

# 과제#1

[총 15점: 5 문항]  
(Q1~Q5: 각 문항 당 3점)

컴프실 2022 (Spring)  
cyclops@ajou.ac.kr

# 제출물

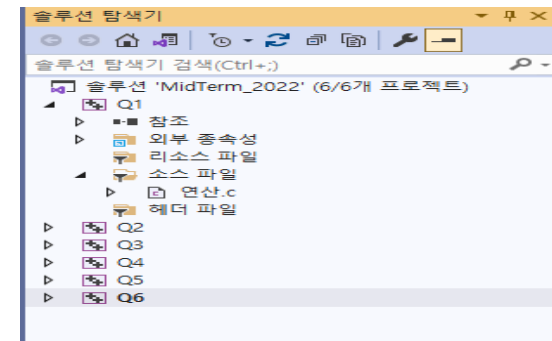
- 각 문항의 해답을 **소스코드 포함하여 하나의 문서 파일**로 (여러 개의 파일인 경우 하나의 압축 파일로) 제출합니다. 프로그램을 VS C로 구현한 경우, 아래 예와 같이, 각 문항의 프로그램이 개별 Project로 등록된 하나의 솔루션으로 구성하고, 이 솔루션 폴더를 압축한 파일을 제출하여도 됩니다..

※ 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] 다트게임 hit 확률 Simulation <3점>

- Dart 게임에서는 Dart는 임의로 코너의 좌표가  $(-1, -1)$ ,  $(1, 1)$ 인 정사각형에 던져지는데 중심이  $(0,0)$ 이고 반지름이 1인 원안에 Dart가 던져지면 hit한 것으로 간주되고 그렇지 않으면 miss라고 가정한다. 본 과제는 실제 dart를 던져 보지 않고 컴퓨터 시뮬레이션을 통하여 테스트 횟수가 많아지면 hit/try 값이 특정 값에 수렴하는 것을 증명할 수 있다. try 횟수  $n$  ( $1 \leq n < 10000$ )을 입력 받아 결과 값 hit/try 값을 출력하는 프로그램을 완성하라. 참고로 점  $(a, b)$ 를 중심으로 하고 반지름이  $r$ 인 원의 방정식은  $(x - a)^2 + (y - b)^2 = r^2$ 로 나타낸다.  
(※ 난수 발생 함수 사용)
- 주의사항
  - ① 프로그램 시작에 친절하게 "Enter a trial number (1 ~ 10000)"을 출력해 반복하여 입력을 받을 수 있도록 하고, "0"이 입력되면 프로그램을 종료한다.
  - ② 입력 처리 후 결과 출력은 <입력 처리 예시> 를 따른다.
  - ③ 프로그램을 종료하는 경우 친절한 문구를 출력해 줄 것

## Q1 <출력 예시>

```
Enter a trial number (1 ~ 10000): 100
** hit/trial is : 0.yyyyyyy
Enter a trial number (1 ~ 10000) : 100000
** Trial number is out of range.
Enter a trial number (1 ~ 10000): 0
Bye!!!
```

## [Q2] Anagram - Permutation <3점>

- 애너그램은 단어나 문장을 구성하고 있는 문자의 순서를 바꾸어 다른 단어나 문장을 만드는 놀이이다. 단어의 경우 주어진 글자들을 재배열, Permutation-순열로 새로운 단어를 만든다.
  - ※영어 예로, live → Evil, Garden → Danger, Heart → Earth
  - ※한자 예로, 民音訓正 (백성의 소리를 새김이 마땅하다) → 訓民正音 (훈민정음)
- 5개 이하의 문자의 개수, N을 입력받아, 영어 대문자 'A' 부터 N 번째 까지의 문자들로 만들 수 있는 모든 애너그램을 출력하는 C 프로그램을 작성하라.
  - ① 프로그램 시작에 친절하게 "문자 수 (5 이하): "를 출력해 반복 입력할 수 있도록 하고, 숫자 '0'이 입력되면 프로그램을 종료한다.
  - ② 입력 처리 후 결과 출력은 <입/출력 예시 1> ~ <입/출력 예시 2>를 따른다.

## Q2 <입/출력 예시 1>

문자 수 (5 이하): 2

1 AB

2 BA

문자 수 (5 이하): 1

1 A

문자 수 (5 이하): 3

1 ABC

2 ACB

3 BAC

4 BCA

5 CBA

6 CAB

문자 수 (5 이하): 4

1 ABCD

2 ABDC

3 ACBD

4 ACDB

5 ADCB

6 ADBC

7 BACD

8 BADC

9 BCAD

10 BCDA

11 BDCA

12 BDAC

13 CBAD

14 CBDA

15 CABD

16 CADB

17 CDAB

18 CDBA

19 DBCA

20 DBAC

21 DCBA

22 DCAB

23 DACB

24 DABC

문자 수 (5 이하):

## Q2 <입/출력 예시 2>

문자 수 (5 이하): 0

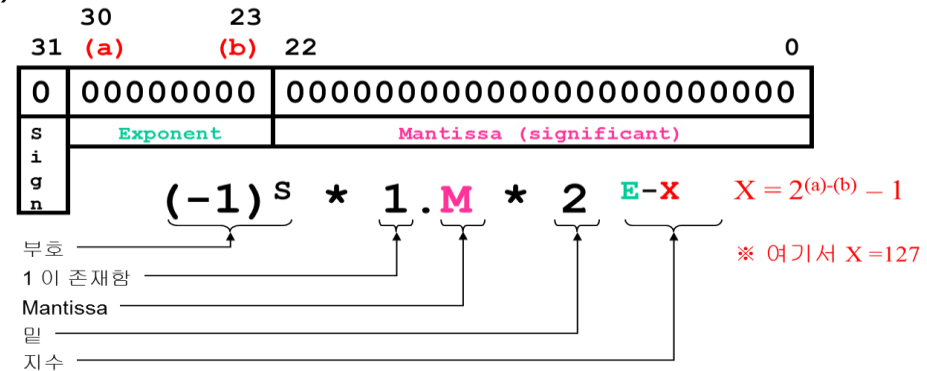
Bye!!!

계속하려면 아무 키나

누르십시오 ...

## [Q3] Float Type 실수 이진수로 출력 <3점>

- float type의 내부 저장 32-비트 열에 따른 값 설정 방법 (IEEE 754)은 아래 그림과 같습니다. (강의자료 참조)



- 실수 형의 숫자를 입력 받아, 위의 내부 저장 형태의 비트 열을 출력하는 C 프로그램을 작성합니다.
  - 프로그램 시작에 친절하게 "Enter a float-type real number (0 to quit):"를 출력해 반복 입력할 수 있도록 하고, 숫자 '0'이 입력되면 <입/출력 예시 3>과 같이 출력하고, 프로그램을 종료합니다.
  - 입력 처리 후 결과 출력은 <입/출력 예시 1> ~ <입/출력 예시 3>를 따른다.
  - 이진수는 4개의 숫자를 묶어, 사이에 공란을 두어 출력합니다.

※ Float type의 최대 상수는 표준 라이브러리 <float.h> 내 **FLT\_MAX** 로 정의되어 있음.

### Q3 <입/출력 예시 1>

```
Enter a float-type real number (0 to exit): 0.1
** Binary representation = 0011 1101 1100 1100 1100 1100 1100 1101
    Sign                = 0
    Exponent             = 0111 1011
    Mantissa             = 1001 1001 1001 1001 1001 101
Enter a float-type real number : Enter a float-type real number (0 for exit):
```

### Q3 <입/출력 예시 2>

```
Enter a float-type real number (0 for exit): -123.8125
** Binary representation = 1100 0010 1111 0111 1010 0000 0000 0000
    Sign                = 1
    Exponent             = 1000 0101
    Mantissa             = 1110 1111 0100 0000 0000 000
Enter a float-type real number (0 for exit):
```

### Q3 <입/출력 예시 3>

```
Enter a float-type real number (0 for exit): 0
** Binary representation = 0000 0000 0000 0000 0000 0000 0000 0000
    Sign                = 0
    Exponent             = 0000 0000
    Mantissa             = 0000 0000 0000 0000 0000 000
NAX of float number = 340282346638528859811704183484516925440.000000
** Binary representation = 0111 1111 0111 1111 1111 1111 1111 1111
    Sign                = 0
    Exponent             = 1111 1110
    Mantissa             = 1111 1111 1111 1111 1111 111
Bye!!!
```



## [Q4] 계단 올라가기 <3점>

- N개의 계단이 있다. 한번에 한 계단이나 두 계단을 올라갈 수 있습니다. N번째 계단까지 올라가는 방법이 얼마나 많이 있는지 알아 보는 문제입니다.
- 계단의 개수, **N**을 입력 받아, 올라가는 방법의 수, **X**를 출력합니다.
- 위 과정을 반복하고, 입력된 계단 개수  $N = 0$ 일 경우 종료합니다.

### <입/출력 예시>

```
계단 수 (입력) : 1
오르는 방법 가짓수: 1
계단 수 (입력) : 2
오르는 방법 가짓수: 2
계단 수 (입력) : 10
오르는 방법 가짓수: 89
계단 수 (입력) : 0
Bye!!!
```

## [Q5] 개미수열 <3점>

- 수열, 1, 11, 21, 1211, 111221, 312211, ... 은 '보고 말하기' ([look-and-say](#)) 수열입니다 ('개미'수열이라고도 함.). 주어진 첫수 이후, 계속되는 수는, 앞의 수에 어떤 숫자가 몇 개 있는지를, 보고 말하는 것처럼 (영어로) 나타낸 것입니다.

1	1
11	one 1
21	two 1s
1211	one 2 one 1
111221	one 1 one 2 two 1s
312211	three 1s two 2s one 1

- 1986년 수학자 Horton Conway가 발표한 수열로, 그림이나 영상 압축에 사용되는 [Run-Length Encoding](#)과 같은 방식으로 생성되는 수열임.
- 2020년 4월 11일 COVID-19로 사망 (82세)

- 즉, 위와 같이 앞의 숫자를 읽고 표시한 수열입니다. 주어진 첫수로부터 **N** 번째 수까지 차례로 출력하는 프로그램을 작성합니다.
  - 화면에 친절한 메시지를 출력하여, 첫수와 **N**을 입력 받습니다. 첫수와 **N**은 1에서 9까지의 정수이고, 상기와 같이 (설명 제외하고) 한 줄에 한 수씩 출력합니다. (입/출력에 참조)
  - 위 (1) 항을 반복하여 실행하며, 첫수나 **N**이 0일 경우 종료합니다..

## Q5 <입/출력 예시>

```
*** Look & Say Sequence ***
* Enter Base Number [1-9] & Depth [1-9]: 1 5
Level 0: 1
Level 1: 1 1
Level 2: 2 1
Level 3: 1 2 1 1
Level 4: 1 1 1 2 2 1
Level 5: 3 1 2 2 1 1
* Enter Base Number [1-9] & Depth [1-9]: 5 5
Level 0: 5
Level 1: 1 5
Level 2: 1 1 1 5
Level 3: 3 1 1 5
Level 4: 1 3 2 1 1 5
Level 5: 1 1 1 3 1 2 2 1 1 5
* Enter Base Number [1-9] & Depth [1-9]: 3 3
Level 0: 3
Level 1: 1 3
Level 2: 1 1 1 3
Level 3: 3 1 1 3
• Enter Base Number [1-9] & Depth [1-9]: 0 0
Bye!!!
```