RITTVIK S 2024-CSE 2116240701616

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

Week-01-01-Practice Session-Coding

Question **1**

Correct

Marked out of 3.00

Flag question

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.

Source code

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    printf("Hello, World!");
    return 0;
}
```

Output

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

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

Question 2
Correct
Marked out of 5.00
Flag question

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***.

Source code

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    char ch;
    scanf("%c",&ch);
    printf("%c", ch);
    return 0;
}
```

Output

	Input	Expected	Got	
✓	c	c	c	✓

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

Question **3**

Correct

Marked out of 7.00

Flag question

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.

Source code

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a,b;
    float c,d;
    scanf("%d%d",&a,&b);
    scanf("%f%f",&c,&d);
    printf("%d ",a+b);
    printf("%d\n",a-b);
    printf("%0.1f ",c+d);
    printf("%0.1f",c-d);
    return 0;
}
```

Output

	Input	Expected	Got	
✓	10 4 4.0 2.0	14 6 6.0 2.0	14 6 6.0 2.0	✓
✓	20 8 8.0 4.0	28 12 12.0 4.0	28 12 12.0 4.0	✓

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

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.

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.

Source code

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    char name;
    int m1,m2,m3,avg;
    scanf("%c",&name);
    scanf("%d",&m1);
    scanf("%d",&m2);
    scanf("%d",&m3);
    avg=(m1+m2+m3)/3;
    printf("%c\n",name);
    printf("%d",avg);
    return 0;
}
```

Output

	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! ✓

Result

The above program is executed successfully and provides the above output.

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:

Source code

```c
#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%ld\n%c\n%.3f\n%.9lf",a,b,c,d,e);
    return 0;
}
```

Output

	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! ✓

Result

The above program is executed successfully and provides the above output.

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

```c
#include<stdio.h>
int main()
{
    char a;
    scanf("%c",&a);
    printf("%d\n%c %c",(int)a,a-1,a+1);
    return 0;
}
```

Output

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

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.