

LAPORAN PRAKTIKUM
PEMROGRAMAN TERSTRUKTUR
PRAKTIKUM I – PENDAHULUAN
KELAS B



Disusun Oleh:
Nama : Rona Wahyu B
NIM : 175090801111003
Hari/tgl Praktikum : Selasa, 9 April 2019

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

A. 1. Masalah

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.
```

Gambar 1.1(Masalah)

2. Penyelesaian

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main ()
5  {
6      char r[100];
7      fgets(r, sizeof(r), stdin);
8      printf("Hello, World!\n%s", r);
9
10     return 0;
11 }
12
```

Gambar 1.2 (Kode Program Penyelesaian)



Gambar 1.3 (Output)

3. Penjelasan

Program di atas adalah program dasar untuk menampilkan kata “Hello World” dengan perintah yaitu *Char* yang berfungsi untuk mendeklarasikan character “*r*” dengan panjang 100 huruf, *fgets* berfungsi untuk mengambil input dari *STDIN* dengan *size of* yang berfungsi sebagai seperti return 0, sedangkan *printf* berfungsi untuk mencetak.

B. 1. Masalah

Objective

This challenge will help you to learn how to take a character, a string and a sentence as input in C.

To take a single character *ch* as input, you can use `scanf("%c", &ch);` and `printf("%c", ch)` writes a character specified by the argument `ch` to `stdout`

```
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.

Output Format

Print three lines of output. The first line prints the character, *ch*.
The second line prints the string, *s*.
The third line prints the sentence, *sen*.

Sample Input 0

```
C
Language
WelCome To C!!
```

Sample Output 0

```
C
Language
WelCome To C!!
```

Gambar 2.1(Masalah)

2. Penyelesaian

```
1 |include <stdio.h>
2 |include <string.h>
3 |include <math.h>
4 |include <stdlib.h>
5
6 int main()
7 {
8     char ch;
9     scanf("%c",&ch);
10    char s[20];
11    scanf("%s",s);
12    char t;
13    scanf("%c",&t);
14    char sen[100];
15    scanf("%s",sen);
16    printf("%c\n",ch);
17    printf("%s\n",s);
18    printf("%s\n",sen);
19    return 0;
20 }
21
22
```

Gambar 2.2 (Kode Program Penyelesaian)

Input (stdin)

[Download](#)

C
Language
Welcome To C!!

Your Output (stdout)

C
Language
Welcome To C!!

Expected Output

[Download](#)

C
Language
Welcome To C!!

Gambar 2.3 (Output)

3. Penjelasan

Ada empat input berupa character yang diberikan kepada kita.. Yaitu ch, s, t, sen. Pertama kita harus declare variable dari keempat input tersebut, lalu setelah declare variable kita menggunakan scanf untuk mengambil data input.. Lalu setelah pengambilan data input.. Kita harus mencetak dengan menuliskan printf("%c\n", c) \n disini digunakan untuk melakukan pergantian baris. Dan %c digunakan untuk pembacaan suatu variable yang berbentuk character.

C. 1. Masalah

Objective

The fundamental data types in c are int, float and char. Today, we're discussing int and float data types.

The `printf()` function prints the given statement to the console. The syntax is `printf("format string", argument_list);`. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write `%d` (integer), `%c` (character), `%s` (string), `%f` (float) respectively.

The `scanf()` function reads the input data from the console. The syntax is `scanf("format string", argument_list)`; For ex: The `scanf("%d", &number)` statement reads integer number from the console and stores the given value in variable *number*.

To input two integers separated by a space on a single line, the command is `scanf("%d %d", &n, &m)`, where *n* and *m* are the two integers.

Task

Your task is to take two numbers of int data type, two numbers of float data type as input and output their sum:

1. Declare **4** variables: two of type `int` and two of type `float`.
2. Read **2** lines of input from `stdin` (according to the sequence given in the 'Input Format' section below) and initialize your **4** variables.
3. Use the `+` and `-` operator to perform the following operations:
 - o Print the sum and difference of two `int` variable on a new line.
 - o Print the sum and difference of two `float` variable rounded to one decimal place on a new line.

Input Format

The first line contains two integers.

The second line contains two floating point numbers.

Constraints

- $1 \leq \text{integer variables} \leq 10^4$
- $1 \leq \text{float variables} \leq 10^4$

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

Explanation

When we sum the integers **10** and **4**, we get the integer **14**. When we subtract the second number **4** from the first number **10**, we get **6** as their difference.

When we sum the floating-point numbers `4.0` and `2.0`, we get `6.0`. When we subtract the second number `2.0` from the first number `4.0`, we get `2.0` as their difference.

Gambar 3.1(Masalah)

2. Penyelesaian

```
1 |include <stdio.h>
2 |include <string.h>
3 |include <math.h>
4 |include <stdlib.h>
5
6 int main()
7 {
8     int i,j;
9     float f,g;
10
11     scanf("%d %d %f %f", &i, &j, &f, &g);
12     printf("%d %d\n%.1f %.1f", i+j, i-j, f+g, f-g);
13     return 0;
14 }
15
16
```

Gambar 3.2 (Kode Program Penyelesaian)

Input (stdin)

[Download](#)

10 4
4.0 2.0{-truncated-}

[Download to view the full testcase](#)

Your Output (stdout)

14 6
6.0 2.0

Expected Output

[Download](#)

14 6
6.0 2.0{-truncated-}

[Download to view the full testcase](#)

Gambar 3.3 (Output)

3. Penjelasan

Pada program diatas, integer dan float dengan input x1, x2, y1 dan y2. Kemudian diberi perintah scanf untuk yang integer menggunakan persen d. Akan tetapi, scanf yang float menggunakan persen f. Pada keduanya ditambah &x1 dan &x2 untuk mengambil data dari input. Setelah itu, menggunakan perintah printf berfungsi untuk menampilkan di layar hasil dari sum and difference dengan ditambahkan perintah sum and difference agar output yang diminta dapat dipenuhi.