

2019년 2학기

프로그래밍과 문제해결

Assignment #1

담당교수 **윤은영**
학번 **20190084**
학과 **무은재학부**
이름 **권민재**
POVIS ID **mzg00**

명예서약(Honor code)

"나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다."

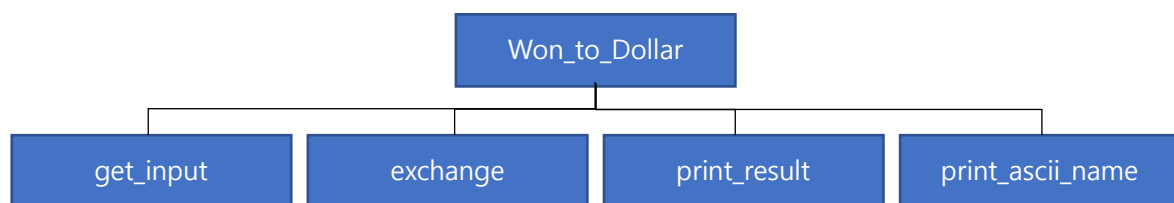
Problem 1: 환화를 달러로 환전하기

1. 문제의 개요

본 프로그램의 역할은 아래와 같다.

- 사용자로부터 환전할 원화의 양과, 환율을 입력받는다.
- 입력 받은 수에 따라 달러로 전환한다.
- 전환한 달러를 지정된 단위에 맞게 출력한다.

이 때 사용되는 구조 차트는 아래와 같이 표현될 수 있다.



- **입력부:** 사용자가 입력한 원화와 환율을 변수에 저장한다.
- **처리부:** 원화를 달러로 전환하고, 각각의 단위에 맞게 분배한다.
- **출력부:** 전환이 완료된 결과를 단위에 맞춰 출력한다.
- **기타:** 자신의 이름을 ASCII 코드로 출력한다.

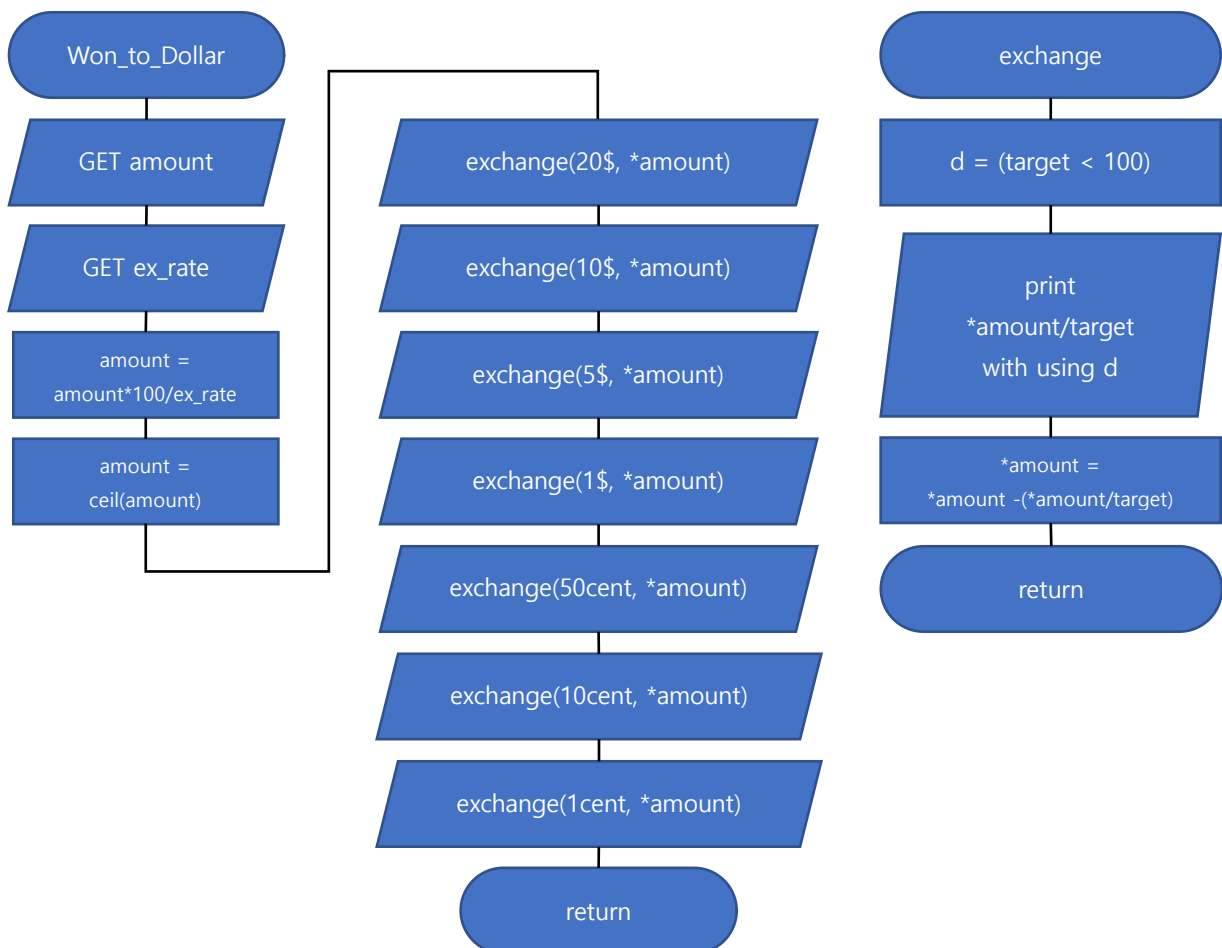
2. 알고리즘

Pseudo-code for Won_to_Dollar

1	include essential headers <stdio.h>, <math.h>
2	function exchange – use call by reference to modify amount
3	set d = (target < 100) – It's used to determine print 'cent' or '\$'
4	print number of target unit
5	subtract converted dollar from amount
6	
7	get amount from user
8	get ex_rate from user
9	convert won to dollar by amount*100/ex_rate – multiply 100 is for ceil
10	ceil dollar to the second decimal place
11	
12	call exchange with target unit 20 \$ and amount's address

13	call exchange with target unit 10 \$ and amount's address
14	call exchange with target unit 5 \$ and amount's address
15	call exchange with target unit 50 cent and amount's address
16	call exchange with target unit 10 cent and amount's address
17	call exchange with target unit 1 cent and amount's address
18	
19	print my name with ASCII code

위의 의사 코드를 Flowchart를 이용해서 나타내면 아래와 같다. ASCII 코드를 이용하여 이름을 출력하는 부분은 생략되었다.



3. 프로그램 구조 및 설명

a) 환전할 원화의 양과 환율 입력

- 프로그램을 실행하면 환전할 원화의 양과 환율을 입력받는다. 이들은 각각 변수 amount와 ex_rate에 저장된다.

b) 원화를 달러로 소수 둘째 자리까지 올림하여 저장

- 원화를 달러로 변환한 후 소수 둘째 자리까지 ceil 함수를 이용하여 올림한다. 소수 둘째 자리까지 올림하기 위해, `amount/ex_rate`에 100을 곱한 값을 ceil 함수의 인자로 실행시키고, 그의 return 값을 `amount`에 int 형식으로 변환하여 저장한다. 이때, `amount`에는 환전해야 하는 양의 100배의 달러가 저장될 것이다. 하지만, 컴퓨터의 실수 처리 과정에서 일어나는 오차를 최대한 줄이기 위해, 이 값을 100으로 나누지 않고 그대로 int 형식으로 사용할 것이다.

c) 올림하여 저장한 달러를 지정된 단위 별로 출력

- 반복적인 출력 작업을 줄이기 위해, `exch`라는 함수를 별도로 작성하여 구현한다. `exch`는 변환을 원하는 단위의 100배의 값을 입력 받는 int형의 변수 `target`과, `main` 함수의 변수 `amount`의 주소를 입력하는 포인터 변수 `*amount`를 매개변수로 가진다. 만약, `amount`를 20달러 단위로 변환 할 때 몇 개가 필요한지 출력하기 위해서는 `amount(2000, &amount);`와 같이 호출하면 된다. 이때, 단위에 100을 곱한 값이 들어가야 하는 이유는 `amount`가 이미 100배 되어 있는 값이기 때문이다. `exch`가 호출되면, 우선 비교 연산자를 이용해 `target`이 100보다 작으면 지역 변수 `d`에 1을, 아니면 `d`에 0을 저장한다. 이때, ASCII 코드를 `36 + d*63, 1 + d*100, 1 + d*109, 1 + d*115`와 같이 잘 이용하면, `d`에 따라 'cent'나 '\$'를 출력할 수 있다. 그 후, `*amount/target`을 통해 해당 `target` 단위에 필요한 개수를 구하여 출력하고, `*amount`에서 `target*(amount/target)`을 빼서 변환한 만큼 `amount`를 차감한다. 이때, `exch` 함수의 `target`을 큰 단위부터 순서대로 모두 호출하여 실행하면 문제에서 요구하는 출력을 얻을 수 있다.

d) ASCII 코드를 이용하여 이름 출력

- ASCII 코드 표를 참고하면, 'Gwon Min Jae'의 각 글자의 ASCII 코드는 71, 119, 111, 110, 32, 77, 105, 110, 32, 74, 97, 101임을 알 수 있다. 이를 `%c`를 이용하여 `printf("Written by %c%c%c%c%c%c%c%c%c%c%c%c%c\n", 71, 119, 111, 110, 32, 77, 105, 110, 32, 74, 97, 101);`와 같이 `printf`를 호출하여 ASCII 코드로 이름을 출력할 수 있다.

4. 프로그램 실행 방법 및 예제

```
beta@beta-MacbookPro ~ ssh mzg00@programming2.postech.ac.kr -p 2022
mzg00@programming2.postech.ac.kr's password:
Last login: Thu Oct 3 19:18:38 2019 from mdom18313-1.postech.ac.kr
mzg00 ~
```

위 명령어를 실행하고 비밀번호를 입력하여 실습 서버에 접속하여 본인의 리눅스 계정에 로그인한다.

```

mzg00 ~ cd ~/CSED101/ASSN_1
mzg00 master ~/CSED101/ASSN_1 ls
assn1.pdf assn1_1.c assn1_1_test.c assn1_2.c chck1_1.py
mzg00 master ~/CSED101/ASSN_1 gcc assn1_1.c -o assn1_1.out -lm; ./assn1_1.out

```

이동한 폴더에서 작성한 프로그램의 소스코드를 컴파일 하고 실행한다. 리눅스에서 ';'를 이용하면 명령어를 연속적으로 수행할 수 있다. 소스 코드에 문제가 없으므로, 정상적으로 컴파일되어 실행 되는 모습을 볼 수 있다.

```

mzg00 master ~/CSED101/ASSN_1 gcc assn1_1.c -o assn1_1.out -lm; ./assn1_1.out
Enter the amount of exchange: 123456
Enter today's exchange rate: 1190

```

위와 같이 제시된 실행 예제와 같은 원화의 양과 환율을 입력한다.

```

mzg00 master ~/CSED101/ASSN_1 gcc assn1_1.c -o assn1_1.out -lm; ./assn1_1.out
Enter the amount of exchange: 123456
Enter today's exchange rate: 1190
20 $: 5
10 $: 0
5 $: 0
1 $: 3
50 cent: 1
10 cent: 2
1 cent: 5
Written by Gwon Min Jae

```

제시된 실행 예제와 같은 결과가 출력됨을 볼 수 있다. 또한, ASCII 코드를 통해 정상적으로 이름 이 출력됨을 볼 수 있다.

5. 토론

- 주어진 문제는 소수점 둘째 자리까지 올림하는 것인데, 소수점 셋째 자리에서 올림하기 위해서는 $\text{amount} = (\text{int})\text{ceil}(((\text{double})((\text{int})((\text{double})\text{amount}/\text{ex_rate}*\text{pow}(10,3)))/10);$ 와 같이 하면 된다.
- \$와 cent를 다른 형식으로 출력하기 위해 함수를 두 개 만들 수도 있지만, 비교연산자와 ASCII 코드를 잘 이용하면 if와 같은 조건문을 사용하지 않고도 하나의 함수 내에서 구분하여 다른 형식으로 출력할 수 있다.
- 이 프로그램에서 실수 자료형을 이용해 잘못 처리하면 부동 소수점 오차로 인해 의도되지 않은 결과가 나올 수 있다. 그렇기 때문에, 실수를 이용할 때에는 최대한 오차가 적도록 유의해야 한다.

6. 결론

- 본 과제를 통해 미리 정의된 printf, scanf, ceil 등의 함수를 사용하는 방법을 익힐 수 있었고, ASCII 코드를 통해 \$, cent, 그리고 이름을 출력하는 과정을 통해 아스키 코드에 대한 이해도 높일 수 있었다.

7. 개선방향

- 해당 프로그램에서는 `exch`를 반복적으로 호출하여야만 했지만, 앞으로 배열 반복문을 이용하면 더 간결하게 코드를 작성할 수 있을 것이다.

Problem 2: 초음파 센서로 거리 측정하기

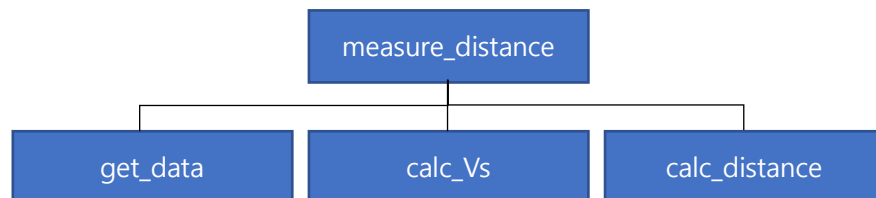
1. 문제의 개요

본 프로그램의 역할은 아래와 같다.

- 사용자로부터 섭씨 온도를 입력 받고, 그에 따른 음속을 계산해서 출력한다.
- 사용자로부터 측정 시간을 입력 받아 평균값을 구한다.
- 구한 음속과 시간 평균값을 이용하여 계산한 거리를 출력한다.
- 이 때 사용되는 주어진 공식들은 아래와 같다.

$$L = \frac{t \times V_s}{2}, \quad V_s = 331.3 + 0.606 \times T$$

이 때 사용되는 구조 차트는 아래와 같이 표현될 수 있다.



- **입력부:** 사용자로부터 섭씨 온도와 시간들을 입력받는다.
- **처리부:** 섭씨 온도로 음속을 구하고, 시간의 평균값을 이용하여 거리를 구한다.
- **출력부:** 구한 음속과 거리를 출력한다.

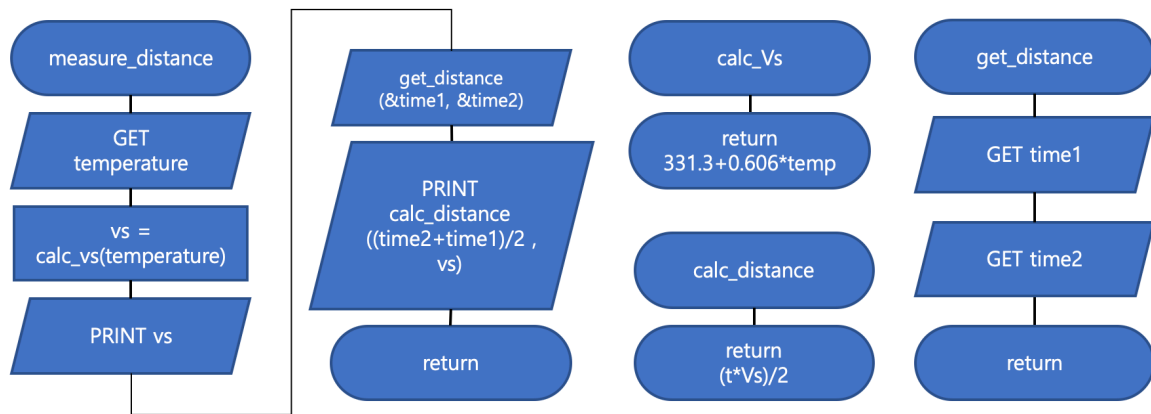
2. 알고리즘

Pseudo-code for measure_distance

1	include essential headers <stdio.h>, <math.h>
2	function calc_Vs
3	compute and return 331.3 + 0.606 * temp
4	
5	function calc_distance
6	compute and return (t * Vs) / 2
7	
8	function get_data – use
9	get first measured time from user and save to main function's variable time1
10	get second measured time from user and save to main function's variable time2

11	
12	get temperature from user
13	call calc_Vs and print sound speed
14	
15	call get_data to get times from user
16	compute distance and print it

위의 의사 코드를 Flowchart를 이용해서 나타내면 아래와 같다.



3. 프로그램 구조 및 설명

a) 온도 입력 및 음속 계산

- 프로그램을 실행하면 사용자로부터 scanf로 섭씨 온도를 입력 받아 함수 calc_vs를 통해 음속을 계산하여 변수 vs에 저장한다. vs를 printf로 출력한다.

b) 시간 입력 및 거리 계산

- 함수 get_data를 통해 main 함수의 변수 time1과 time2에 시간 2개를 입력받는다. 입력한 시간들의 평균값과 vs를 이용하여 calc_distance를 호출하여 거리를 계산하고 출력한다.

4. 프로그램 실행 방법 및 예제

```

beta@beta-MacbookPro ~$ ssh mzg00@programming2.postech.ac.kr -p 2022
mzg00@programming2.postech.ac.kr's password:
Last login: Thu Oct  3 23:15:32 2019 from pbs3.postech.ac.kr
mzg00 ~

```

위 명령어를 실행하고 비밀번호를 입력하여 실습 서버에 접속하여 본인의 리눅스 계정에 로그인한다.


```
mzg00 ~ cd ~/CSED101/ASSN_1
mzg00 master ~/CSED101/ASSN_1 ls
assn1.pdf assn1_1.c assn1_1_test.c assn1_2.c chck1_1.py
mzg00 master ~/CSED101/ASSN_1 gcc assn1_2.c -o assn1_2.out -lm; ./assn1_2.out
```

이동한 폴더에서 작성한 프로그램의 소스코드를 컴파일 하고 실행한다. 리눅스에서 ';'를 이용하면 명령어를 연속적으로 수행할 수 있다. 소스 코드에 문제가 없으므로, 정상적으로 컴파일되어 실행되는 모습을 볼 수 있다.

```
mzg00 master ~/CSED101/ASSN_1 gcc assn1_2.c -o assn1_2.out -lm; ./assn1_2.out
Input the temperature in Celsius: 30.5
Speed of sound is 349.78 m/s
```

실행 예제와 같은 섭씨 온도를 입력하였을 때 예제와 같은 음속이 나오는 것을 볼 수 있다.

```
mzg00 master ~/CSED101/ASSN_1 gcc assn1_2.c -o assn1_2.out -lm; ./assn1_2.out
Input the temperature in Celsius: 30.5
Speed of sound is 349.78 m/s
Input the first measured time (seconds): 0.019
Input the second measured time (seconds): 0.021
The distance is 3.50 m
```

이후, 예제와 같이 시간들을 입력하였을 때, 예제와 같은 거리가 나오는 것을 볼 수 있다.

5. 토론

- 사용자로부터 온도, 시간 등을 입력 받을 때 숫자 외의 문자들도 입력 받을 수 있어서 의도하지 않은 동작이 발생할 수 있다.
- 이를 해결하기 위해서는 사용자로부터 숫자가 아닌 문자열로 입력 받고, 문자 하나하나가 ASCII 코드의 숫자 범위에 있거나 '.'의 아스키 코드에 속하는지 검사할 방법이 있다.
- 이는 앞으로 배울 반복문과 조건문을 이용하면 구현할 수 있을 것이다.

6. 결론

- 본 과제를 통해 Call-by-address를 통한 함수 호출을 이해할 수 있었고, 수학 문제를 프로그래밍을 통해 풀 수 있다는 사실을 알 수 있었다.

7. 개선방향

- 해당 프로그램은 float 자료형의 변수를 사용하는데, float는 double보다 실수를 표현하는 정밀도가 낮기 때문에 계산에 오차가 발생할 수 있다. 그렇기 때문에 더 정확하게 계산할 수 있는 방법이 필요하다.