

# GE23131-Programming Using C-2024

## Quiz navigation

[Show one page at a time](#)[Finish review](#)

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Thursday, 10 October 2024, 10:46 AM
<b>Duration</b>	74 days 6 hours

### Question 1

Correct

Marked out of 3.00

[Flag question](#)

Write a program to input a name (as a single character) and marks of three tests as m1, m2, and m3. The marks have been given in integer format.

Now, you need to calculate the average of the given marks and print it along with the name and average.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to calculate the average in integer and neglect the decimal part.

Input format :

Line 1 : Name(Single character)

Line 2 : Marks scored in the 3 tests separated by single space.

Output format :

First line of output prints the name of the student.

Second line of the output prints the average mark.

Constraints

Marks for each student lie in the range 0 to 100 (both inclusive)

Sample Input 1 :

A  
3 4 6

Sample Output 1 :

A  
4

Sample Input 2 :

T  
7 3 8

Sample Output 2 :

T  
6

REC-CIS

```

2  int main(){
3      char ch;
4      int a,b,c,d;
5      scanf("%c",&ch);
6      scanf("%d %d %d",&a,&b,&c);
7      d=(a+b+c)/3;
8      printf("%c\n%d",ch,d);
9      return 0;
10 }

```

	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	

Passed all tests!

## Question 2

Correct

Marked out of 5.00

[Flag question](#)

Some C data types, their format specifiers, and their most common bit widths are as follows:

- *Int* ("%d"): 32 Bit integer
- *Long* ("%ld"): 64 bit integer
- *Char* ("%c"): Character type
- *Float* ("%f"): 32 bit real value
- *Double* ("%lf"): 64 bit real value

**Reading**

To read a data type, use the following syntax:

```
scanf("format_specifier", &val)
```

For example, to read a *character* followed by a *double*:

```
char ch;
double d;
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

**Printing**

To print a data type, use the following syntax:

```
printf("format_specifier", val)
```

For example, to print a *character* followed by a *double*:

```
char ch = 'd';
double d = 234.432;
printf("%c %lf", ch, d);
```

**Note:** You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a n lines, it is faster to use *scanf* and *printf*.

**Input Format**

Print each element on a new line in the same order it was received as input. Note that the float has 2 decimal places and the double to 9 decimal places.

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     int a;
4     long b;
5     char c;
6     float d;
7     double e;
8     scanf("%d %ld %c %f %lf",&a,&b,&c,&d,&e);
9     printf("%d\n%ld\n%c\n%.3f\n%.9lf",a,b,c,d,e);
10    return 0;
11 }
```

	Input	Expected	Got	
	3 12345678912345 a 334.23 14049.30493	3 12345678912345 a 334.230 14049.304930000	3 12345678912345 a 334.230 14049.304930000	
Passed all tests!				

Question **3**

Correct

Marked out of 7.00

Flag question

Write a program to print the [ASCII value](#) and the two adjacent characters of the given character.

Input

E

Output

REC-CIS

**Answer:** (penalty regime: 0 %)

```
1 #include <stdio.h>
2 int main(){
3     char ch;
4     scanf("%c",&ch);
5     printf("%d\n",ch);
6     printf("%c %c",ch-1,ch+1);
7     return 0;
8 }
```

	Input	Expected	Got	
	E	69 D F	69 D F	

Passed all tests!