

# CSED101. Programming & Problem solving

## Fall, 2015

### Programming Assignment #1

(30 points)

이효련(wowlhr@postech.ac.kr)

■ **Due:** 2015.09.30 23:59

■ **Development Environment.** GNU C Compiler (GCC) and Vi Editor (Editor is optional)

#### ■ 제출물

- **C Code files (\*.c)**

- 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.

- **보고서 파일** (.doc(x) or .hwp) 예) assn1.doc(x) 또는 assn1.hwp

- AssnReadMe.pdf 를 참조하여 작성할 것.

- 리눅스 서버에 접속하는 것부터 시작해서 프로그램 컴파일 및 실행하는 과정까지를 화면 캡처하여 보고서에 포함시키고 간단히 설명 할 것!!

- 명예서약(Honor code): 표지에 다음의 내용을 포함한다. "나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다." 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.

- 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

#### ■ 주의사항

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.

- 모든 문제의 출력 형식은 아래의 예시들과 동일해야 하며, 같지 않을 시는 감점이 된다.

- 각 문제에 제시되어 있는 파일이름으로 제출 할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.

- 컴파일 & 실행이 안되면 무조건 0점 처리된다.

- 하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)

- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 'POSTECH 전자컴퓨터공학부 부정행위 정의'를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)

- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.

## (15 pts) Problem 1: 자신의 이니셜 암호화하기

### (설명)

자신의 이니셜 문자를 입력으로 받아 암호화하는 프로그램을 작성하시오.

- 본인의 이니셜을 입력으로 받으며, 문자를 입력으로 받는 방법은 다음 예시를 활용한다.

```
scanf("%c %c %c", ...)
```

- 암호화에 사용될 한자리의 정수를 입력 받는다.
- 입력한 이니셜의 ASCII 코드를 확인하고, 한자리 정수를 이용해 이니셜을 암호화한다.

### (요구사항)

1. 사용자로부터 이니셜을 입력 받는다.
  - i. 각 좌표는 공백(space)로 구분하여 대·소문자에 상관없이 입력 받는다.
  - ii. 모든 이니셜은 **문자 타입**으로 입력 받는다.
  - iii. 숫자, 특수문자 등과 같이 영어 대·소문자 이외의 입력은 없는 것으로 가정한다.  
(예외처리 필요 없음)
2. 사용자로부터 암호화 정수를 입력 받는다.
  - i. 문자, 두 자리 이상의 정수 등과 같은 입력은 없는 것으로 가정한다. (예외처리 필요 없음)
  - ii. **각 이니셜 문자에서 입력 정수를 뺀 암호화 문자를 얻는다.**
3. 프로그램의 출력
  - i. 입력한 이니셜의 ASCII 코드를 출력한다.
  - ii. 암호화된 문자를 출력한다.

### (주의사항)

- 파일 이름은 **"assn1\_1.c"**로 저장 할 것
- 보고서는 **"assn1.doc"** or **"assn1.hwp"**로 저장 할 것 (보고서는 통합하여 작성)
- 출력은 아래의 "실행예제"와 동일하도록 작성 할 것 (모든 공백은 'space' 사용)
- 아직 수업시간에 다루지 않은 if, for, while 문 등은 사용하지 않는다.

(실행예제) 빨간 밑줄은 사용자가 입력하는 부분에 해당됨.

```
1 programming.postech.ac.kr:2022 x
[wowlhr@programming ~]$ gcc assn1_1.c -o assn1_1.out
[wowlhr@programming ~]$ ./assn1_1.out
Input your initial with white space: H Y O

Input encoding number(1~9): 8

ASCII code of your input: 72 89 79

Encoded output :@ Q G

[wowlhr@programming ~]$
```

## (15 pts) Problem 2: 두 점을 지나는 직선의 방정식과 두 점 사이의 거리 구하기

### (설명)

2차원 상 두 개의 점의 좌표  $P_1(x_1, y_1)$ 과  $P_2(x_2, y_2)$ 를 입력 받아 두 점을 지나는 직선의 방정식과 두 점 사이의 거리를 구하는 프로그램을 작성하시오.

- $P_1$ 과  $P_2$ 를 지나는 직선의 방정식은 다음과 같다.

$$y = (y_2 - y_1)/(x_2 - x_1) * (x - x_1) + y_1$$

- $P_1$ 과  $P_2$ 의 거리는 다음의 공식을 사용하여 계산한다.

$$d_{P_1, P_2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

### (요구사항)

1. 사용자로부터 두 개의 좌표  $P_1(x_1, y_1)$ 과  $P_2(x_2, y_2)$ 를 입력 받는다.
  - i. 각 좌표는 공백(space)로 구분하여 입력 받으며, 앞에서부터 차례로  $x, y$ 의 값에 해당한다.
  - ii. 모든 숫자는 **실수(float)형태**로 입력 받는다.
  - iii. 숫자가 아닌 문자가 입력되는 경우는 없다고 가정한다. (예외처리 필요 없음)
  - iv. 두 점은 서로 다른 좌표가 입력되며, 같은  $x$ 좌표는 입력되지 않는다고 가정한다.
2. 프로그램의 출력
  - i. 두 점을 지나는 직선의 방정식은 다음의 출력 형태를 만족한다.
$$y = ax + b$$
  - ii. 이 때,  $a$ 와  $b$ 의 값은 계산된 숫자로 출력한다.
  - iii. 두 점 사이의 거리를 계산하여 출력한다.
  - iv. 모든 결과의 출력은 **소수점 이하 둘째 자리까지만 출력한다.**

### (문제힌트)

- 거리 벡터의 distance를 구하는데 필요한 수학 함수
  - $x$ 의 제곱근을 구하는 함수: `sqrt(x);`
- 위 함수를 사용하기 위한 방법은 다음과 같다.
  - 수학 함수가 미리 선언되어 있는 **math.h 파일을 포함(include)** 해야 한다.
  - Linux 환경에서 컴파일을 수행할 때, math 라이브러리를 연결하기 위해 옵션(-lm)을 추가해준다.  
예) `gcc test.c -o test.out -lm`

### (주의사항)

- 파일 이름은 **"assn1\_2.c"**로 저장 할 것
- 보고서는 **"assn1.doc"** or **"assn1.hwp"**로 저장 할 것 (보고서는 통합하여 작성)
- 출력은 아래의 "실행예제"와 동일하도록 작성 할 것 (모든 공백은 'space' 사용)
- 아직 수업시간에 다루지 않은 if, for, while 문 등은 사용하지 않는다.

(실행예제) 빨간 밑줄은 사용자가 입력하는 부분에 해당됨.

```
1 programming.postech.ac.kr:2022 ×  
[wowlhr@programming ~]$ gcc assn1_2.c -o assn1_2.out -lm  
[wowlhr@programming ~]$ ./assn1_2.out  
Enter P1(x1, y1): 3.141 -5.2  
Enter P2(x2, y2): 10.5 6  
The line passing through P1 and P2 is  $y = (1.52)x + (-9.98)$   
The distance between P1 and P2 is 13.40  
[wowlhr@programming ~]$
```