

CSED101. Programming & Problem solving

Spring, 2019

Programming Assignment #1 (30 points)

정예원(ywjeong@postech.ac.kr)

■ **Due:** 2019.03.21 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: 단리, 복리 방식의 원리합계 구하기

(설명)

원금, 연이율, 기간(단위: 년)을 입력 받아 단리와 복리 두 방식의 원리합계를 구하는 프로그램을 작성하시오.

- 단리는 원금에 대해서만 이자를 붙이고 원금에서 생기는 이자에는 다시 이자를 붙이지 않는 방식으로 원리합계 계산은 다음과 같다.

$$\text{원리합계} = \text{원금} \times (1 + \text{연이율} \times \text{기간})$$

- 복리는 원금에 대해서뿐만 아니라 원금에서 생기는 이자에도 원금과 동일한 이율의 이자를 붙이는 방식으로 다음과 같이 계산한다.

$$\text{원리합계} = \text{원금} \times (1 + \text{연이율})^{\text{기간}}$$

(요구사항)

1. 사용자로부터 원금, 연이율, 기간을 입력 받는다.
 - i. 공백(space)로 구분하여 입력 받으며, 앞에서부터 차례로 원금(단위: 원), 연이율(단위: %), 기간(단위: 년)의 값에 해당한다.
 - ii. 입력 값이 저장되는 변수는 원금과 기간은 정수(int) 타입, 연이율은 실수(float) 타입을 이용한다.
 - iii. 잘못된 입력은 없다고 가정한다. 즉, 원금과 기간에 실수 값이 입력되는 경우 등은 없다고 가정한다. (예외처리 필요 없음)
2. 프로그램의 출력
 - i. 출력 값은 실수(float) 타입으로 소수점 이하에서 반올림하여 정수로 출력하며, 방식은 아래와 같다.

```
printf("%.0f", ...);
```

(문제힌트)

- 문제를 해결하는데 필요한 수학 함수
 - 지수 함수: `pow(밑, 지수);`
- 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_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.
- 사용자 정의 함수를 정의하여 사용할 수 있다.

(실행예제 1) 빨간 밑줄은 사용자가 입력하는 부분에 해당됨.

```
[ywjeong@programming2 assn1_yw]$ gcc assn1_1.c -o assn1_1.out -lm
[ywjeong@programming2 assn1_yw]$ ./assn1_1.out
Enter the principal(won), interest(%) and period(year) with space: 450000 8 2
Simple Interest: 522000won
Compound Interest: 524880won
[ywjeong@programming2 assn1_yw]$
```

(실행예제 2)

```
[ywjeong@programming2 assn1_yw]$ ./assn1_1.out
Enter the principal(won), interest(%) and period(year) with space: 150000 1.5 4
Simple Interest: 159000won
Compound Interest: 159205won
[ywjeong@programming2 assn1_yw]$
```

(15 pts) Problem 2: 입력한 소문자 알파벳 대문자 변환, 암호화 및 순번 구하기

(설명)

소문자 알파벳 하나를 입력으로 받아 사용자 정의 함수를 통해 대문자 변환, 암호화 및 해당 알파벳의 순서 번호를 구하고 이를 출력하는 프로그램을 작성하시오.

(요구사항)

1. 사용자로부터 1개의 소문자 알파벳을 입력 받는다.
 - i. 입력 값이 저장되는 변수는 문자(char) 타입이어야 한다.
 - ii. 영문 소문자 이외의 입력은 고려하지 않는다. (예외처리 필요 없음)
2. 사용자로부터 암호화 정수를 입력 받는다.
 - i. 1~9까지의 한 자리 정수만을 암호화 정수로 입력 받는다.
이외의 입력은 고려하지 않는다. (예외처리 필요 없음)
3. 프로그램의 출력
 - i. 입력한 소문자 알파벳을 대문자로 변환하고 출력한다.
 - ii. 대문자로 변환된 알파벳이 몇 번째 알파벳인지 출력한다. (예: **A면 1**)
 - iii. 대문자로 변환된 알파벳 문자에서 입력 정수를 뺀 암호화 문자를 출력한다.
4. 아래의 사용자 정의 함수를 반드시 정의하고 사용한다.
 - i. `lowerToUpper(char c)` : 소문자를 대문자로 변환하여 반환
 - ii. `getSequenceNum(char capital)` : 해당하는 알파벳 대문자의 순서 번호를 반환
 - iii. `encrypt(char c, int num)` : 알파벳 문자에서 한자리 정수를 뺀 문자를 반환

(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.

```
[ywjjeong@programming2 assn1_yw]$ ./assn1_2.out
Enter a character in Lowercase: a
Enter encoding number(1~9): 3
Uppercase of Entered character: A
Sequence Number of entered chracter: 1
Encrypted output: >
[ywjjeong@programming2 assn1_yw]$
```