|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.  No. | | C PROGRAMMING PROBLEMS | Page  No. | Teacher’s  Signature |
| 1 | | Write a C program to swap two numbers using bitwise operator. |  |  |
| 2 | | Write a C program to check whether a number is even or odd using bitwise operator. |  |  |
| 3 | | Write a C program to check whether a number is even or odd using switch case. |  |  |
| 4 | | Write a C program to check whether a number is positive, negative or zero using switch case. |  |  |
| 5 | | Write a C program to create Simple Calculator using switch case. |  |  |
| 6 | | Write a C program to print all natural numbers from 1 to n. - using while loop |  |  |
| 7 | | Write a C program to print all natural numbers in reverse (from n to 1). - using while loop |  |  |
| 8 | | Write a C program to print all alphabets from a to z. - using while loop |  |  |
| 9 | | Write a C program to print all odd number between 1 to 100. |  |  |
| 10 | | Write a C program to find sum of all natural numbers between 1 to n. |  |  |
| 11 | | Write a C program to find all even numbers between 1 to 100. |  |  |
| 12 | | Write a C program to print multiplication table of any number. |  |  |
| 13 | | Write a C program to count number of digits in a number. |  |  |
| 14 | | Write a C program to find first and last digit of a number. |  |  |
| 15 | | Write a C program to calculate sum of digits of a number. |  |  |
| 16 | | Write a C program to enter a number and print its reverse. |  |  |
| 17 | | Write a C program to check whether a number is palindrome or not. |  |  |
| 18 | | Write a C program to find frequency of each digit in a given integer. |  |  |
| 19 | | Write a C program to enter a number and print it in words. |  |  |
| 20 | | Write a C program to print all ASCII character with their values. |  |  |
| 21 | | Write a C program to find power of a number using for loop. |  |  |
| 22 | | Write a C program to find all factors of a number. |  |  |
| 23 | | Write a C program to calculate factorial of a number. |  |  |
| 24 | | \*\*\*\*\*Write a C program to find HCF (GCD) of two numbers.  \*\*At the end\*\* |  |  |
| 25 | | Write a C program to check whether a number is Prime number or not. |  |  |
| 26 | | Write a C program to print all Prime numbers between 1 to n. |  |  |
| 27 | | Write a C program to find sum of all prime numbers between 1 to n. |  |  |
| 28 | | Write a C program to find all prime factors of a number. |  |  |
| 29 | | Write a C program to check whether a number is Armstrong number or not. |  |  |
| 30 | | Write a C program to print all Armstrong numbers between 1 to n. |  |  |
| 31 | | Write a C program to check whether a number is Perfect number or not. |  |  |
| 32 | | Write a C program to print all Perfect numbers between 1 to n. |  |  |
| 33 | | Write a C program to check whether a number is Strong number or not. |  |  |
| 34 | | Write a C program to print all Strong numbers between 1 to n. |  |  |
| 35 | Write a C program to print Fibonacci series up to n terms. | |  |  | |
| 36 | Write a C program to find HCF (GCD) of two numbers. | |  |  | |

C CODE:

// Write a C program to swap two numbers using bitwise operator.

#include <stdio.h>

int main(){

int a, b;

printf("Enter two number: \n");

scanf("%d%d", &a,&b);

printf("Before using bitwise operator %d and %d\n", a,b);

a=a^b;

b=a^b;

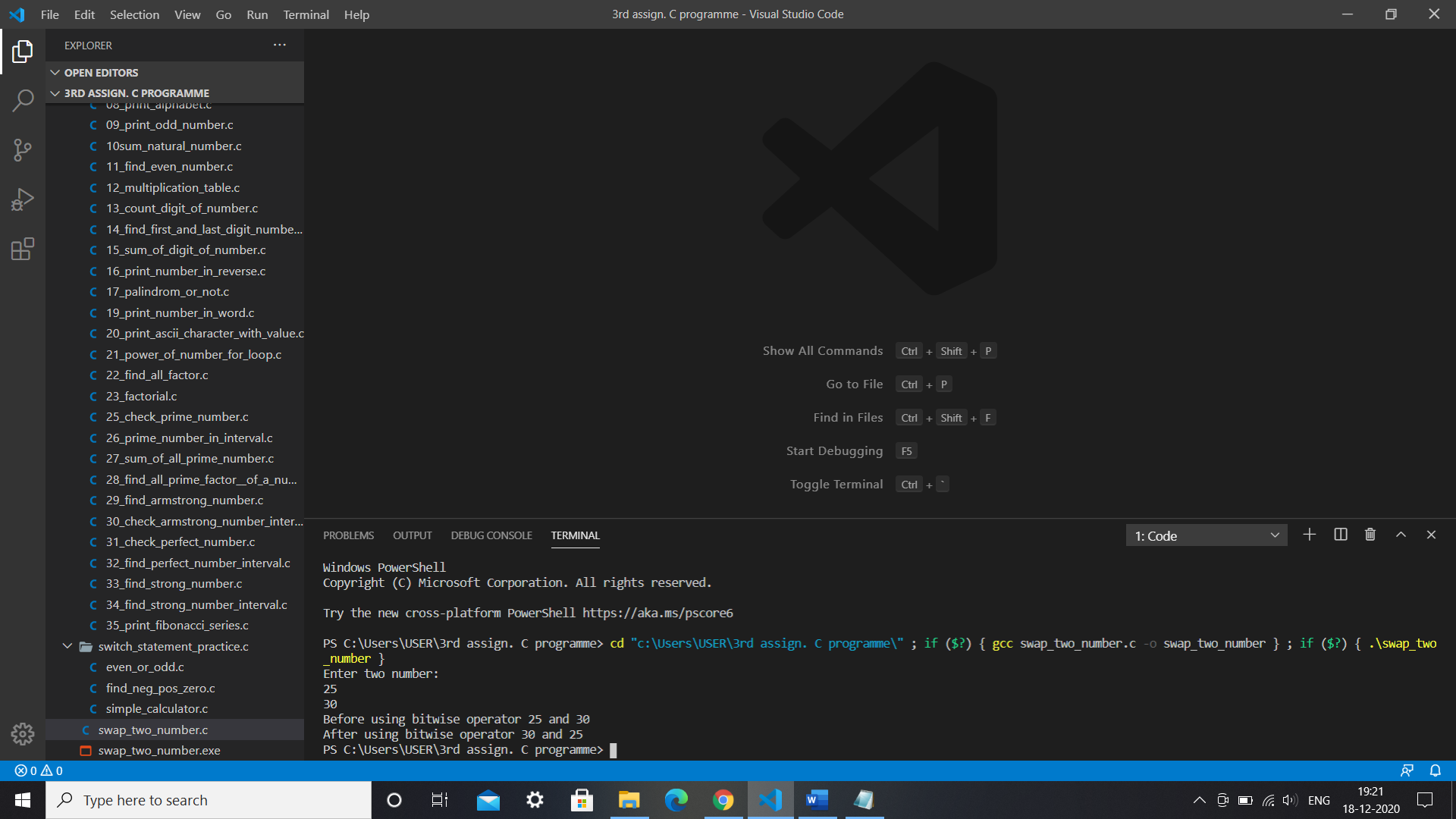
a=a^b;

printf("After using bitwise operator %d and %d", a,b);

return 0;

}

OUTPUT:



C CODE: Write a C program to check whether a number is even or odd using bitwise operator.

#include <stdio.h>

int main()

{

    int num;

    printf("Enter a number:\n");

    scanf("%d", &num);

    if (num & 1)

    {

        printf("This is a odd number.\n");

    }

    else

    {

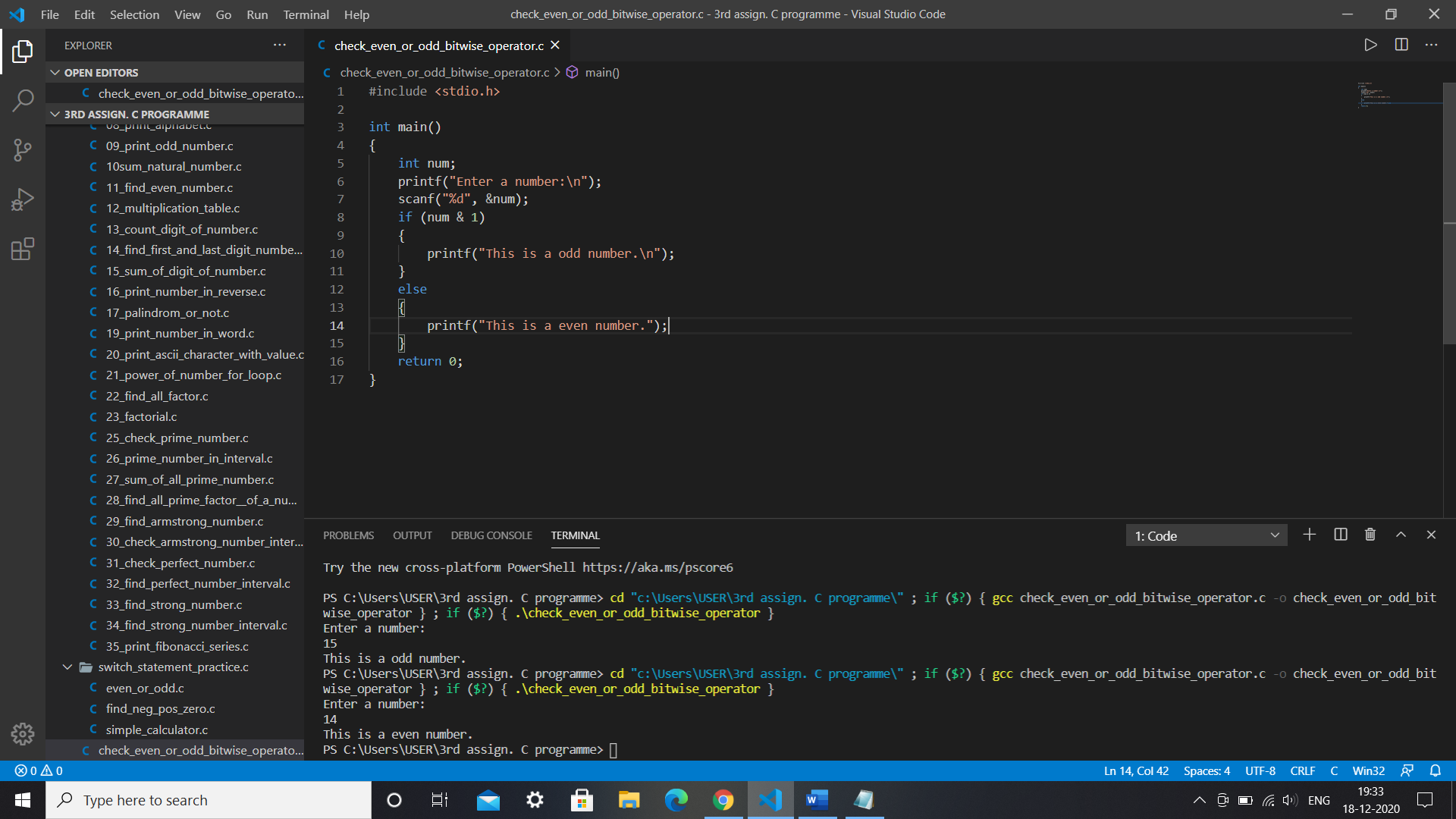
        printf("This is a even number.");

    }

    return 0;

}

OUTPUT:



C CODE:

//Write a C program to check whether a number is even or odd using switch case

#include <stdio.h>

int main(){

int number;

printf("Enter a number: \n");

scanf("%d", & number);

switch (number%2)

{

case 0:

printf("This is even number.");

    break;

case 1:

printf("This is odd number.");

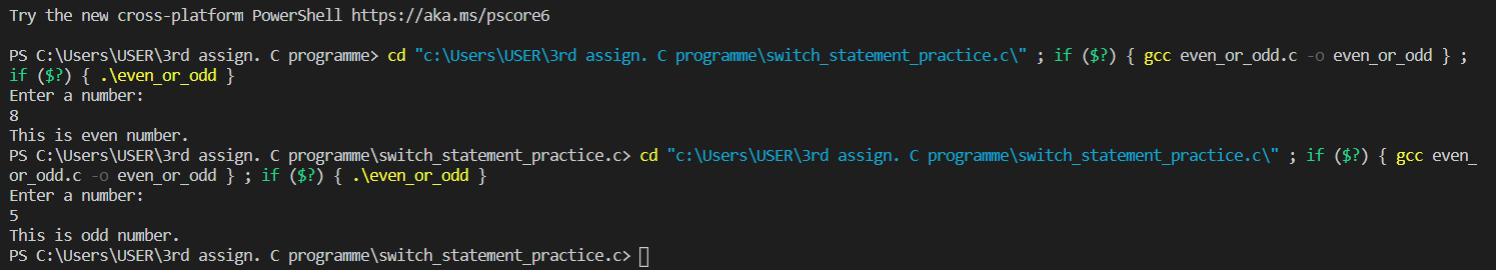
    break;

}

return 0;

}

OUTPUT:



* Write a C program to check whether a number is positive, negative or zero using switch case.

C CODE:

int main()

{

    int num;

    printf("Enter a number: \n");

    scanf("%d", &num);

    switch (num > 0)

    {

    case 1:

        printf("positive");

        break;

    case 0:

        switch (num < 0)

        {

        case 1:

            printf("negative");

            break;

        case 0:

            printf("zero");

            break;

        }

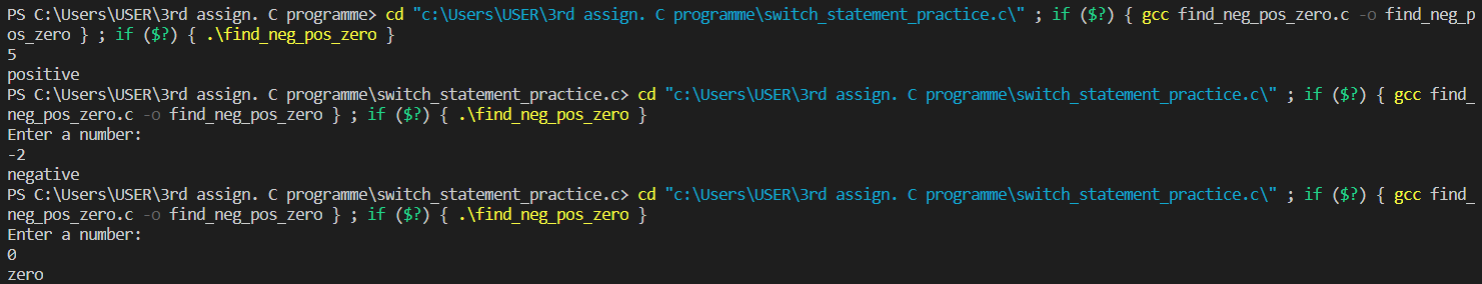
        break;

    }

    return 0;

}

OUTPUT:



* Write a C program to create Simple Calculator using switch case.

C CODE:

#include <stdio.h>

int main(){

char operator;

float a,b;

printf("Enter:: (number1)::(+ - \* /)::(number2)\n");

scanf("%f %c %f", &a, &operator, &b);

switch (operator){

case '+':

printf("Summation=%.2f", a+b);

    break;

case '-':

printf("Substraction=%.2f", a-b);

break;

case '\*':

printf("Multiplication=%.2f", a\*b);

break;

case '/':

printf("Division=%.2f", a+b);

break;

default:

printf("Error! operator is mismatch.");

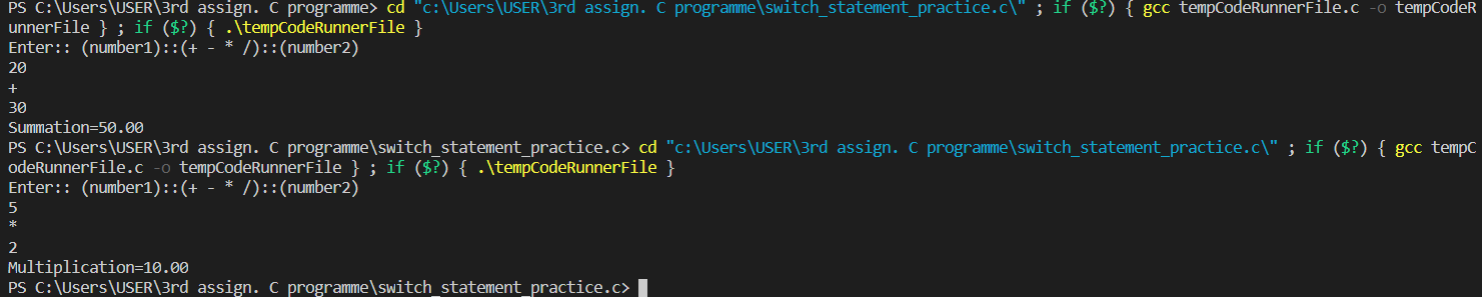
    break;

}

return 0;

}

OUTPUT:



* Write a C program to print all natural numbers from 1 to n. - using while loop.

C CODE:

#include <stdio.h>

int main(){

int N;

printf("Enter the value of N (natural number):\n");

scanf("%d", &N);

int i=0;

while(i<=N){

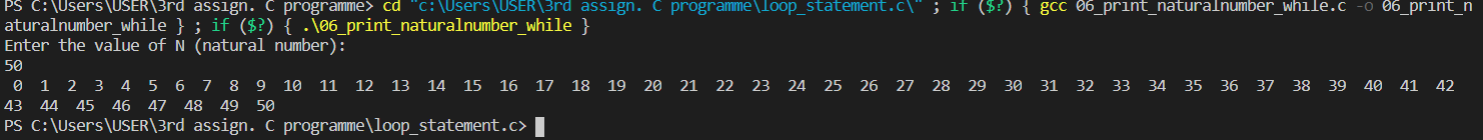
printf(" %d ", i); i++;

}

return 0;

}

OUTPUT:



* Write a C program to print all natural numbers in reverse (from n to 1). - using while loop.

C CODE:

#include <stdio.h>

int main()

{

    int N;

    printf("Enter the value of N (natural number):\n");

    scanf("%d", &N);

    printf(" N (natural number) in reverse:\n");

    while (N >= 1)

    {

        printf("%d ", N);

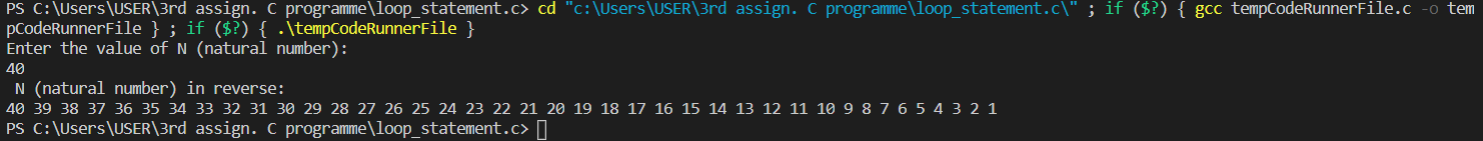
        N--;

    }

    return 0;

}

OUTPUT:



* Write a C program to print all alphabets from a to z. - using while loop.

C CODE:

#include <stdio.h>

int main(){

int i;

printf("Enter a number to print alphabet (between 65-90 for capital and 97-122 for small):: \n");

scanf("%d", &i);

while((i>=65 && i<=90)||(i>=97 && i<=122)){

printf("%c ", i);

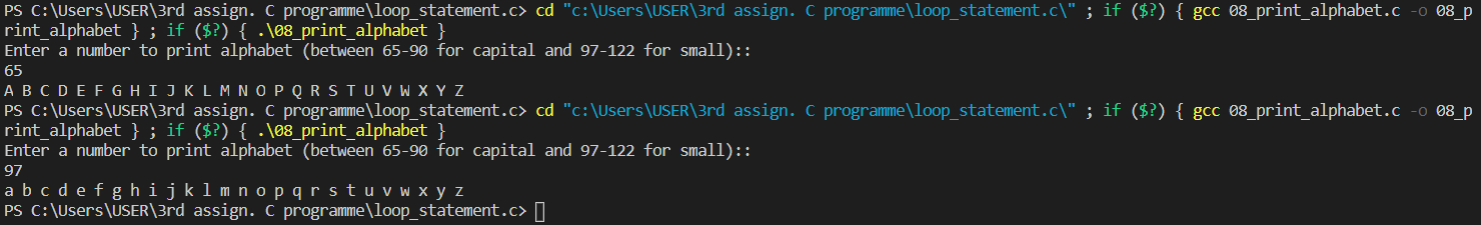
i++;

}

return 0;

}

OUTPUT:



* Write a C program to print all odd number between 1 to 100.

C CODE:

#include <stdio.h>

int main(){

int i=1;

printf("All odd number between 1 to 100 :: \n");

while(i<=100){

if (i%2!=0){

printf("%d ", i);

}

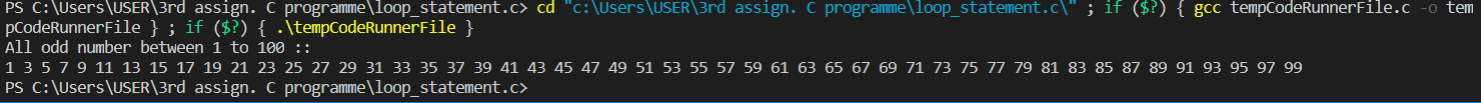
i++;

}

return 0;

}

OUTPUT:



* Write a C program to find sum of all natural numbers between 1 to n.

C CODE:

#include <stdio.h>

int main()

{

    int i, n, sum = 0;

    printf("Enter the value of N (first natural number) :\n");

    scanf("%d", &n);

    for (i = 1; i <= n; i++)

    {

        sum += i;

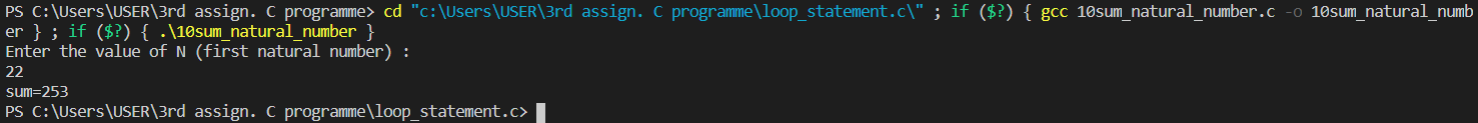
    }

    printf("sum=%d\n", sum);

    return 0;

}

OUTPUT:



* Write a C program to find all even numbers between 1 to 100.

C CODE:

#include <stdio.h>

int main(){

int i;

printf(":ALL EVEN NUMBER BETWEEN 1 TO 100:\n");

for ( i = 1; i <= 100; i++){

if(i%2==0){

printf("%d ", i );

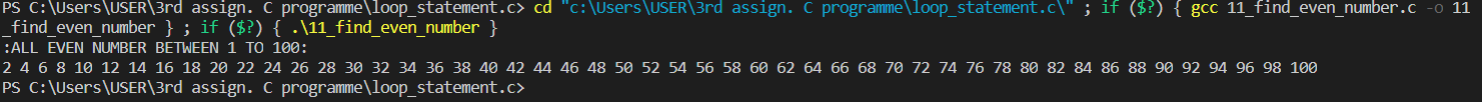
}

}

return 0;

}

OUTPUT:



* Write a C program to print multiplication table of any number.

C CODE:

#include <stdio.h>

int main(){

int n,i,mul=1;

printf("Enter your table number:\n");

scanf("%d", &n);

for ( int i = 1; i <= 10; i++){

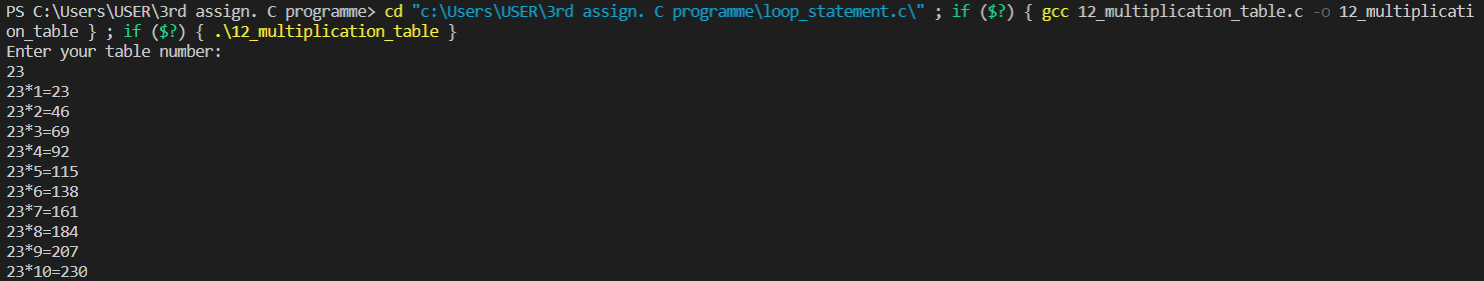
mul= n\*i;

printf("%d\*%d=%d\n", n,i,mul);

}

return 0;

}

OUTPUT:

* Write a C program to count number of digits in a number.

C CODE:

#include <stdio.h>

int main()

{

    int num, count = 0;

    printf("Enter a number::\n");

    scanf("%d", &num);

    while (num != 0)

    {

        num = num / 10;

        count++;

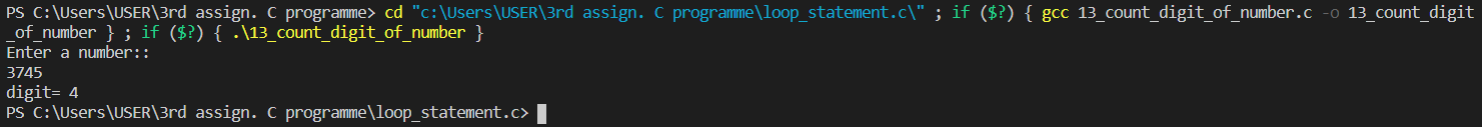
    }

    printf("digit= %d", count);

    return 0;

}

OUTPUT:



* Write a C program to find first and last digit of a number.

C CODE:

#include <stdio.h>

int main()

{

    int num, rem;

    printf("Enter a number:\n");

    scanf("%d", &num);

    rem = num % 10;

    printf("The last digit of the number is %d\n", rem);

    while (num >= 10)

    {

        num = num / 10;

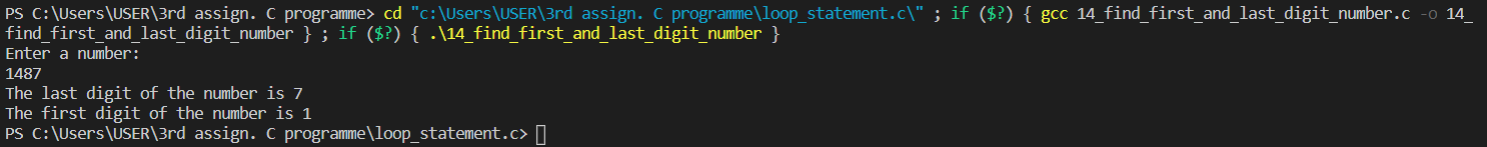
    }

    printf("The first digit of the number is %d\n", num);

    return 0;

}

OUTPUT:



* Write a C program to calculate sum of digits of a number.

C CODE:

#include <stdio.h>

int main()

{

    int num, sum = 0, rem;

    printf("Enter a number:\n");

    scanf("%d", &num);

    while (num != 0)

    {

        rem = num % 10;

        sum = sum + rem;

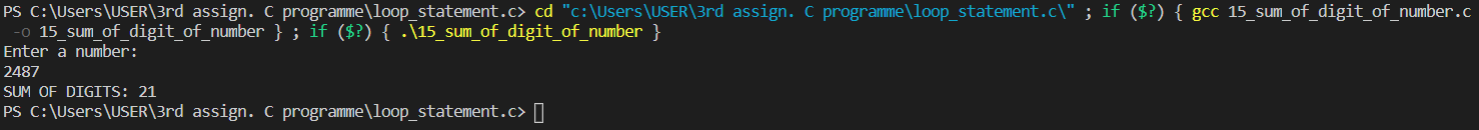
        num = num / 10;

    }

    printf("SUM OF DIGITS: %d", sum);

    return 0;

}

OUTPUT: 

* Write a C program to enter a number and print its reverse.

C CODE:

#include <stdio.h>

int main()

{

    int num, rem, rev = 0;

    printf("Enter a  number: ");

    scanf("%d", &num);

    while (num != 0)

    {

        rem = num % 10;

        // rev = rev \* 10 + rem; // or use this logic.

        num = num / 10;

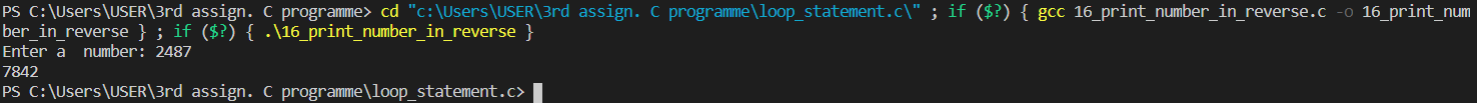
        printf("%d", rem);

    }

    return 0;

}

OUTPUT:



* Write a C program to check whether a number is palindrome or not.

C CODE:

#include <stdio.h>

int main()

{

    int temp, num, rem, rev = 0;

    printf("Enter the number:\n");

    scanf("%d", &num);

    temp = num;

    while (temp != 0)

    {

        rem = temp % 10;

        rev = rev \* 10 + rem;

        temp = temp / 10;

    }

    if (rev == num)

    {

        printf("%d is the palindrome number.\n", num);

    }

    else

    {

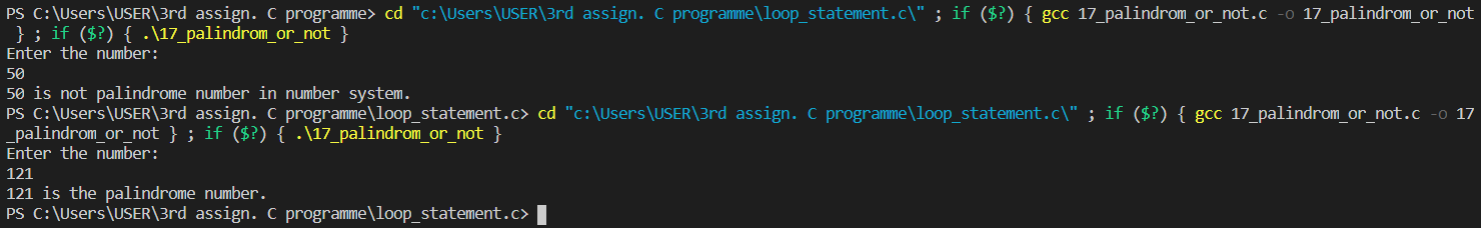
        printf("%d is not palindrome number in number system.\n", num);

    }

    return 0;

}

OUTPUT:



* Write a C program to find frequency of each digit in a given integer.

C CODE:

* Write a C program to enter a number and print it in words.

C code:

#include <stdio.h>

int main()

{

    int N, num = 0;

    printf("Enter a number:\n");

    scanf("%d", &N);

    while (N != 0)

    {

        num = (num \* 10) + (N % 10);

        N /= 10;

    }

    while (num != 0)

    {

        switch (num % 10)

        {

        case 0:

            printf("Zero ");

            break;

        case 1:

            printf("One ");

            break;

        case 2:

            printf("Two ");

            break;

        case 3:

            printf("Three ");

            break;

        case 4:

            printf("Four ");

            break;

        case 5:

            printf("Five ");

            break;

        case 6:

            printf("Six ");

            break;

        case 7:

            printf("Seven ");

            break;

        case 8:

            printf("Eight ");

            break;

        case 9:

            printf("Nine ");

            break;

        }

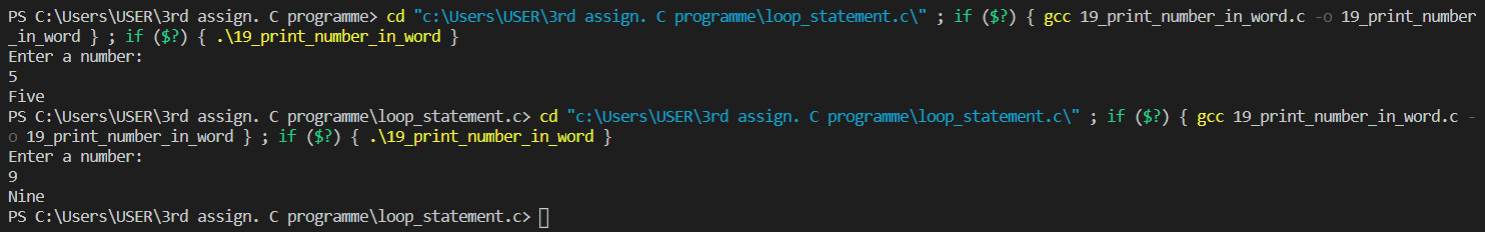
            num = num / 10;

    }

    return 0;

}

OUTPUT:



* Write a C program to print all ASCII character with their values.

C CODE:

#include <stdio.h>

int main()

{

    printf("-----------ASCII CHARECTER------------");

    for (int i = 0; i < 256; i++)

    {

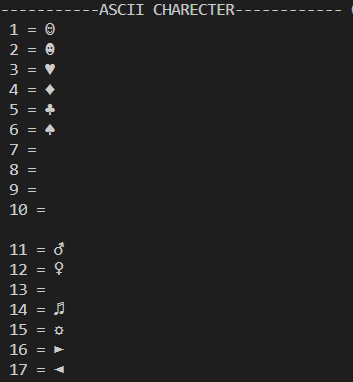
        printf(" %d = %c\n", i, i);

    }

    return 0;

}

OUTPUT:



* Write a C program to find power of a number using for loop.

C CODE:

#include <stdio.h>

int main(){

int num,index;

int pow=1;

printf("Enter the number and index limit:\n");

scanf("%d%d", &num,&index);

for (int i = 1; i <= index; i++)

{

pow=pow\*num;

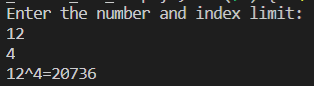
}

printf("%d^%d=%d\n", num,index,pow);

return 0;

}

OUTPUT:



* Write a C program to find all factors of a number.

C CODE:

#include <stdio.h>

int main()

{

    int num;

    printf("Enter a number:\n");

    scanf("%d", &num);

    printf("Factors:\n");

    for (int i = 1; i <= num; i++)

    {

        if (num % i == 0)

        {

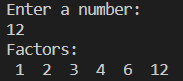
            printf(" %d ", i);

        }

    }

    return 0;

}

OUTPUT: 

* Write a C program to calculate factorial of a number.

C CODE:

#include <stdio.h>

int main()

{

    int num, fact = 1, i;

    printf("Enter a number:\n");

    scanf("%d", &num);

    for (i = 1; i <= num; i++)

    {

        fact = fact \* i;

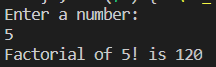
    }

    printf("Factorial of %d is %d", num, fact);

    return 0;

}

OUTPUT:



* Write a C program to check whether a number is Prime number or not.

C CODE: #include <stdio.h>

int main()

{

    int num, count = 0;

    printf("Enter a number:\n");

    scanf("%d", &num);

    for (int i = 1; i < num; i++)

    {

        if (num % i == 0)

        {

            count++;

        }

    }

    if (count > 2)

    {

        printf("This is not a prime number.\n");

    }

    else

    {

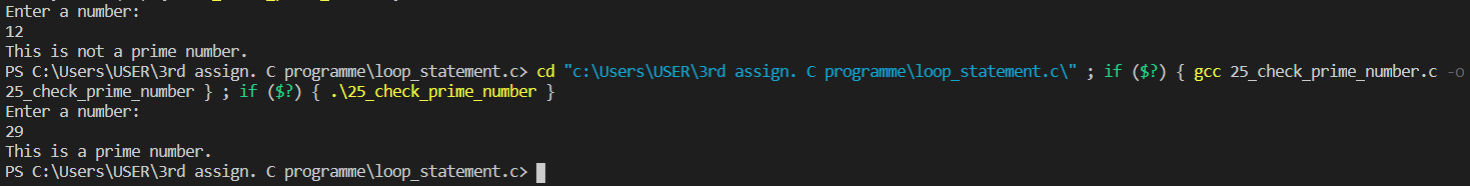
        printf("This is a prime number.\n");

    }

    return 0;

}

OUTPUT:



* Write a C program to print all Prime numbers between 1 to n.

C CODE:

#include <stdio.h>

int main()

{

    int n, i, f, lim;

    printf("Enter a limit:\n");

    scanf("%d", &lim);

    printf("Prime Numbers: ");

    for (n = 1; n <= lim; n++)

    {

        f = 1;

        for (i = 2; i <= n / 2; i++)

        {

            if (n % i == 0)

            {

                f = 0;

                break;

            }

        }

        if (f)

        {

            printf(" %d", n);

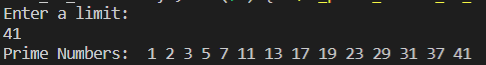
        }

    }

    return 0;

}

OUTPUT:



* Write a C program to find sum of all prime numbers between 1 to n.

C CODE:

#include <stdio.h>

int main()

{

    int n, i, f, lim, sum = 0;

    printf("Enter a limit:\n");

    scanf("%d", &lim);

    printf("All prime numbers:\n");

    for (n = 1; n <= lim; n++)

    {

        f = 1;

        for (i = 2; i <= n / 2; i++)

        {

            if (n % i == 0)

            {

                f = 0;

                break;

            }

        }

        if (f)

        {

            printf(" %d", n);

            sum = sum + n;

        }

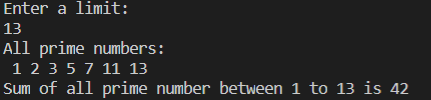
    }

    printf("\nSum of all prime number between 1 to %d is %d", lim, sum);

    return 0;

}

OUTPUT:



* Write a C program to find all prime factors of a number.

C CODE:

#include <stdio.h>

int main()

{

    int num, i, j, count = 0;

    printf("Enter a number:\n");

    scanf("%d", &num);

    printf("Prime Factors:\n");

    for (i = 1; i <= num; i++)

    {

        if (num % i == 0)

        {

            count = 0;

            for (j = 1; j <= i; j++)

            {

                if (i % j == 0)

                {

                    count++;

                }

            }

            if (count == 2)

            {

                printf(" %d", i);

            }

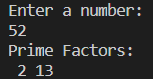
        }

    }

    return 0;

}

OUTPUT:



* Write a C program to check whether a number is Armstrong number or not.

C CODE:

#include <stdio.h>

int main()

{

    int num, temp, rem, sum = 0;

    printf("Enter a number:\n");

    scanf("%d", &num);

    temp = num;

    while (temp != 0)

    {

        rem = temp % 10;

        sum = sum + (rem \* rem \* rem);

        temp = temp / 10;

    }

    if (sum == num)

    {

        printf(" This  is a armstrong number.");

    }

    else

    {

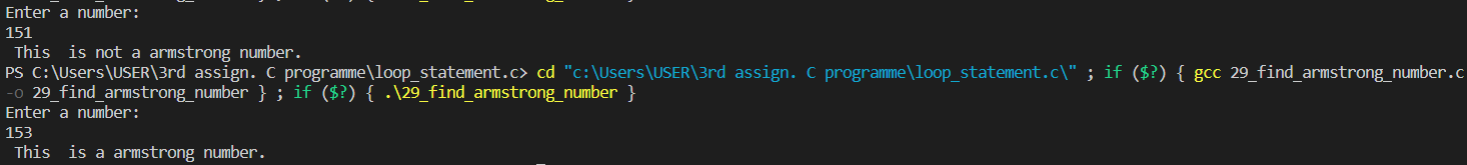
        printf(" This  is not a armstrong number.");

    }

    return 0;

}

OUTPUT:



* Write a C program to print all Armstrong numbers between 1 to n.

C CODE:

#include <stdio.h>

int main()

{

    int num, temp, rem, i, sum;

    printf("Enter a upper limit :\n");

    scanf("%d", &num);

    printf("All armstrong number between 1 to %d :\n", num);

    for (i = 1; i <= num; i++)

    {

        temp = i;

        sum = 0;

        while (temp > 0)

        {

            rem = temp % 10;

            sum = sum + (rem \* rem \* rem);

            temp = temp / 10;

        }

        if (sum == i)

        {

            printf(" %d", i);

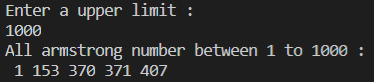
        }

    }

    return 0;

}

OUTPUT:



* Write a C program to check whether a number is Perfect number or not.

C CODE:

#include <stdio.h>

int main()

{

    int num, temp, sum = 0;

    printf("Enter a number:\n");

    scanf("%d", &num);

    for (int i = 1; i < num; i++)

    {

        temp = num;

        if (temp % i == 0)

        {

            sum = sum + i;

        }

    }

    if (sum == num)

    {

        printf("This is a perfect number.");

    }

    else

    {

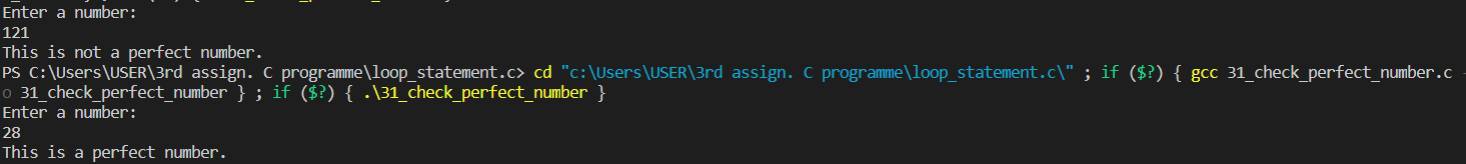
        printf("This is not a perfect number.");

    }

    return 0;

}

OUTPUT:



* Write a C program to print all Perfect numbers between 1 to n.

C CODE:

#include <stdio.h>

int main()

{

    int limit, temp, sum;

    printf("Enter upper limit:\n");

    scanf("%d", &limit);

    printf("Perfect Numbers:");

    for (int j = 1; j <= limit; j++)

    {

        sum = 0;

        for (int i = 1; i < j; i++)

        {

            temp = j;

            if (temp % i == 0)

            {

                sum = sum + i;

            }

        }

        if (sum == j)

        {

            printf(" %d,", j);

        }

    }

    return 0;

}

OUTPUT:



* Write a C program to check whether a number is Strong number or not.

C CODE:

#include <stdio.h>

int fact(int x);

int main()

{

    int n, rem, sum, temp;

    printf("Enter a number:\n");

    scanf("%d", &n);

    temp = n;

    sum = 0;

    while (temp != 0)

    {

        rem = temp % 10;

        sum = sum + fact(rem);

        temp = temp / 10;

    }

    if (sum == n)

    {

        printf("This is a strong number.\n");

    }

    else

    {

        printf("This is not a strong number.\n");

    }

    return 0;

}

int fact(int x)

{

    int result;

    if (x == 1)

    {

        return 1;

    }

    else

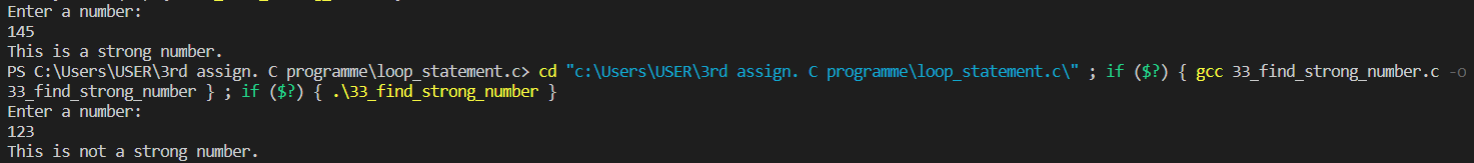
    {

        return (x \* fact(x - 1));

    }

}

OUTPUT:



* Write a C program to print all Strong numbers between 1 to n.

C CODE:

#include <stdio.h>

int main()

{

    int n, rem, sum, temp, fact;

    printf("Enter upper limit:\n");

    scanf("%d", &n);

    printf("Strong Numbers:");

    for (int i = 1; i <= n; i++)

    {

        temp = i;

        sum = 0;

        while (temp > 0)

        {

            rem = temp % 10;

            fact = 1;

            for (int j = 1; j <= rem; j++)

            {

                fact = fact \* j;

            }

            sum = sum + fact;

            temp = temp / 10;

        }

        if (sum == i)

        {

            printf(" %d", i);

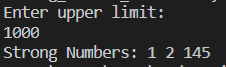
        }

    }

    return 0;

}

OUTPUT:



* Write a C program to print Fibonacci series up to n terms.

C CODE:

#include <stdio.h>

int main()

{

    int n, a = 0, b = 1, c = 0;

    printf("Enter upper limit:\n");

    scanf("%d", &n);

    while (c <= n)

    {

        printf(" %d", c);

        a = b;

        b = c;

        c = a + b;

    }

    return 0;

}

OUTPUT:



* Write a C program to find HCF (GCD) of two numbers.

C CODE:

#include <stdio.h>

int main()

{

    int n1, n2, gcd;

    printf("Enter two numbers:\n");

    scanf("%d%d", &n1, &n2);

    for (int i = 1; i <= n1 && i <= n2; i++)

    {

        if (n1 % i == 0 && n2 % i == 0)

        {

            gcd = i;

        }

    }

    printf("GCD of %d and %d is: %d",n1,n2,gcd);

    return 0;

}

OUTPUT:

