

CSED101 Programming & Problem Solving

Fall, 2018

Programming Assignment #3

무은재 새내기학부

20180038

박형규

POVIS ID : hyeongkyu

Honor Code : 나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하

였습니다.

1. 프로그램 개요

1.1 목적

이번 프로그램의 목적은 `printf()`, `scanf()`, 그리고 `int`와 `char`등의 프로그래밍 언어를 이용하여 직접 문제를 해결하여 보는 것에 있다. 또한 작업 중 발생하는 에러에 대처할 수 있는 능력을 향상시킴에도 그 목적이 있다.

1.2 전체 구조

Assnignment 1에서는 총 두 가지 문제를 해결해야 한다.

1.2.1 Problem 1. 물 옮기기

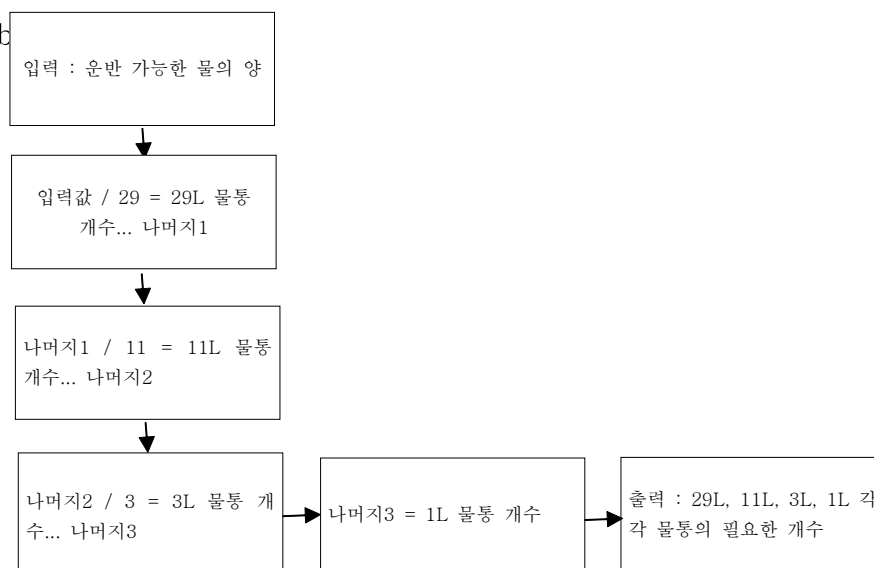
사용자가 특정한 정수값을 입력하면 29L, 11L, 3L, 1L의 물통을 사용하여 가능한 큰 사이즈의 용기를 최대한 많이 사용하여 물을 채울 수 있는 방법이 무엇인지 답을 제시한다. 이는 L수가 큰 물통을 먼저 채우고 남은 물의 양을 나머지 물통에 큰 용량의 물통 순서로 채워 나가는 원리로 나눗셈과 빼기 연산을 적절히 사용하여 계산한다.

1.2.2 Problem 2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기

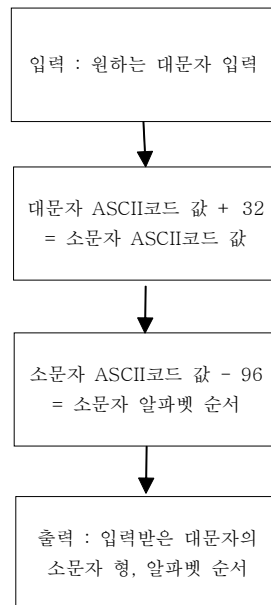
사용자가 알파벳 대문자를 입력하면 이를 `char` 형태로 입력받는다. 그 후 ASCII코드에서 대문자 첫 번째 알파벳(A)와 소문자 첫 번째 알파벳(a)의 간격이 32라는 것을 이용하여 대문자 알파벳을 소문자 알파벳으로 바꾼다. 그 후 A의 ASCII코드 값이 97인 것을 이용하여 적절한 변수 `letter`을 선언하여 알파벳 순서를 구한다.

2. 전체 구조도

2.1 Problem 1



2.2 Problem2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기



3. pseudocode

3.1 pseudocode for problem 1 물 옮기기

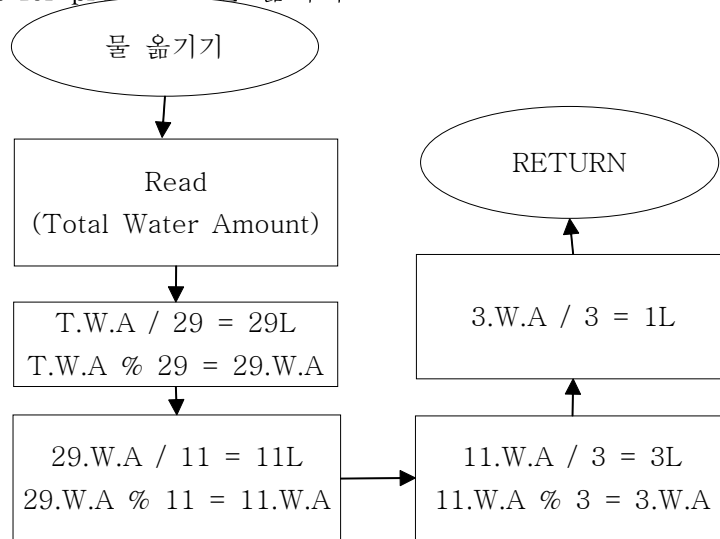
```
Algorithm 물 옮기기
1 set a for input value which is amount of carryable water
2 count of 29L bottle = a / 29
3 set c for amount of water except 29L bottle
4 count of 11L bottle = c / 11
5 set e for amount of water except 29L, 11L bottle
6 count of 3L bottle = e / 3
7 count of 1L bottle = amount of remain water
8 price of all bottle = count of bottle * 1000
9 return price of all bottle
```

3.2 pseudocode for problem 2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기

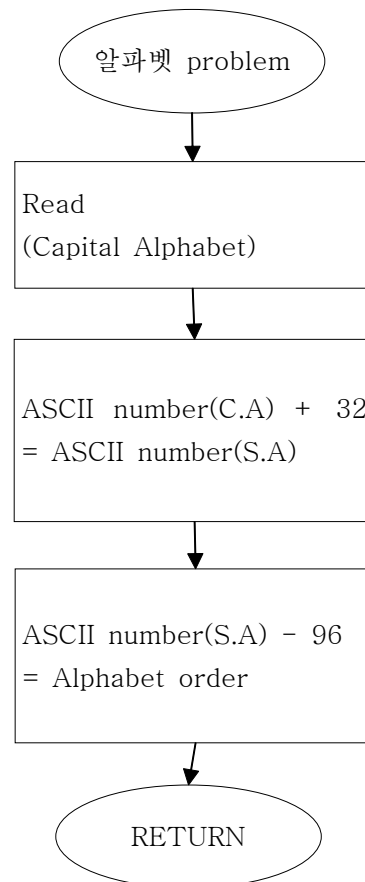
```
Algorithm 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기
1 set variables which are big, sum, letter
2 sum = big + 32
3 letter = sum - 96
4 return letter & sum
end Algorithm 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기
```

3. flowchart

3.1 flowchart for problem1 물 옮기기



3.2 flowchart for problem2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기



4. 프로그램 코드

4.1 problem1. 물 옮기기

Header file 포함

```
#include <stdio.h>
```

Main 함수

```
int main(void)
{
    int a, b, c, d, e, f, g, h;
    printf("운반 가능한 물의 양은? : ");
    scanf("%d", &a);
    b = a / 29;
    c = a - b * 29;
    d = c / 11;
    e = c - d * 11;
    f = e / 3;
    g = e - f * 3;
    h = (b + d + f + g) * 1000;
    printf("생수통 %d개, 약수통 %d개, 큰 물통 %d개, 작은 물통 %d개로 총 운반 비용은 %d원입니다.\n", b, d, f, g, h);
    return 0;
}
```

사용자로부터 물의 양을 입력 받았을 때, 가장 큰 물통의 양인 29L부터 차례대로 나누어 주고, 전체 물의 양에서 이전에 큰 물통이 포함하는 물의 양을 제외하고 이를 반복한다. 그리고 이렇게 하여 나온 생수통, 약수통, 큰 물통, 작은 물통의 총 개수에 1000을 곱하면 운반 비용을 계산 할 수 있다.

4.2 problem 2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기

Header file 포함

```
#include <stdio.h>
```

Main 함수

```

int main(void)
{
    char big;
    printf("대문자 값을 입력해주시오: ");
    scanf("%c", &big);
    char sum = big + 32;
    printf("소문자 값은 다음과 같습니다 : %cWn", sum);
    int letter = sum - 96;
    printf("알파벳 위치 순서는 다음과 같습니다 : %dWn", letter);
    return 0;
}

```

사용자가 대문자 값을 입력하면 각 대문자와 동일한 알파벳의 소문자가 ASCII 코드 상에서 32가 크다는 점을 이용하여 소문자 값을 출력한다. 또한 ASCII코드 상에서 소문자의 첫 알파벳 a가 97부터 시작한다는 점을 이용하여 소문자의 ASCII코드 번호 - 96을 이용하여서 알파벳 순서를 출력한다.

5. 프로그램 실행 방법

5.1 problem1. 물 옮기기

프로그램을 실행시키면 ‘운반 가능한 물의 양은? : ’ 이라는 메시지가 표시 될 것이다. 여기에 자연수 값을 입력하면 입력한 물의 양에 따라 필요한 생수통(29L), 약수통(11L), 큰 물통(3L), 작은 물통(1L)의 필요한 개수가 화면에 표시 될 것이다.

5.2 problem 2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기

프로그램을 실행시키면 ‘대문자 값을 입력해주시오 : ’ 이라는 메시지가 표시 될 것이다. 여기에 적절한 대문자 값을 입력하면 아래에 입력한 대문자 값에 따른 소문자 값과 입력한 알파벳의 순서를 알려주는 메시지가 표시된다.

6. 프로그램 실행 예제

6.1 problem1. 물 옮기기

```
Microsoft Visual Studio 디버그 콘솔
운반 가능한 물의 양은? : 300
생수통 10개, 약수통 0개, 큰 물통 3개, 작은 물통 1개로 총 운반 비용은 14000원입니
다.
C:\Users\박형규\source\repos\assn1_1\Debug\assn1_1.exe(5248 프로세스)이(가) 0 코
드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

물의 양을 300L로 입력하였더니 생수통 10개, 약수통 0개, 큰 물통 3개, 작은 물통 1개가 필요하며 총 비용이 14000원이 필요하다는 결과가 표시되었다.

6.2 problem2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기

```
Microsoft Visual Studio 디버그 콘솔
대문자 값을 입력해 주십시오 : G
소문자 값은 다음과 같습니다 : g
알파벳 위치 순서는 다음과 같습니다 : 7
C:\Users\박형규\source\repos\assn1_2\Debug\Project2.exe(12664 프로세스)이(가) 0
코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

알파벳 대문자 G를 입력하였더니 G의 소문자는 g이고 G는 알파벳 순서 7번째라는 결과 값이 화면에 표시되었다.

7. 토론 및 결론

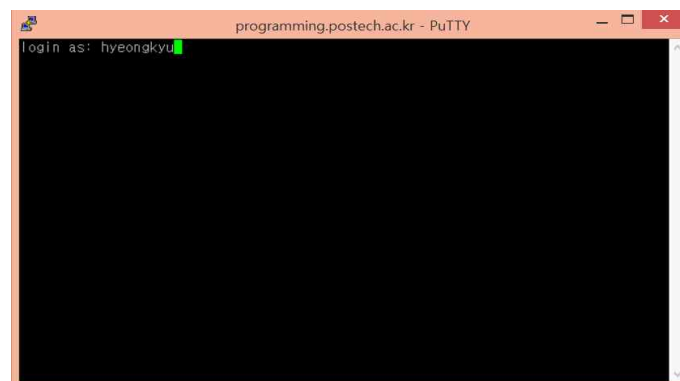
7.1 problem1. 물 옮기기

물 옮기기 problem을 수행하기 위해서는 먼저 특정한 물의 값이 제시되었을 때 29L, 11L, 3L, 1L짜리의 물통을 가지고 큰 물통부터 채워나가는 방법을 먼저 생각해 봐야 했다. 그에 따른 방법으로 큰 물통부터 채워나가면 간편하다는 결론에 이르렀고 이는 problem수행 중 나눗셈을 이용하여 큰 물통의 개수부터 구하는 방법으로 귀결되었다. problem1을 수행하면서 ‘나눗셈’이 이뤄질 때 정확히 나누어떨어지지 않을 때에도 결과값은 나머지를 제외한 몫의 값만이 나타난다는 것을 알게 되었다. 또한 처음에 이 problem을 해결하기 위한 알고리즘을 짤 때 입력값을 29로 나눌 때 나머지의 범위에 따라 분류를 하려고 하였는데, 이렇게 problem을 해결하려고 하면 아직 배우지 않은 함수를 사용해야 한다는 것을 알고 나눗셈 연산자를 이용하는 새로운 방법을 도출하여 문제를 해결한 점도 흥미로웠다.

7.2 problem2. 입력 받은 알파벳 대문자를 소문자로 변환하기 및 위치 구하기

problem2를 수행하기 위해서는 평소 사용하던 int형이 아닌 char형을 사용해야 했다. char형을 사용하면서 ASCII코드를 이용하여 char형과 int형을 연동하여 문제를 해결할 수 있었던 점이 새로웠다. 우리가 평소 문자로만 인식하는 알파벳을 ASCII코드에서는 특정한 정수 값을 부여하여 정수 값과 문자를 연동하여 문제 해결을 하는 데 사용하는 점이 가장 인상 깊었던 것 같다.

8. 리눅스 서버 접속 & 프로그램 컴파일 및 실행 과정



리눅스 서버에 접속하여 로그인 하였다.


```
hyeongkyu@programming:~  
login as: hyeongkyu  
hyeongkyu@programming.postech.ac.kr's password:  
Last login: Fri Sep 21 14:05:39 2018 from m1-pc13.postech.ac.kr  
===== Welcome to programming server =====  
Inquiry: 279-2520, hmos@postech.ac.kr  
===== [hyeongkyu@programming ~]$
```

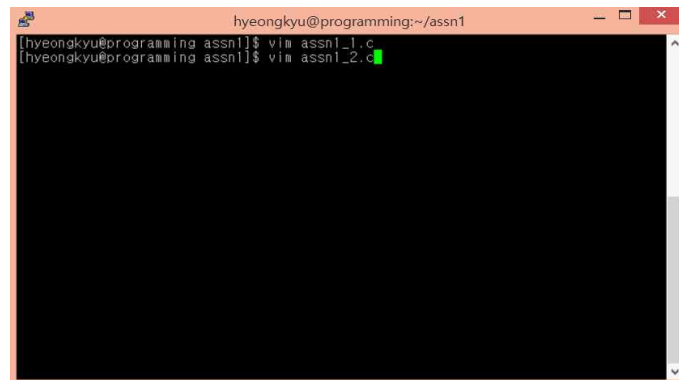
로그인에 성공하면 다음과 같은 창이 뜬다.

```
hyeongkyu@programming:~/asn1  
[hyeongkyu@programming asn1]$ vim asn1_1.c
```

asn1_1.c 파일을 생성/ 편집한다.

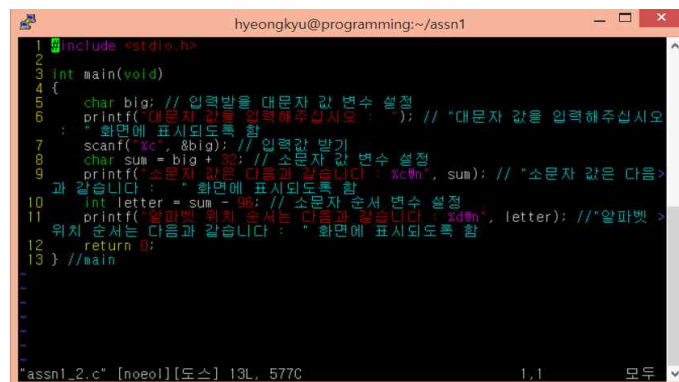
```
hyeongkyu@programming:~/asn1  
1 #include <stdio.h>  
2  
3 int main(void)  
4 {  
5     int a, b, c, d, e, f, g, h;  
6     printf("문만 가능한 물의 양은? : "); //문만 가능한 물의 양은? : 물 출력  
7     scanf("%d", &a); // 사용자가 원하는 물의 양을 입력받음  
8     b = a / 29; // b는 29L의 물통 개수  
9     c = a - b * 29; // c는 29L의 물통을 채우고 남은 물의 양  
10    d = c / 11; // d는 11L의 물통 개수  
11    e = c - d * 11; // e는 11L의 물통을 채우고 남은 물의 양  
12    f = e / 3; // f는 3L의 물통 개수  
13    g = e - f * 3; // g는 1L의 물통 개수  
14    h = (b + d + f + g) * 1000; // 물통 개수에 따른 가격값  
15    printf("물통을 %d개, 약수를 %d개, 큰 물통 %d개, 작은 물통 %d개를 총 %d번  
16    배를 들 %d원입니다.\n", b, d, f, g, h); // 마지막 결과값을 출력해주도록 함  
17    return 0;  
18 }  
19  
20  
asn1_1.c [noel] [도스] 17L, 704C 17,1 모두
```

asn1_1.c 파일을 편집하는 모습이다.



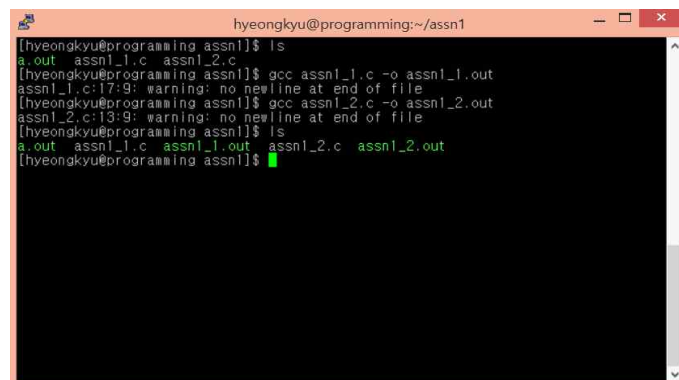
```
hyeongkyu@programming:~/assn1
[hyeongkyu@programming assn1]$ vim assn1_1.c
[hyeongkyu@programming assn1]$ vim assn1_2.c
```

assn1_2.c 파일을 생성/ 편집한다.



```
hyeongkyu@programming:~/assn1
1 #include <stdio.h>
2
3 int main(void)
4 {
5     char big; // 입력받을 대문자 값 변수 설정
6     printf("대문자 값을 입력해주세요 : "); // "대문자 값을 입력해주세요"
7     scanf("%c", &big); // 입력값 받기
8     char sum = big + 32; // 소문자 값 변수 설정
9     printf("소문자 값은 다음과 같습니다 : %c\n", sum); // "소문자 값은 다음>
10    과 같습니다 : " 화면에 표시되도록 함
11    int letter = sum - 96; // 소문자 순서 변수 설정
12    printf("알파벳 위치 순서는 다음과 같습니다 : %d\n", letter); // "알파벳 >
13    위치 순서는 다음과 같습니다 : " 화면에 표시되도록 함
14    return 0;
15 }
```

assn1_2.c 파일을 편집하는 모습이다.



```
hyeongkyu@programming:~/assn1
[hyeongkyu@programming assn1]$ ls
a.out assn1_1.c assn1_2.c
[hyeongkyu@programming assn1]$ gcc assn1_1.c -o assn1_1.out
assn1_1.c:17:9: warning: no newline at end of file
[hyeongkyu@programming assn1]$ gcc assn1_2.c -o assn1_2.out
assn1_2.c:13:9: warning: no newline at end of file
[hyeongkyu@programming assn1]$ ls
a.out assn1_1.c assn1_1.out assn1_2.c assn1_2.out
[hyeongkyu@programming assn1]$
```

assn1 폴더로 이동하여서 assn1_1.c와 assn1_2.c를 gcc를 통해서 컴파일 하는 모습이다.

```
hyeongkyu@programming:~/assn1
[hyeongkyu@programming assn1]$ ls
a.out  assn1_1.c  assn1_2.c
[hyeongkyu@programming assn1]$ gcc assn1_1.c -o assn1_1.out
assn1_1.c:17:9: warning: no newline at end of file
[hyeongkyu@programming assn1]$ gcc assn1_2.c -o assn1_2.out
assn1_2.c:13:9: warning: no newline at end of file
[hyeongkyu@programming assn1]$ ls
a.out  assn1_1.c  assn1_1.out  assn1_2.c  assn1_2.out
[hyeongkyu@programming assn1]$ ./assn1_1.out
문반 가능한 물의 양은? : 300
생수통 10개, 약수통 0개, 큰 물통 3개, 작은 물통 1개로 총 문반 비용은 14000원입니다.
[hyeongkyu@programming assn1]$
```

컴파일 한 assn1_1.c파일을 실행한 모습이다.

```
hyeongkyu@programming:~/assn1
[hyeongkyu@programming assn1]$ ls
a.out  assn1_1.c  assn1_2.c
[hyeongkyu@programming assn1]$ gcc assn1_1.c -o assn1_1.out
assn1_1.c:17:9: warning: no newline at end of file
[hyeongkyu@programming assn1]$ gcc assn1_2.c -o assn1_2.out
assn1_2.c:13:9: warning: no newline at end of file
[hyeongkyu@programming assn1]$ ls
a.out  assn1_1.c  assn1_1.out  assn1_2.c  assn1_2.out
[hyeongkyu@programming assn1]$ ./assn1_1.out
문반 가능한 물의 양은? : 300
생수통 10개, 약수통 0개, 큰 물통 3개, 작은 물통 1개로 총 문반 비용은 14000원입니다.
[hyeongkyu@programming assn1]$ ./assn1_2.out
대문자 값을 입력해 주십시오 : g
소문자 값을 다음과 같읍니다 : g
알파벳 위치 순서는 다음과 같읍니다 : 7
[hyeongkyu@programming assn1]$
```

컴파일 한 assn1_2.c 파일을 실행한 모습이다.