LAPORAN PRAKTIKUM
PEMROGRAMAN TERSTRUKTUR
PRAKTIKUM I – PENDAHULUAN
KELAS B

Disusun oleh :
Nama : Monica Sari
NIM : 175090800111002
Hari/tgl Praktikum : Selasa, 9 April 2019

LABORATORIUM KOMPUTASI
JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
2019

A. Percobaan 1

Problem

Objective

In this challenge, we will learn some basic concepts of C that will get you started with the language. You will need to use the same syntax to read input and write output in many C challenges.

Task

This challenge requires you to print $Hello, World!$ on a single line, and then print the already provided input string to stdout.

Note: You do not need to read any input in this challenge.

Input Format

You do not need to read any input in this challenge.

Output Format

Print $Hello, World!$ on the first line, and the string from the given input on the second line.

Sample Input 0

```
Welcome to C programming.
```

Sample Output 0

```
Hello, World!
Welcome to C programming.
```

Penyelesaian

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


int main()
{

    char r[100];
    fgets(r,sizeof(r),stdin);
    printf("Hello, World!\n%s",r);

    return 0;
}
```

Penjelasan

Program tersebut merupakan program dasar untuk menampilkan kalimat “Hello, World!” dengan menggunakan perintah *char* yang berfungsi untuk mendeklarasikan karakter *r* dengan panjang karakter 100 huruf, fungsi *fgets* untuk mengambil input dari *stdin* dengan *size of* yang berfungsi untuk mencetak hasil program.

B. Percobaan 2

Problem

```
char ch;
scanf("%c", &ch);
printf("%c", ch);
```

This piece of code prints the character ch.

You can take a string as input in C using `scanf("%s", s)`. But, it accepts string only until it finds the first space.

In order to take a line as input, you can use `scanf("%[^\n]%*c", s);` where s is defined as `char s[MAX_LEN]` where MAX_LEN is the maximum size of s. Here, `[]` is the scanset character. `^\n` stands for taking input until a newline isn't encountered. Then, with this `%*c`, it reads the newline character and here, the used $*$ indicates that this newline character is discarded.

Note: After inputting the character and the string, inputting the sentence by the above mentioned statement won't work. This is because, at the end of each line, a new line character (`\n`) is present. So, the statement: `scanf("%[^\n]%*c", s);` will not work because the last statement will read a newline character from the previous line. This can be handled in a variety of ways and one of them being: `scanf("\n");` before the last statement.

Task

You have to print the character, ch, in the first line. Then print s in next line. In the last line print the sentence, sen.

Input Format

First, take a character, ch as input.
Then take the string, s as input.
Lastly, take the sentence sen as input.

Penyelesaian

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{

    char ch;
    scanf("%c",&ch);
    char s[20];
    scanf("%s",s);
    char t;
    scanf("%c",&t);
    char sen[100];
    scanf("%[^\n]",sen);
    printf("%c\n",ch);
    printf("%s\n",s);
    printf("%s\n",sen);
    return 0;

}
```

Penjelasan

Dalam kode ini dapat dilihat bahwa deklarasi ch, lalu deklarasi a dengan b diberikan perbedaan sehingga akan memberi variasi perintah dengan hasil yang sama. Baris ke 11 dapat dilihat scanf yang berarti menginputkan nilai ch dengan "%c" dan diikuti oleh & didepan ch. Sedangkan pada baris ke 12 menggunakan "%s" tanpa diikuti oleh & didepan a dapat dilihat perbedaannya. \n digunakan untuk menuliskan kode pada baris baru sehingga tidak berjejer. Pada baris ke 13 digunakan perintah yang berbeda dari sebelumnya tapi hasilnya pun akan tetap sama. Lalu perintah akan di print pada baris 15 untuk memunculkan nilai ch digunakan perintah %c. Untuk memunculkan nilai a dan b digunakan perintah %s.

C. Percobaan 3

Problem

Input Format

The first line contains two integers.
The second line contains two floating point numbers.

Constraints

- $1 \leq$ integer variables $\leq 10^4$
- $1 \leq$ float variables $\leq 10^4$

Output Format

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to 1 decimal place) separated by a space on the second line.

Sample Input

```
10 4
4.0 2.0
```

Sample Output

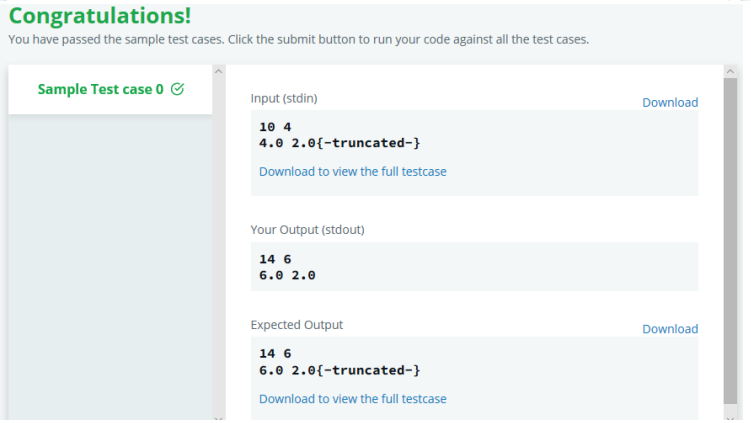
```
14 6
6.0 2.0
```

Penyelesaian

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
    int i,j;
    float f,g;

    scanf("%d %d %f %f",&i,&j,&f,&g);
    printf("%d %d\n%.1f %.1f",i+j,i-j,f+g,f-g);
    return 0;
}
```



Penjelasan

Program tersebut adalah program penjumlahan dan pengurangan bilangan sederhana. Variabel x dan y dideklarasikan dengan perintah *in*, sedangkan variabel z dan o dideklarasikan menggunakan *float*. Perbedaan antara *int* dan *float* adalah pada tampilan hasil, *float* akan menampilkan hasil yang memiliki angka dibelakang koma sedangkan *int* tidak menampilkan bilangan dibelakang koma. Pada program diberikan input x, y, z dan o. kemudian diberi perintah untuk menampilkan hasil operasi bilangan yaitu x+y, x-y, z+o, dan z-o.