Rohan is learning pointers in C language and he has made a program to swap the value of 3 given integers but the code has some problems He asks for your help to correct this.

Explanation:

1.Do not change the whole logic of the code.

2.Program is to swap the value of 2 variables.

3.The inputted values are a and b by space separation and the output is a and b again by space separation.

4.For more visual representation observe the test cases.

Constraints:

Correct code:

#include <stdio.h>

void SwapTwoNumber(int \*a, int \*b)

{

if(\*a == \*b) // Check if the two addresses are same

return;

\*a = \*a ^ \*b;

\*b = \*a ^ \*b;

\*a = \*a ^ \*b;

}

int main()

{

int x,y;

scanf("%d %d",&x,&y);

SwapTwoNumber(&x, &y);

printf("%d %d",x,y);

return 0;

}

TEST CASE

INPUT OUTPUT

1. 1 2 2 1

2. 3 7 7 3

3. -1 -5 -5 -1

BUG CODE:

#include <stdio.h>

void SwapTwoNumber(int \*a, int \*b)

{

if(\*a == \*b) // Check if the two addresses are same

return;

\*a = \*a & \*b;

\*b = \*a & \*b;

\*a = \*a & \*b;

}

int main()

{

int x,y;

scanf("%d %d",&x,&y);

SwapTwoNumber(&x, &y);

printf("%d %d",x,y);

return 0;

}

2.Ram made a program to convert a decimal function to binary using recursion, but the code has some bugs in it. Please correct this.

EXPLANATION:

1.Do not change the whole logic of the code.

2.The data type used in the variables are not subjected to change.

3.Code takes a decimal and converts it to its equivalent binary number.

CONSTRAINTS:

CORRECT CODE:

#include<stdio.h>

long convertBinary(int);

long binaryNo,r,factor = 1;

int main()

{

long binaryNo;

int decimalNo;

scanf("%d",&decimalNo);

binaryNo = convertBinary(decimalNo); //call the function convertBinary

printf("%ld\n",binaryNo);

return 0;

}

long convertBinary(int decimalNo)

{

if(decimalNo != 0)

{

r = decimalNo % 2;

binaryNo = binaryNo + r \* factor;

factor = factor \* 10;

convertBinary(decimalNo / 2); //calling the function convertBinary itself recursively

}

return binaryNo;

}

TEST CASES:

SAMPLE INPUT SAMPLE OUTPUT

1.23 10111

2.34 10010

3.0 0

BUG CODE:

#include<stdio.h>

long convertBinary(int);

int main()

{

long binaryNo;

int decimalNo;

scanf("%d",&decimalNo);

binaryNo = convertBinary(decimalNo);//call the function convertBinary

printf("%ld\n",binaryNo);

return 0;

}

long convertBinary(int decimalNo)

{

long binaryNo,r,factor = 1;

if(decimalNo != 0)

{

r = decimalNo % 2;

binaryNo = binaryNo + r \* factor;

factor = factor \* 10;

convertBinary(decimalNo / 2); //calling the function convertBinary itself recursively

}

return binaryNo;

}

3.Raj made a code to find whether the given number is prime or not, but the code is not working properly, he asks for your help to fix it.

EXPLANATION:

1.Do not change the whole logic of the code.

2.Basic knowledge of python built in functions is required.

CONSTRAINTS:

CORRECT CODE:

code\_str = """

num = int(input())

# define a flag variable

flag = False

# prime numbers are greater than 1

if num > 1:

# check for factors

for i in range(2, num):

if (num % i) == 0:

# if factor is found, set flag to True

flag = True

# break out of loop

break

# check if flag is True

if flag:

print("NO")

else:

print("YES")"""

code = compile(code\_str,'',"exec")

exec(code)

TEST CASE:

SAMPLE INPUT SAMPLE OUTPUT

1.34 NO

2.3 YES

3.7 YES

BUG CODE

code\_str = """

num = int(input())

# define a flag variable

flag = False

# prime numbers are greater than 1

if num > 1:

# check for factors

for i in range(2, num):

if (num % i) == 0:

# if factor is found, set flag to True

flag = True

# break out of loop

break

# check if flag is True

if flag:

print("NO")

else:

print("YES")"""

code = compile(“code\_str”,” '',"eval")

exec(code)