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

F Flag question

Objective

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

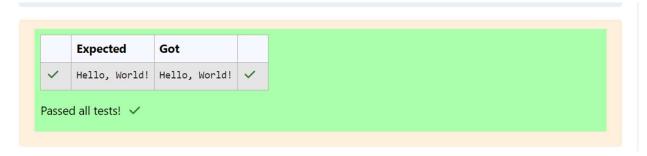
We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string *Hello, World!* to stdout.

Source code

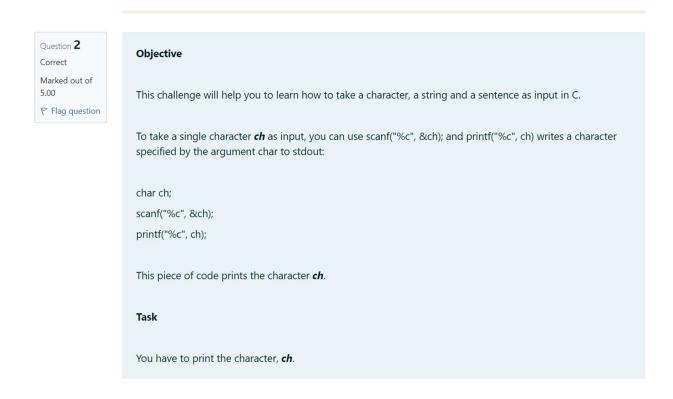
Answer: (penalty regime: 0 %)

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

Output

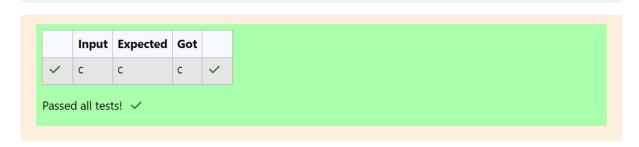


Result



Source code

Output



Result

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.

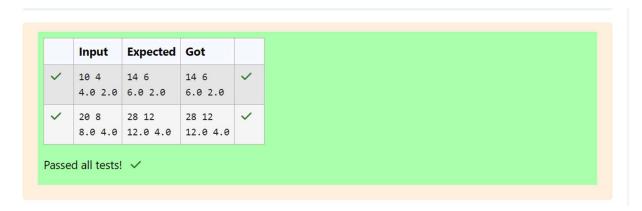
The scanf() function reads the input data from the console. The syntax is scanf("format string",argument_list);. For ex: The scanf("%d",&number) statement reads integer number from the console and stores the given value in variable *number*.

To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where \mathbf{n} and \mathbf{m} are the two integers.

Source code

```
Answer: (penalty regime: 0 %)
     #include<stdio.h>
      int main()
   2
   3 ₹ {
   4
         int a,b;
        float c,d;
   5
         scanf("%d%d",&a,&b);
   6
         scanf("%f%f",&c,&d);
   7
        printf("%d ",a+b);
   8
         printf("%d\n",a-b);
   9
  10
         printf("%0.1f ",c+d);
         printf("%0.1f",c-d);
  11
  12
          return 0;
  13 }
```

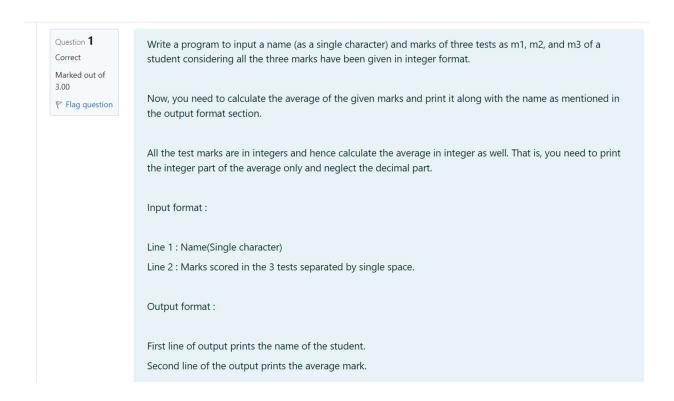
Output



Result

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

Week-01-02-Practice Session-Coding



Source code

```
Answer: (penalty regime: 0 %)
    1 #include<stdio.h>
    2
       int main()
    3 ₹ {
    4
             char name;
        int m1,m2,m3,avg;
scanf("%c",&name);
scanf("%d",&m1);
scanf("%d",&m2);
scanf("%d",&m3);
    5
    6
    7
    8
   9
   10
          avg=(m1+m2+m3)/3;
   11
             printf("%c\n",name);
   12
             printf("%d",avg);
   13
             return 0;
   14 }
```

Output



Result

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

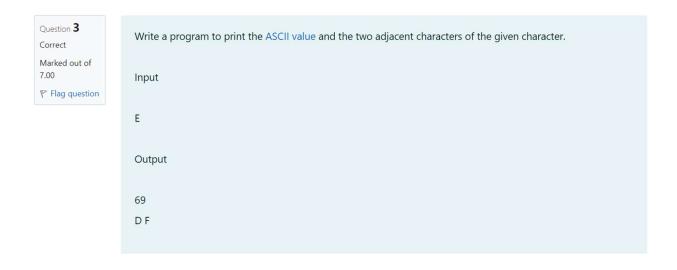
To print a data type, use the following syntax:

Source code

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

Output

Result

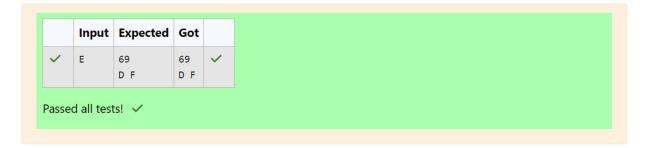


Source code

```
#include<stdio.h>
int main()

char a;
scanf("%c",&a);
printf("%d\n%c %c",(int)a,a-1,a+1);
return 0;
}
```

Output



Result