Q1. Write a c program to clear the nth bit of a number input by the user.

**Program:**

#include <stdio.h>

int main()

{

int num, n, newNum;

printf("Enter any number: ");

scanf("%d", &num);

printf("Enter nth bit to clear (0-31): ");

scanf("%d", &n);

newNum = num & (~(1 << n));

if(n>31)

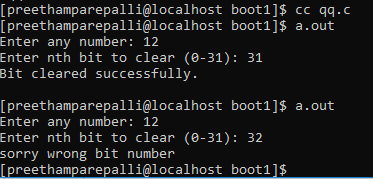
printf(“sorry wrong bit number\n”);

else

printf("Bit cleared successfully.\n\n");

return 0;

}

**Output:**

Q2. Write a c program to swap two number using bitwise operator.

**Program:**

#include <stdio.h>

void swap(int\*, int \*);

void main()

{

int num1, num2;

printf("\nInput two numbers:");

scanf("%d %d", &num1, &num2);

printf(" numbers before swap %d %d", num1, num2);

swap(&num1, &num2);

printf("\n numbers after swap %d %d\n", num1, num2);

}

void swap(int \*x, int \*y)

{

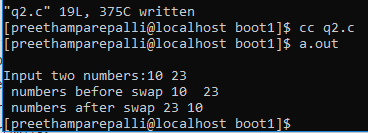
\*x = \*x ^ \*y;

\*y = \*x ^ \*y;

\*x = \*x ^ \*y;

}

**OUTPUT:**



Q4. Print the following out-put:

1

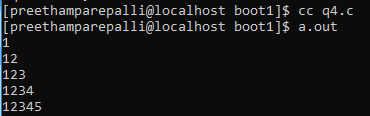
1 2

1 2 3

1 2 3 4

1 2 3 4 5

Output:



Q6. Read a number and print whether the number is palindrome or not. If the input number exceeds the upper bound of integer type, then re-prompt another integer.

**Program:**

#include <stdio.h>

int main()

{

int n, reversedInteger = 0, remainder, originalInteger;

printf("Enter an integer: ");

scanf("%d", &n);

originalInteger = n;

while( n!=0 )

{

remainder = n%10;

reversedInteger = reversedInteger\*10 + remainder;

n /= 10;

}

if (originalInteger == reversedInteger)

printf("%d is a palindrome.", originalInteger);

else

printf(" is not a palindrome.");

return 0;

}

**Output:**

