누른 경우

| (준비 큐) |  | (작업 목록) |    |    |      |      |
|--------|--|---------|----|----|------|------|
|        |  | 순번      | 명칭 | 크기 | 요구시간 | 남은시간 |
| 상위     |  | 1       | A  | 2  | 3    | 종료   |
|        |  | 2       | B  | 2  | 2    | 종료   |
|        |  | 3       | C  | 3  | 5    | 종료   |
|        |  | 4       | D  | 1  | 1    | 종료   |
|        |  | 5       | E  | 3  | 4    | 종료   |
|        |  | 6       | F  | 3  | 3    | 종료   |
|        |  | 7       | G  | 7  | 6    | 종료   |
|        |  | 8       | H  | 2  | 2    | 종료   |
|        |  | 9       | I  | 1  | 3    | 종료   |
| 하위     |  | 10      | J  | 3  | 1    | 종료   |
|        |  | 11      | K  | 2  | 2    | 종료   |
|        |  | 12      | L  | 3  | 5    | 종료   |

경과시간=37

(7) '종료(y/n) ?' 물음에 'y'를 입력하면 처음부터 다시 실행하고, 'n'을 입력하면 프로그램을 종료한다.

### [문제 5] 압축코드생성

다음 처리 조건에 따라 압축된 코드를 생성하는 프로그램을 작성하시오.

<처리조건>

- (1) [그림 1]은 압축 부호화 방법을 단계별로 나타내고 있다.
- ① 문자열 'AAAAAABBBBCDDRR'이 주어지면 각 문자의 빈도수에 따라서 [단계1]과 같은 리스트를 만든다(사각형 안의 숫자는 괄호 안의 문자에 해당하는 빈도수를 나타낸다).
- ② 가장 빈도수가 적은 두 문자를 선택하여 [단계2]와 같이 빈도수의 합(원 내부의 숫자는 빈도수의 합을 나타낸다)을 상위의 값으로 하는 부분 이진트리를 만든다. 하위의 두 문자 중에서 빈도수가 적은 것은 좌측으로, 큰 것은 우측으로 놓는다.
- ③ 하위로 내려간 문자를 제외한 나머지 문자의 빈도수와 상위의 값 중에서 가장 작은 두 개를 선택하고 [단계3], [단계4]와 같이 부분 이진트리를 구성한다. 문자의 빈도수와 상위의 값이 같은 경우에는 문자를 먼저 선택한다.
- ④ [단계5]와 같이 하나의 이진트리를 구성한 후, [단계6]과 같이 좌측 경로에는 0, 우측 경로에는 1을 각각 할당한다.
- ⑤ 최상위로부터 하위의 각 문자에 이르는 전체 경로를 이진트리에 할당된 번호로 표현하면 다음과 같이 압축된 이진코드가 구성된다.

A = 0      B = 111      C = 100      D = 101      R = 110

[그림 1] 압축 부호화 방법

|  |  |
|--|--|
| <p>[단계1]    6(A) 3(B) 1(C) 2(D) 2(R)</p>   | <p>[단계2]    6(A) 3(B) 3    2(R)<br/>                      / \<br/>                      1(C) 2(D)</p>  |
| <p>[단계3]    6(A) 5                    3<br/>                      / \                    / \<br/>                      2(R) 3(B) 1(C) 2(D)</p>   | <p>[단계4]    6(A)                    8<br/>                      / \                    / \<br/>                      3                        5<br/>                      / \                    / \<br/>                      1(C) 2(D) 2(R) 3(B)</p>   |
| <p>[단계5]                    14<br/>                      / \                    / \<br/>                      6(A)                    8<br/>                      / \                    / \<br/>                      3                        5<br/>                      / \                    / \<br/>                      1(C) 2(D) 2(R) 3(B)</p> | <p>[단계6]                    14<br/>                      0/                    \1<br/>                      6(A)                    8<br/>                      / \                    / \<br/>                      0/                    \1<br/>                      3                        5<br/>                      / \                    / \<br/>                      0/                    \1    0/                    \1<br/>                      1(C) 2(D) 2(R) 3(B)</p> |

- (2) 프로그램을 실행시키면 <보기 1>의 <1>과 같이 압축코드를 구성하는데 필요한 임의의 문자열을 입력받는다. 입력되는 문자는 영문 대문자와 공백문자만 허용한다(최대 길이는 80 자로 제한함).

<보기 1> 실행 화면

```

<1> 문자열 입력 (80자 이내)
AAAAAABBBBCDDRR

<2> 압축코드표
A | 0          B | 111          C | 100          D | 101          E |
F |           G |           H |           I |           J |
K |           L |           M |           N |           O |
P |           Q |           R | 110          S |           T |
U |           V |           W |           X |           Y |
Z |           sp |

<3> 압축된 문자열
00000011111111111001011011101110

<4> 압축률 ==> 57 퍼센트

<*> 계속?(Y/N) _
  
```

- (3) (1)의 방법으로 구성된 압축코드를 <보기 1>의 <2>와 같은 형식의 코드표로 출력한다. 이때, 입력된 문자열에 포함되지 않은 문자는 코드란을 비워둔다. 문자열에 공백문자가 포함된 경우는 'sp |' 다음에 코드를 출력한다.
- (4) 압축코드표를 이용하여 <보기 1>의 <1>에서 입력된 문자열을 압축하여 <3>과 같이 이진코드로 출력하고, <4>와 같이 압축률을 표시한다.
- (5) 압축률을 구하는 방법은 다음과 같다(문자열의 크기는 이진코드의 비트수를 기준으로 하며, 압축전 문자열의 각 문자는 5 비트로 구성된 코드라고 가정한다.).

$$\text{압축률} = \frac{(\text{압축전 문자열의 크기} - \text{압축후 문자열의 크기}) \times 100}{\text{압축전 문자열의 크기}} \quad (\text{퍼센트})$$

- (6) '계속?(y/n)' 물음에 'y'를 입력하면 <보기 2>와 같이 새로운 문자열을 입력받아 다시 실행하며, 'n'을 입력하면 프로그램을 종료한다.

<보기 2> 새로운 실행 화면

```

<1> 문자열 입력 (80자 이내)
LOVE MEANS YOU DO NOT HAVE TO SAY YOUR SORRY EVER

<2> 압축코드표
A | 1000          B |           C |           D | 111110          E | 000
F |           G |           H | 00100          I |           J |
K |           L | 00101          M | 111111          N | 11110          O | 101
P |           Q |           R | 1100          S | 0011          T | 11100
U | 11101          V | 1001          W |           X |           Y | 1101
Z |           sp | 01

<3> 압축된 문자열
0010110110010000111111100010001111000110111011011110101111101011111010111101011110
001001001000100100001111001010100111000110101110110111101110001001110111001100
11010100010010001100

<4> 압축률 ==> 28 퍼센트

<*> 계속?(Y/N) _
  
```