

LAB 7

■ LAB7_1 (do ~ while) (난이도 중)

1 부터 100 까지 합을 구하는 코드를 작성하시오.

□ 힌트

```
초기식;
do {
    ...

    증감식;
}(조건식);
```

실행에 1

1 부터 100 까지 합 = 5050

■ LAB7_2 (do ~ while) (난이도 중하)

입력받은 정수를 1/2 씩 줄여가면서 1 이 될때까지 출력하는 프로그램을 작성하시오

실행에 1: (밀줄은 입력)	실행에 2: (밀줄은 입력)
정수? 100 100 50 25 12 6 3 1	정수? 1 1

■ LAB7_3(감시값 제어 while 문 연습)(난이도 하)

4 직연산자(+, -, *, /)와 연산항을 읽어서 결과를 반복적으로 출력하는 프로그램을 작성하시오.

실행결과 예: (밀줄은 입력)

```
수식을 입력하시오: 6 + 7
6.000000 + 7.000000 = 13.000000
계속 하시겠습니까(Y/N)? y
수식을 입력하시오: 6 * 7
6.000000 * 7.000000 = 42.000000
계속 하시겠습니까(Y/N)? y
수식을 입력하시오: 6 / 7
6.000000 / 7.000000 = 0.857143
계속 하시겠습니까(Y/N)? n
```

□ 힌트 및 주의사항

- 버퍼 비우기

```
int main(void)
{
    //변수 선언
    char yesno = " Y" ;
    while (yesno == 'Y' || yesno == 'y') {
        // 연산자에 따른 연산 수행
        ...

        printf( "%f %c %f = %f\n", x, op, y, result);
        // while(getchar() != '\n' ); // 버퍼비우기
        printf("계속 하시겠습니까(Y/N)? ");
        scanf("%c", &yesno); // 버퍼비우기를 여기서 하려면 %c 앞에 공백 추가
    }
}
```

■ LAB7_4 (while 문 : 감시값 제어 반복문)

여러 개의 성적을 읽어서 그 총점을 계산하는 프로그램을 작성하시오. -1 은 성적의 끝을 나타낸다.

감시값 제어 while (0 또는 여러 번 반복)

// version 1	// version 2
<pre>input 을 읽는다; while (input 이 감시값이 아닌동안) { 반복 부분; input 을 읽는다; }</pre>	<pre>while (1) { input 을 읽는다; if (input 이 감시값이면) break; 반복 부분; }</pre>

□ LAB7_4_1 위에 제시한 감시값 제어문의 형태 중 version1 을 사용하여 프로그래밍한다.

```
#include <stdio.h>
int main(void)
{
    // 변수 선언 및 초기화

    // score 를 읽는다
    while (score != -1)
    {
        // total 계산한다
        // score 를 읽는다
    }
    printf("The total is %d.\n", total);
}
```

실행에 1(밀줄 친 부분은 입력)	실행에 2(밀줄친 부분은 입력)
Enter a score(-1 for exit): 30 Enter a score(-1 for exit): 50 Enter a score(-1 for exit): 30 Enter a score(-1 for exit): -1 The total is 110.	Enter a score(-1 for exit): -1 The total is 0.

□ LAB7_4_2 위에 제시한 감시값 제어문의 형태중 version2 를 사용하여 프로그래밍하라.

■ LAB7_5 (조건문, 반복문, 분기문)

소수는 1 과 자기 자신만으로 나누어지 떨어지는 1 보다 큰 양의 정수이다. 정수를 입력받아 소수인지 검사하는 프로그램을 작성하시오.

실행에 1: (밀줄은 입력)	실행에 2: (밀줄은 입력)
양의 정수? 7 7 은 소수이다.	양의 정수? 12 12 는 소수가 아니다.

■ LAB7_6 (rand() 함수)(난이도 하)

단계 1 부터 시작해서 단계 4 까지 차례로 프로그램을 수정하면서 진행하도록 한다.

단계 1: 수업시간에 다른 난수 예제 1 을 실행시켜보라. 여러 번 실행시켜도 같은 결과가 나옴을 확인하라.

단계 2: 위의 프로그램을 난수 예제 2 로 수정하여 실행시켜보라. 씨드를 바꾸어서 다른 결과가 나옴을 확인하라. 같은 씨드를 넣었을때는 같은 결과가 나옴을 확인하라.

단계 3: 위의 프로그램을 난수 예제 3 으로 수정하여 실행시켜보라. 여러 번 실행시켜보고 매번 다른 결과가 나옴을 확인하라.

단계 4: 위의 프로그램을 0 부터 99 까지의 난수를 발생시키도록 수정하라.

실행에 1:	실행에 2:
난수 출력: 27	난수 출력: 28
난수 출력: 69	난수 출력: 90
난수 출력: 54	난수 출력: 87
난수 출력: 66	난수 출력: 65

(주의: 실행시 마다 각각 다른 난수를 발생시키므로 위에 주어진 실행결과와 여러분의 결과는 같지 않습니다.)

■ LAB7_7 (rand() 함수를 이용한 산술 문제 내기)(난이도 하)

0 에서 99 까지의 수자를 임의로 발생시켜서, 다음과 같은 덧셈 문제를 출력하는 프로그램을 작성하시오.

[주의사항]

- stdlib.h 와 time.h 를 포함시키고 srand(time(NULL))을 사용한다.
- 매번 실행시 다른 난수가 발생되어야 한다.

단계 1 : 문제내기

아래의 실행결과를 같도록 프로그램을 작성하라.

실행에 1:
48 + 3 =

단계 2 : 답 평가하기

위의 프로그램을 수정(발전)시켜서 각 덧셈문제에 대해서 사용자가 답을 입력하면, 답이 맞았는지 틀렸는지를 화면에 출력하는 프로그램을 작성한다.

실행에 1: (밀줄은 입력)	실행에 2: (밀줄은 입력)
65 + 71 = 136 Your answer is right	48 + 90 = 123 Your answer is wrong. 138 is the right answer

HW 7

■ HW7_1(do ~ while 문)

0 보다 큰 양의 정수를 입력받아 역순으로 출력하는 프로그램을 작성하시오.

실행에 1: (밀줄은 입력)	실행에 1: (밀줄은 입력)
Enter the number: 12345 5 4 3 2 1	Enter the number: 300 0 0 3

■ HW7_2(do ~ while, 숫자 추측게임)

1~99 사이의 정수 중 하나를 컴퓨터가 가지고 있고, 사용자가 그 수를 알아맞히는 프로그램을 작성하시오. 사용자가 알아맞히는데 걸린 시도횟수를 함께 출력한다.

□ 힌트:

```
do {
    사용자로부터 숫자를 guess 로 입력받는다.
    시도횟수를 증가한다.
```

```
    정답이 아닌 경우 힌트를 제시한다.
} while(guess != answer)
```

실행에: (밀줄은 입력)

정답을 추측해 보시오: 78
더 작은 수를 말해봐
정답을 추측해 보시오: 45
더 큰 수를 말해봐
정답을 추측해 보시오: 59
축하합니다. 3 번만에 맞췄네요

■ HW7_3(감시값 제어 while 문)

LAB7_5 를 이용하여 학생수, 총점, 평균, 가장 좋은 점수를 출력하도록 프로그램을 작성하시오.

□ 힌트:

- 먼저 총점, 평균을 계산하여 출력하는 부분을 성공시킨 후, 그다음 최고점수를 찾아 출력하는 부분을 추가하여 실행시킨다.

주의사항: 데이터가 없이 바로 -1 이 입력되는 경우(실행에 2) 아래와 같은 실행예를 내도록 하라.

실행에 1(밀줄 친 부분은 입력) Enter a score(-1 for exit): 30 Enter a score(-1 for exit): 50 Enter a score(-1 for exit): 30 Enter a score(-1 for exit): -1 You' ve entered 3 students. The total is 110. The average is 36.7. The best score is 50.	실행에 2(밀줄친 부분은 입력) Enter a score(-1 for exit): -1 You' ve entered 0 students. There is no data.
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------

■ HW7_4(감시값 제어 while 문)

정수를 2개 입력받아 두 수의 최대 공약수를 구하는 프로그램을 작성하시오.

□ 힌트: 유클리드 호제법

- ① 두 수 가운데 큰 수를 x, 작은 수를 y 라 한다.
- ② y 가 0 이면 최대 공약수는 x 와 같고 알고리즘을 종료한다.
- ③ $r \leftarrow x \% y$
- ④ $x \leftarrow y$
- ⑤ $y \leftarrow r$
- ⑥ 단계 ②로 되돌아간다.

실행에 1: (밀줄은 입력)

두 개의 정수를 입력하시오(큰수, 작은수): 18 12
GCD= 6

실행에 2: (밀줄은 입력)

두 개의 정수를 입력하시오(큰수, 작은수): 5 7
GCD= 1

■ HW7_5(반복문, 조건문, 분기문)

소문자를 입력받아 대문자로 변경하여 출력하는 프로그램을 작성하시오. 단, 대문자 'Q' 를 입력하면 프로그램을 종료하도록 하며, 소문자와 대문자 'Q' 가 아니면 다시 입력받는다.

- 힌트
- 무한 루프를 사용하고 continue 와 break 를 이용한다.

실행에 1: (밀줄은 입력)	실행에 2: (밀줄은 입력)
소문자를 입력: a 변환된 대문자는 A 소문자를 입력: t 변환된 대문자는 T 소문자를 입력: z 변환된 대문자는 Z 소문자를 입력: Q	소문자를 입력: 1 소문자를 입력: 2 소문자를 입력: 3 소문자를 입력: Q

■ HW7_6(반복문, 조건문, 분기문)

정수 N을 입력받아 1~N 사이의 소수를 모두 구하여 출력하는 프로그램을 작성하시오. 소수는 한 줄에 10 개씩 출력한다.

- 힌트:
- 한 줄에 10 개씩은 10 으로 나눈 나머지 이용

실행에 1: (밀줄은 입력)	실행에 2: (밀줄은 입력)
양의 정수? 100 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97	양의 정수? 12 2 3 5 7 11

■ HW7_7 (rand(), srand()함수, 감시값 제어 do while 문)(난이도 중) 덧셈문제 연습하기

LAB7_7 을 참고하여 사용자가 원하는 만큼 덧셈문제를 연습할 수 있도록 프로그램을 작성하시오

- 힌트 및 주의점
- 문자를 읽기 전에 while(getchar() != 'Wn');을 쓰는 것을 잊지말라!

실행에 1: (밀줄은 입력)	실행에 2: (밀줄은 입력)
9 + 0 = 9 Your answer is right continue?(y/n): y 8 + 1 = 9 Your answer is right continue?(y/n): y 5 + 4 = 7 Your answer is wrong. 9 is the right answer continue?(y/n): y 8 + 9 = 2 Your answer is wrong. 17 is the right answer continue?(y/n): n	6 + 8 = 14 Your answer is right continue?(y/n): n