SHERINKATHERINA D 2024-CSE S2

240701495

Week-01-Overview of C, Constants, Variables and Data Types

Week-01-02-Practice Session-Coding

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.

Source Code

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.

Result

	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 million numbers as input and printing a million lines, it is faster to 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.

Source Code

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a;
    long b;
    char c;
    float d;
    double e;
    scanf("%d %ld %c %f %lf",&a,&b,&c,&d,&e);
    printf("%d\n",a);
    printf("%ld\n",b);
    printf("%c\n",c);
    printf("%0.3f\n",d);
    printf("%0.9f",e);
    return 0;


}
```

SHERINKATHERINA D 2024-CSE S2

240701495

Result

	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

69
D F

Source Code

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    char a;
    scanf("%c",&a);
    printf("%d\n%c %c",(int)a,a+-1,a+1);

    return 0;
}
```

Result

SHERINKATHERINA D 2024-CSE S2

240701495

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

Passed all tests! ✓