

프로그래밍기초

Lab 3 (2×10=20점)

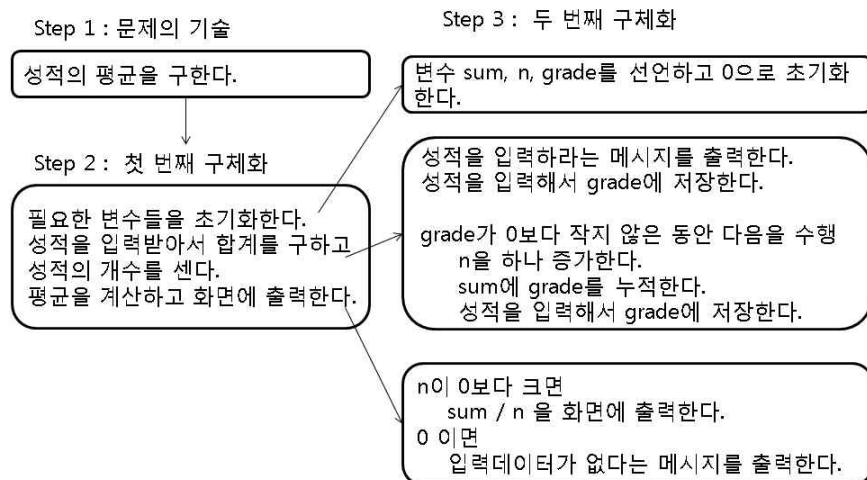
2021.6.24.

■ 프로그래밍 문제

1. [알고리즘 기술 후 구현하기] 사용자로부터 임의의 개수의 성적을 받아서 평균을 계산한 후에 출력하는 프로그램을 작성하시오. 센티널로는 음수 값을 사용한다. 즉 성적이 음수로 입력되면 반복을 중단한다. ※ 아래 주어진 알고리즘 대로 구현할 것

▶ 프로그램 설명

알고리즘의 기술



※ 센티널(sentinel)

데이터 입력의 끝을 알리는데 사용되는 값을 센티널 또는 보초값이라고 한다. 센티널은 일반적인 데이터값에서는 절대 등장할 수 없는 값으로 선택하는 것이 좋다.

▶ 실행결과

```
C:\Windows\System32\cmd.exe
성적입력을 마치려면 음수를 입력하시오.
성적을 입력하시오
-1
입력된 성적이 없습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\System32\cmd.exe
성적입력을 마치려면 음수를 입력하시오.
성적을 입력하시오
10
10
10
-1
3명의 평균은 10.000000입니다.
계속하려면 아무 키나 누르십시오 . . .
```

2. 다음과 같은 메뉴를 출력하고 사용자 입력을 받는 프로그램을 작성해 보자. 입력된 값이 1 부터 3 사이가 아니라면 메뉴를 다시 표시하여 사용자가 새 값을 선택하도록 요청한다.

※ 힌트

do while문은 반복문의 몸체부분을 항상 한 번 이상 실행하므로 메뉴를 출력하고 사용자로부터 특정 선택사항을 입력받기 위한 목적에 유용하다.

3. 글자 알아맞히기 게임을 작성해 보자. 프로그램은 하나의 특정한 글자를 저장하고 있고 사용자는 질문을 통하여 그 글자를 알아맞히려고 노력한다. 사용자가 답을 제시하면 프로그램은 자신이 저장한 글자와 비교하여, 제시된 글자의 아스키 코드가 더 큰지 작은지 만을 알려준다. 프로그램은 사용자가 정확히 글자를 알아맞힐 때까지 반복한다. 사용자가 알아맞히면 몇 번 만에 알아맞혔는지 출력해 준다.

※ 힌트

do while문과 if문을 사용하라. do while문을 사용하여 사용자가 정답을 입력할 때까지 (즉, 입력 값이 정답이 아닌 동안) 반복한다. 매 반복 시 입력 값과 정답사이의 비교를 통해 “입력한 문자의 아스키 코드 값이 더 큼니다” 혹은 “입력한 문자의 아스키 코드 값이 더 작습니다”라는 정보를 출력한다.

4. 정수 1개를 입력받고 입력받은 수가 몇 자리수로 이루어져 있는지와 각 자릿수의 숫자들의 합을 구해서 출력하는 프로그램을 작성하시오.

▶ 프로그램 설명

- while을 사용한다.
- 나누기(/) 및 나머지(%) 연산자를 이용한다.

▶ 실행결과

```
정수1개를 입력하세요 :
12345

입력한 수의 자릿수 5이고
각 자릿수의 합은 15입니다.
```

5. 사용자로부터 하나의 양의 정수를 입력받은 후, 연산자 '+', '-', '*', 또는 '/'를 입력받고, 사용자로부터 또 다른 하나의 양의 정수를 입력받아 그 연산을 수행한 결과를 화면에 출력하는 프로그램을 작성하시오.

▶ 프로그램 설명

- 연산자가 '+'이면 두 정수를 더하기하며, '-'이면 빼기를, '*'이면 곱하기를, '/'이면 실수형 나누기 연산을 수행한다.
- "1번 더 수행하겠습니까?"라는 물음에 Y 또는 y를 입력하면 1번 더 수행하며 N 또는 n을 입력하면 프로그램을 종료해야 한다.

▶ 실행결과

```
정수를 입력하세요 : 32
연산자를 입력하세요 : *
정수를 입력하세요 : 4
>>> 32 * 4 = 128
1번 더 수행 하시겠습니까? (Y/N) : Y

정수를 입력하세요 : 30
연산자를 입력하세요 : /
정수를 입력하세요 : 4
>>> 30 / 4 = 7.50
1번 더 수행 하시겠습니까? (Y/N) : Y

정수를 입력하세요 : 25
연산자를 입력하세요 : /
정수를 입력하세요 : 0
>>> Error
1번 더 수행 하시겠습니까? (Y/N) : N

프로그램을 종료합니다.
```

6. 일련의 정수를 읽어 다음의 정보를 출력하는 프로그램을 작성하시오.

- (1) 입력 데이터의 개수
- (2) 합과 평균
- (3) 최대수와 최소수
- (4) 20보다 작은 수가 있는가? (true/false)
- (5) 모든 수가 10에서 90사이인가? (true/false)

▶ 프로그램 설명

- 사용자로부터 연속해서 정수를 입력받는다.
- 사용자는 모든 정수를 입력하였으면 마지막을 알리는 수 (99999)를 입력한다.

▶ 실행결과

```
정수 데이터를 입력하세요 (종료: 99999);  
3  
81  
-999  
101  
500  
79  
99999
```

```
입력한 데이터의 개수:          6  
총합:                        -235  
평균:                        -39.17  
최소 값:                      -999  
최대 값:                      500  
20보다 작은수가 있는가 ?    true  
모든 수가 10에 90사이의 정수인가 ? false
```

7. 사용자로부터 정수 n 을 입력받고 $1! + 2! + 3! + \dots + (n-1)! + n!$ 을 계산해서 출력하는 프로그램을 작성하시오.

▶ 프로그램 설명

- $1! + 2! + 3! + \dots + (n-1)! + n!$ 은 재귀호출이 아닌 반복문을 사용해서 구현해야 한다.
- 프로그램은 사용자가 음수를 입력할 때까지 반복하여 수행되어야 한다.

▶ 실행결과

```
정수 1개를 입력하세요: 5
```

```
1! = 1
```

```
2! = 2
```

```
3! = 6
```

```
4! = 24
```

```
5! = 120
```

```
-----
```

```
total = 153
```

```
정수 1개를 입력하세요: 9
```

```
1! = 1
```

```
2! = 2
```

```
3! = 6
```

```
4! = 24
```

```
5! = 120
```

```
6! = 7206
```

```
7! = 5040
```

```
8! = 40320
```

```
9! = 362880
```

```
-----
```

```
total = 409113
```

```
정수 1개를 입력하세요: -7
```

```
프로그램을 종료합니다.
```

8. 입력한 정수의 하위 8비트를 추출 후 왼쪽으로 4비트 이동한 값이 178보다 작으면 “178보다 작습니다.” 크면 “178보다 큼니다.” 같으면 “178과 같습니다.”를 출력하는 프로그램을 작성하라.

<실행 예>

정수 입력: __

최하위 8비트를 왼쪽으로 4비트 이동한 값이 178 보다 작습니다.

정수 입력: __

최하위 8비트를 왼쪽으로 4비트 이동한 값이 178 보다 큼니다.

정수 입력: __

최하위 8비트를 왼쪽으로 4비트 이동한 값이 178과 같습니다.

9. 사용자가 입력한 정수의 최하위로부터 13~16번째 비트에 대해 (1) 0으로 지정한 수, (2) 1로 지정한 수, (3) 0은 1로 1은 0으로 바꾼 수를 출력하라.

※ 힌트: 13~16번째 비트가 1인 **비트마스크**를 만드는 방법

0000 0000	0000 0000	1111 0000	0000 0000
-----------	-----------	-----------	-----------

<case 1>

```
int mask = 0xf000; // 0x0000f000
```

<case 2>

```
int mask = 0;
```

```
mask |= 1 << (13-1);
```

```
mask |= 1 << (14-1);
```

```
mask |= 1 << (15-1);
```

```
mask |= 1 << (16-1);
```

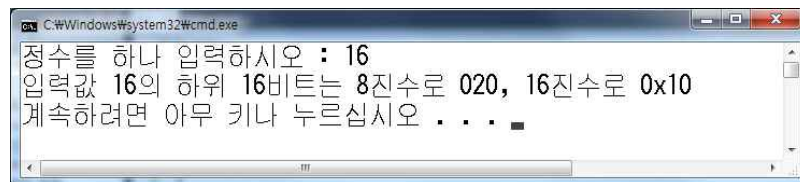
<case 3>

```
int mask = 0, i;
```

```
for ( i = 13; i<= 16; i++ )
```

```
    mask |= 1 << (i-1);
```

10. 적당한 비트마스크를 이용하여 표준입력으로 받은 정수의 하위 2바이트의 값을 8진수와 16진수로 출력하는 프로그램을 작성하시오.



※ 힌트

int형 변수에 대해 하위 16비트 정보를 얻어오기 위한 마스크

$1111_{(2)} \Leftrightarrow f_{(16)}$

$00000000\ 00000000\ 11111111\ 11111111_{(2)} \Leftrightarrow ffff_{(16)}$

```
int mask = 0xffff;
```

[제출]

- 마감 : 수업 당일 자정

- 형식 :

프로그래밍 문제 -> 1.c~10.c (소스 파일 첫 줄에 학번 이름 주석 처리)

모두 “학번.zip”으로 압축하여 업로드

- 제출처 : lms > 과제 게시판

- 주의 : 소스 복사의 경우 0점 처리함