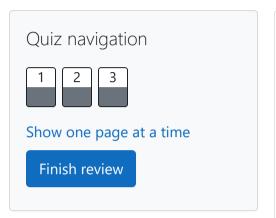
# GE23131-Programming Using C-2024



Status	Finished	
	Monday, 23 December 2024, 5:33 PM	
Completed	Tuesday, 22 October 2024, 2:36 PM	
Duration	62 days 2 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 of a student considering all the three 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 as mentioned in the output format section.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only 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:

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 :	
Т	

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

```
1 #include<stdio.h>
 2 v int main(){
 3
        int a,b,c,avg;
       char d;
       scanf("%c",&d);
       scanf("\n%d%d%d",&a,&b,&c);
       avg=(a+b+c)/3;
       printf("%c",d);
 8
       printf("\n%d",avg);
 9
10
       return 0;
11 }
```

	Input	Expected	Got	
~	A 3 4 6	A 4	A 4	<b>~</b>
~	T 7 3 8	T 6	T 6	<b>~</b>
~	R 0 100 99	R 66	R 66	<b>~</b>

## 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);

use scanf and printf.

#### **Input Format**

Input consists of the following space-separated values: *int, long, char, float,* and *double,* respectively.

#### **Output Format**

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

# **Sample Input**

3 12345678912345 a 334.23 14049.30493

#### **Sample Output**

3

12345678912345

а

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 %)

```
#include<stdio.h>
int main(){
   int a;
   long b;
```

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

```
Output
69
D F
Answer: (penalty regime: 0 %)
  1 #include<stdio.h>
  2 v int main(){
  3
         char ch;
         scanf("%c",&ch);
  4
         printf("%d\n%c %c",ch,ch-1,ch+1);
  6
         return 0;
  7 }
      Input Expected Got
```

