

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", &a, &b)`, where ***a*** and ***b*** 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.

Declare **4** variables: two of type int and two of type float.

Read **2** lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your **4** variables.

Use the **+** and **-** operator to perform the following operations:

Print the sum and difference of two int variable on a new line.

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 ≤ integer variables ≤ 10^4

1 ≤ float variables ≤ 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

Then 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.

Then 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.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c,d;
5     float e,f,g,h;
6     scanf("%d%d",&a,&b);
7     scanf("%f%f",&c,&d);
8     g=a-b;
9     f=c-d;
10    h=a-b;
11    i=c-d;
12    printf("%d %d\n",g,h);
13    printf("%f %f\n",f,i);
14 }
```

	Input	Expected	Got	
✓	10 4 4.0 2.0	14 6 6.0 2.0	14 6 6.0 2.0	✓

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 `char` to stdout:

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

This piece of code prints the character *ch*.

Task

You have to print the character, *ch*.

Input Format

Take a character, *ch* as input.

Output Format

Print the character, *ch*.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     char ch;
5     scanf("%c", &ch);
6     printf("%c", ch);
7 }
```

	Input	Expected	Got	
✓	c	c	c	✓

Passed all tests! ✓

Objective

This is a simple challenge to help you practice printing to stdout.

We're starting out by printing the most famous computing phrase of all time! In the editor below, use either `printf` or `cout` to print the string ***Hello, World!*** to stdout.

Input Format

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

Output Format

Print ***Hello, World!*** to stdout.

Sample Output

Hello, World!

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     printf("Hello, World!");
5 }
```

	Expected	Got	
✓	Hello, World!	Hello, World!	✓

Passed all tests! ✓

Sample Input

3 12345678912345 a 334.23 14049.30493

Sample Output

3
12345678912345
a
334.230
14049.304930000

Explanation

Print *int* **3**,
followed by *long* **12345678912345**,
followed by *char* **a**,
followed by *float* **334.23**,
followed by *double* **14049.30493**.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a;
5     long b;
6     char c;
7     float d;
8     double e;
9     scanf("%d %ld %c %f %lf",&a,&b,&c,&d,&e);
10    printf("%d\n",a);
11    printf("%ld\n",b);
12    printf("%c\n",c);
13    printf("%.3f\n",d);
14    printf("%.9lf",e);
15    return 0;
16 }
```

	Input	Expected	Got	
✓	3 12345678912345 a 334.23 14049.30493	3 12345678912345 a 334.230 14049.304930000	3 12345678912345 a 334.230 14049.304930000	✓

Sample Input 2 :

T

7 3 8

Sample Output 2 :

T

6

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     char a;
5     int b,c,d,e;
6     scanf("%c\n",&a);
7     scanf("%d%d%d",&b,&c,&d);
8     printf("%c\n",a);
9     e=(b+c+d)/3;
10    printf("%d",e);
11 }
```

	Input	Expected	Got	
✓	A 3 4 6	A 4	A 4	✓
✓	T 7 3 8	T 6	T 6	✓
✓	R 0 100 99	R 66	R 66	✓

Write a program to print the **ASCII value** and the two adjacent characters of the given character.

Input

E

Output

69

D F

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     char a,b,c;
5     scanf("%c",&a);
6     printf("%d\n",a);
7     b=a-1;
8     c=a+1;
9     printf("%c %c",b,c);
10 }
```

	Input	Expected	Got	
✓	E	69 D F	69 D F	✓

Passed all tests ✓