

REC-CIS

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

Dashboard / My courses / GE23131-PUC-2024 / Week-01-Overview of C, Constants, Variables and Da...

Navigation

Dashboard

Site home

Site pages

My courses

GE23131-PUC-2024

Participants

Competencies

Grades

General

Skill Test-01-MCQ & Coding

Lecture Notes

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

Coding-C-Language Features-

Lecture Notes

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

 Coding-C-Language Features-Optional

✓ Done

 Week-01-01-Practice Session-Coding

✓ Done

 Week-01-02-Practice Session-Coding

✓ Done

Lecture Notes

Jump to...



Programming Using C-2024

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 19 October 2024, 9:02 PM
Duration	64 days 20 hours

Question 1

Correct

Marked out of
1.00

[Flag question](#)

The code given below contains instructions to print the text "**I love Apples**" to the console.

The `\n` in the text "I love Apples\n" ensures that the line breaks after printing the text "I love Apples" (which means that nothing is printed on the same line).

Follow the steps given below to change the text, execute **compile** command and finally **execute** the file :

1. In the code given below, change the text to print "**I love Mangoes**" instead of "**I love Apples**".

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("I love Mangoes");
6     return 0;
7 }
```

GE23131-Programming Using C-2024

Dashboard / My courses / GE23131-PUC-2024 / Week-01-Overview of C, Constants, Variables and Da...
/ Coding-C-Language Features-Optional

Navigation

Dashboard

Site home

Site pages

My courses

GE23131-PUC-2024

Participants

Competencies

Grades

General

Skill Test-01-MCQ
& Coding

Lecture Notes

Week-01-Overview
of C, Constants,
Variables and Da...

Coding-C-
Language

Coding-C-Language Features-Optional

✓ Done

Re-attempt quiz

Time limit: 1 hour

Grading method: Highest grade

Your attempts

Attempt 3

Status Finished

Started Monday, 23 December
2024, 5:33 PM

Completed Saturday, 19 October 2024,

Attempt 2

Status Finished

Started Monday, 23 December
2024, 5:33 PM

Completed Thursday, 10 October



```
#include <stdio.h>
2
3 int main()
4 {
5     printf("Hello C");
6     return 0;
7 }
```

	Expected	Got	
✓	Hello C	Hello C	✓

Passed all tests! ✓

view of C +

secure rajalakshmicolleges.org/moodle/course/section.php?id=2 ☆ H :

5 KAVYASRI V 2024-CSE K2 ▾

Overview of C, Constants, Variables and Data Types

Courses / GE23131-PUC-2024 / Week-01-Overview of C, Constants, Variables and Da...

◀ Lecture Notes ▶ Assessment-01-Overview of C, Constants, Variables and Data Types

[list] Coding-C-Language Features-Optional ✓ Done

[list] Week-01-01-Practice Session-Coding ✓ Done

[list] Week-01-02-Practice Session-Coding ✓ Done

◀ Lecture Notes Jump to...

2024
es
MCQ
es
Variables
-
atures-

h India reported -0.56...

Programming Using C-2024

Status	Finished
Started	Sunday, 12 January 2025, 1:02 PM
Completed	Sunday, 12 January 2025, 1:09 PM
Duration	6 mins 35 secs

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 string **Hello, World!** to stdout.

Input Format

You do not need to read any input in this challenge.

Output Format

NO more attempts are allowed



India reported -0.56...

Print **Hello, World!** to stdout.

Sample Output

Hello, World!

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     printf("Hello, World!");
5     return 0;
6 }
```

Expected

Got

NO more attempts are allowed



India reported -0.56...

✓	Hello, World!	Hello, World!	✓
---	---------------	---------------	---

Passed all tests! ✓

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 `stdout`:

```
char ch;  
scanf("%c", &ch);  
printf("%c", ch);
```

This piece of code prints the character **ch**.

Task

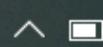
You have to print the character, **ch**.

Input Format

NO more attempts are allowed



India reported -0.56...



```
scanf("%c", &ch);
printf("%c", ch);
```

This piece of code prints the character **ch**.

Task

You have to print the character, **ch**.

Input Format

Take a character, **ch** as input.

Output Format

Print the character, **ch**.

Answer: (penalty regime: 0 %)

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

NO more attempts are allowed



India reported -0.56...

Answer: (penalty regime: 0 %)

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

	Input	Expected	Got	
✓	c	c	c	✓

Passed all tests! ✓

NO more attempts are allowed



India reported -0.56...

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 want to print integer, character, string or float using printf(), then we have to use %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 example, 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 variables.

Task

Your task is to take two numbers of int data type, two numbers of float data type as input and output their sum:

1. Declare **4** variables: two of type int and two of type float.
2. Read **2** lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your variables.
3. Use the + and - operator to perform the following operations:
 - o Print the sum and difference of two int variable on a new line.
 - o Print the sum and difference of two float variable rounded to one decimal place on a new line.

NO more attempts are allowed



India reported -0.56...



- $1 \leq \text{integer variables} \leq 10^4$
- $1 \leq \text{float variables} \leq 10^4$

Output Format

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (up to 1 decimal place) separated by a space on the second line.

Sample Input

10 4

4.0 2.0

Sample Output

14 6

6.0 2.0

Explanation

When we sum the integers **10** and **4**, we get the integer **14**. When we subtract the second number **4** from the first number **10**,

NO more attempts are allowed



India reported -0.56...

Sample Input

10 4

4.0 2.0

Sample Output

14 6

6.0 2.0

Explanation

When we sum the integers **10** and **4**, we get the integer **14**. When we subtract the second number **4** from the first number **10**, we get **6** as their difference.

When we sum the floating-point numbers **4.0** and **2.0**, we get **6.0**. When we subtract the second number **2.0** from the first number **4.0**, we get **2.0** as their difference.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b;
5     float c,d;
6     scanf("%d%d",&a,&b);
7     scanf("%f%f",&c,&d);
```

NO more attempts are allowed



India reported -0.56...



their difference.

When we sum the floating-point numbers **4.0** and **2.0**, we get **6.0**. When we subtract the second number **2.0** from the first number we get **2.0** as their difference.

Answer: (penalty regime: 0 %)

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

	Input	Expected	Got	
✓	10 4 4.0 2.0	14 6 6.0 2.0	14 6 6.0 2.0	✓

NO more attempts are allowed



India reported -0.56...

```
5     int a,b,c,d;
6     scanf("%d%d",&a,&b);
7     scanf("%f%f",&c,&d);
8     printf("%d %d\n",a+b,a-b);
9     printf("%.1f %.1f",c+d,c-d);
10    return 0;
11 }
```

	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! ✓

NO more attempts are allowed



India reported -0.56...

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Thursday, 10 October 2024, 9:31 AM
Duration	74 days 8 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. Now, you need to calculate the average of the given marks and print it along with the name as per the following format section.

Now, you need to calculate the average of the given marks and print it along with the name as per the following format section.

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

Input format :

Line 1 : Name(Single character)



17°C Haze



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



17°C Haze



Sample Output 1 :

A

4

Sample Input 2 :

T

7 3 8

Sample Output 2 :

T

6

Answer: (penalty regime: 0 %)

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



17°C Haze



	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



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(`formatSpecifier`, &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(`formatSpecifier`, val)
```

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

```
char ch = 'd';
```



```
double a = 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 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 values must 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
a
334.230
14049.304930000

Explanation

Print *int* **3**,
followed by *long* **12345678912345**,
followed by *char* **a**,



17°C Haze



ed by float **334.23**,

ed by double **14049.30493**.

er: (penalty regime: 0 %)

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



17°C Haze



	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



Humi

```
1 #include<stdio.h>
2 int main()
3 {
4     char a;
5     scanf("%c",&a);
6     printf("%d\n",a);
7     printf("%c %c",a-1,a+1);
8     return 0;
9 }
10 }
```

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

Passed all tests! ✓



17°C Haze



Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("I love Mangoes");
6     return 0;
7 }
```

	Expected	Got	
✓	I love Mangoes	I love Mangoes	✓

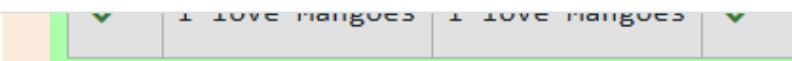
Passed all tests! ✓

Question 2

Given below is a simple program written in C language.



29°C Haze



Passed all tests! ✓

Question 2

Correct

Marked out of
1.00[Flag question](#)

Given below is a simple program written in **C** language.

Change the text in the code given below to make the program print "**Hello C**" instead of "**Hello B**".

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello C");
6     return 0;
7 }
```



```
1 //include <stdio.h>
2
3 int main()
4 {
5     printf("Hello C");
6     return 0;
7 }
```

	Expected	Got	
✓	Hello C	Hello C	✓

Passed all tests! ✓

