

중간시험

[총 20점: 6 문항]
(Q1: 5점, Q2~Q6: 각 3점)

컴프실 2022(Spring)
cyclops@ajou.ac.kr

제출물

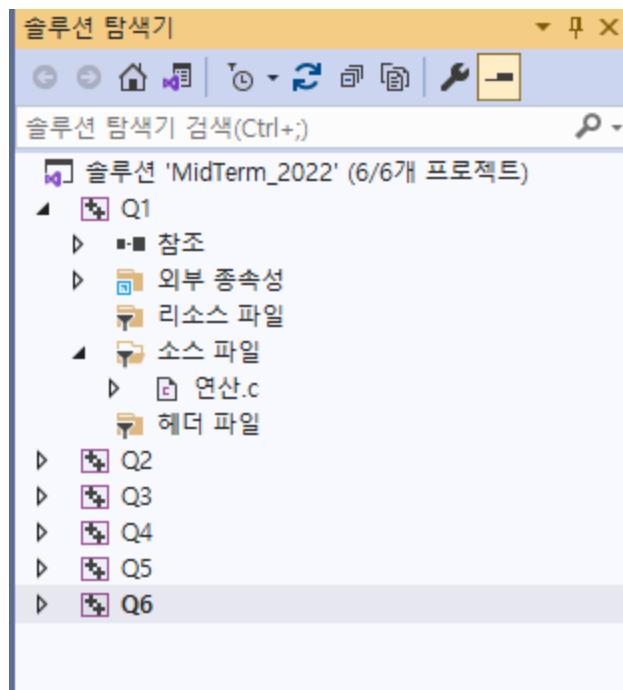
- 각 문항의 해답을 소스코드 포함하여 하나의 문서 파일로 (여러 개의 파일인 경우 하나의 압축 파일로) 제출합니다. 프로그램을 VS C로 구현한 경우, 아래 예와 같이, 각 문항의 프로그램이 개별 Project로 등록된 하나의 솔루션으로 구성하고, 이 솔루션 폴더를 압축한 파일을 제출하여도 됩니다..

※ 프로그램 작성시, 적절한 주석은 필수입니다.

※ 실행되지 않는 코드는 0점 처리됩니다.

※ 각 문항의 해답을 같은 (동일) 솔루션 내의 개별 Project로 추가하여 구현할 수 있습니다. 각 문항 별 해는 main 함수를 가진 프로그램으로 작성되어, 선택적으로 실행시킬 수 있습니다. 이 경우도, MS VS의 **솔루션 폴더를 압축**하여 제출하면 됩니다. (솔루션을 생성할 때, 어떤 폴더를 사용하는지 확인하고, 제출 시 착오 없길 주의하세요.)

※ '.sln' 파일은 솔루션 관련 정보를 가진, VS가 관리하는 파일입니다. 이 파일만 제출하는 실수를 하지 마세요.



Q1. 연산 [5점]

아래 프로그램을 실행하여 출력 결과를 확인하고, 각 **printf** 함수 호출문 아래에 printf 함수 호출문을 추가하여 위 출력 결과에 대한 이유도 같이 출력할 수 있도록 변경하세요.

* 참고 사항: 실수 형 float-type의 크기는 4 bytes (32 bits) 이다.

```
#include <stdio.h>

void main() {
    char ch;
    int a, b, c;
    int temp = 0;
    float f1, f2;
    a = b = c = 100;
    temp = a++ + ++b;
    temp += a-- + c++;
    printf("a=%d, b=%d, c=%d, temp=%d\n", a, b, c, temp);
    a = 10;
    printf("15<%d<20 == %d\n", a, 15<a<20);
    printf("!!%d == %d\n", a, !!a);
    ch = 'a';
    temp = 'A' <= ch && ch <= 'z';
    printf("temp = %d\n", temp);
    f1 = 5e+20;
    f2 = 0.1;
    printf("f1<f1+f2 == %d\n", f1<f1+f2 );
}
```

Q2. 윤년 판별 [3점]

2000년부터 2999년까지, 2000년부터 3999년까지 윤년이 몇 회가 있는지 출력하고, 그 기간 동안 (1000년, 2000년) 동안 윤년이 몇 % 정도 차지하고 있는지도 소수점 두자리까지 출력하는, 프로그램을 C 언어로 작성 하세요..

출력 예

```
*** Leap Year ***  
2000년에서 2999년까지 윤년은 xxx번 있고, xx.xx% 입니다.  
2000년에서 3999년까지 윤년은 xxx번 있고, xx.xx% 입니다
```

Q3. 직사각형 좌표 [3점]

직사각형을 만드는 데 필요한 4개의 점 중 3개의 좌표가 주어질 때, 나머지 한 점의 좌표를 구하고, 이 점의 좌표를 출력하는 프로그램을 C 언어로 작성하시오. 단, 직사각형의 **각 변은 x축, y축에 평행**하며, **반드시 직사각형을 만들 수 있는 경우만 입력**으로 주어집니다.

제한사항

- 세개의 점 **v1, v2, v3**는 요소 수가 2인 1차원 배열로 첫번째 요소는 x축, 두번째 요소는 y축 좌표를 표시한다.
- 위 세 점의 좌표는 (x축 좌표, y축 좌표) 순으로 입력 받습니다. 좌표 값은 1 이상 자연수입니다.
- 직사각형을 만드는 데 필요한 나머지 한 점, v4의 좌표를 (x축 좌표, y축 좌표)순으로 담아 출력합니다.

※ 제어문 (선택문/반복문) 및 3항 연산자는 사용하지 못합니다.

입출력 예

```
*** Rectangle ****
```

```
좌표 입력 (x y): 3 4
```

```
좌표 입력 (x y): 1 4
```

```
좌표 입력 (x y): 3 10
```

```
주어진 세개의 좌표: (3, 4), (1, 4), (3, 10)
```

```
나머지 네번째 좌표: (1, 10)
```

```
*** Rectangle ****
```

```
좌표 입력 (x y): 1 1
```

```
좌표 입력 (x y): 2 2
```

```
좌표 입력 (x y): 1 2
```

```
주어진 세개의 좌표: (1, 1), (2, 2), (1, 2)
```

```
나머지 네번째 좌표: (2, 1)
```

※ 세 점이 [1, 4], [3, 4], [3, 10]
위치에 있을 때, [1, 10]에 점이
위치하면 직사각형이 됨..

※ 세 점이 [1, 1], [2, 2], [1, 2]
위치에 있을 때, [2, 1]에 점이
위치하면 직사각형이 됨.

Q4. 다이아몬드 별 찍기 [3점]

양의 정수 N을 입력 받아, 문자 '*'-별을 사용하여 (2N+1) 행의 다이아몬드 별 찍기 프로그램을 작성합니다.

- ① 1에서 9까지 범위 내의 양의 정수 N을 입력 받아, 문자 '*'을 사용하여 (2N+1) 행의 다이아몬드 형으로 아래 예와 같이 출력합니다. (단, 중앙 행은 '-'(dash)로 채우고, 각 행의 중앙 열에는 '|' (bar)를 출력합니다.)
- ② (1) 항을 반복하여 실행하며, N이 0일 경우 종료합니다.

입/출력 예

```
*** Diamond ***
* Enter N (1 ~ 9): 8
      |
    * | *
  ** | **
 *** | ***
**** | ****
***** | *****
***** | *****
***** | *****
***** | *****
-----|-----
***** | *****
***** | *****
***** | *****
***** | *****
    * | *
      |
* Enter N (1 ~ 9): 0
Bye!!!
```

참고

```
*** Diamond ***
* Enter N (1 ~ 9): 9
[ 9, 1]          |
[ 8, 3]          * | *
[ 7, 5]          ** | **
[ 6, 7]          *** | ***
[ 5, 9]          **** | ****
[ 4, 11]         ***** | *****
[ 3, 13]         **** | ****
[ 2, 15]         ***** | *****
[ 1, 17]         **** | ****
[ 0, 19] -----|-----
[ 1, 17]         **** | ****
[ 2, 15]         ***** | *****
[ 3, 13]         **** | ****
[ 4, 11]         ***** | *****
[ 5, 9]          **** | ****
[ 6, 7]          *** | ***
[ 7, 5]          ** | **
[ 8, 3]          * | *
[ 9, 1]          |
* Enter N (1 ~ 9):
```

Q5. DDN IP (IPv4) 주소를 2진수로 표현 [3점]

- **Domain Name System (DNS)**은 사람들이 기억하기 쉬운 도메인 이름을 **32비트 IP (IPv4) 주소**로 변환하는 시스템을 가리킵니다.
- IPv4의 주소, 32비트의 각 바이트(8 비트)를 십진 숫자로 변환한 후 이를 202.30.0.11과 같이 숫자와 점으로 표현하는 방식을 **Dot-decimal notation(DDN)**이라고 하며, IP 주소는 세 개의 점('.')으로 구별하여 표현됩니다.
- 32비트 IP 주소를 DDN 형태로 입력 받아, 2진수-비트 형태로 출력하는 C 프로그램을 작성합니다.
- 주의사항
 - ① 프로그램 시작에 친절하게 "Enter a DDN (0.0.0.0 for exit): "를 출력해 반복 입력할 수 있도록 하고, DDN이 '0.0.0.0'으로 입력되면 프로그램을 종료합니다.
 - ② 입력 처리 후 결과 출력은 <입/출력 예>를 따릅니다.
 - ③ 프로그램 종료 시, 친절한 문구를 출력하고 끝냅니다.

입/출력 예

```
*** DDN IP ****
* Enter a DDN (0.0.0.0 for exit) : 1.2.4.8
    00000001.00000010.00000100.00001000
* Enter a DDN (0.0.0.0 for exit) : 128.64.1.255
    10000000.01000000.00000001.11111111
* Enter a DDN (0.0.0.0 for exit) : 0.0.0.0
Bye!!!
```

Q6. 유리구슬 [3점]

N 층 건물에서 유리구슬을 떨어뜨려도 깨어지지 않는 최상위층을, 2개의 똑 같은 구슬로 찾는 최적의 방법을 C 프로그램으로 구현합니다.

- ① 프로그램 시작에 친절하게 "건물 층수 N (자연수) : "를 출력해 반복 입력할 수 있도록 하고, 0 이하의 수가 입력되면 프로그램을 종료합니다.
- ② N 층 건물에서 구슬이 깨어지지 않는 최상위층을 찾을 수 최적의 횟수, X를 찾아 출력합니다.
- ③ 위의 과정을 반복합니다.

※ "구슬을 몇 번 떨어뜨려 구슬이 깨어지지 않는 최상위층을 찾을 수 있을까?" 하는 문제입니다.. 즉, 구한 횟수가 X라면, X번 이내에 구슬을 떨어뜨려 깨어지지 않는 최상위층을, 모든 경우에 찾을 수 있다는 것입니다. (중간에 두 구슬이 다 깨어지면 더 이상 진행할 수 없어, 문제를 해결할 수 없게 됩니다.)



참고

```
*** Twin Marbles ***
* 건물 층수 N (자연수): 200
  최고층 찾는 최적의 횟수: 20
* 건물 층수 N (자연수): 1000
  최고층 찾는 최적의 횟수: 45
* 건물 층수 N (자연수): 2000
  최고층 찾는 최적의 횟수: 63
* 건물 층수 N (자연수): 3000
  최고층 찾는 최적의 횟수: 77
* 건물 층수 N (자연수): 0
Bye!!!
```