//1

#include <stdio.h>

int sum2(int num1, int num2) {

  return num1 + num2;

}

void main() {

  printf("Enter n1 = ");

  int num1 = 0;

  scanf("%d", &num1);

  printf("Enter n2 = ");

  int num2 = 0;

  scanf("%d", &num2);

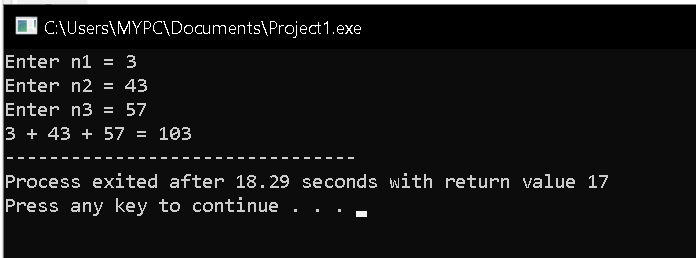
  printf("Enter n3 = ");

  int num3 = 0;

  scanf("%d", &num3);

  printf("%d + %d + %d = %d",num1, num2, num3, sum2(sum2(num1, num2), num3));

}



//2

#include <stdio.h>

int digitSum(int n) {

 int sum = 0;

 while(n) {

   sum += n % 10;

   n /= 10;

 }

 return sum;

}

void main() {

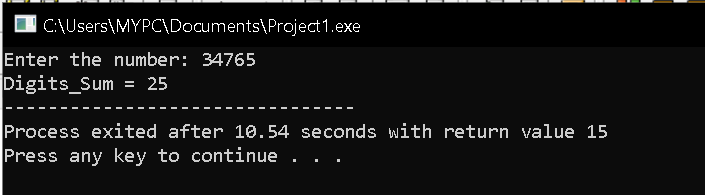
printf("Enter the number: ");

int num = 0;

scanf("%d",&num);

printf("Digits\_Sum = %d", digitSum(num));

}



//3

#include <stdio.h>

void Sieve(int start, int end) {

int prime[end];

int i = 0, j = 0;

for(i = 0; i < end; i++)

  prime[i] = 0;

for(i = 2; i <= end; i++){

  if(prime[i] == 0){

    for(j = i \* i; j <= end; j += i)

       prime[j] = 1;

    }

  }

 printf("The prime numbers within the range [%d,%d] is: ", start, end);

   for(i=2; i<=end; i++){

     if(prime[i]==0  && i>=start && i<=end)

       printf("%4d", i);

  }

}

void main() {

  printf("Enter 2 numbers: ");

  int start = 0, end = 0;

  scanf("%d%d", &start, &end);

  start -= (start < 0) \* 2 \* start;

  end -= (end < 0) \* 2 \* end;

  if(start > end){

     start ^= end;

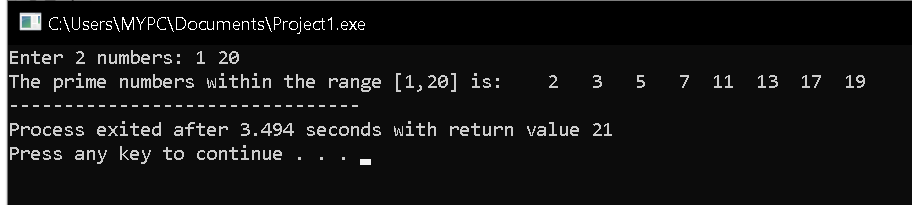
     end ^= start;

     start ^= end;

  }

  Sieve(start, end);

}



//4.Program to find factorial from 0 to 10.

#include <stdio.h>

int factorial(int n) {

  if (n == 0)  //edge case

     return 1;

  int f = 1;

  while (n) //loop for factorial calculation

    f \*= n--;

    return f;

}

int main() {

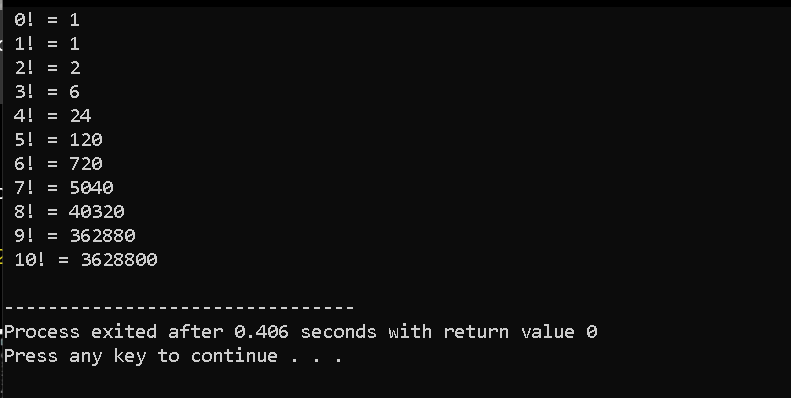
int i = 0;

for(i = 0; i < 11; ++i)

  printf(" %d! = %d\n", i, factorial(i));

return 0;

}



//5.Calculate factorial using reursion.

#include <stdio.h>

int factorial(int n) {

   if (n == 1 || n == 0) //base case

      return 1;

    return (n \* factorial(n-1)); //recursive step

}

void main() {

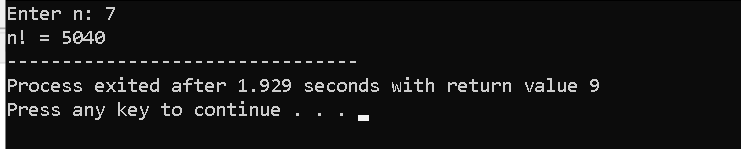
printf("Enter n: ");

int n;

scanf("%d", &n);

printf("n! = %d", factorial(n)); //fxn calling

}



//6. GCD via Recursion

#include <stdio.h>

int GCD(int num1, int num2) {

 if (num2)

   return GCD(num2, (num1 % num2)); //recursive step

return num1;

}

void main() {

printf("Enter n1: ");

int num1 = 0;

scanf("%d", &num1);

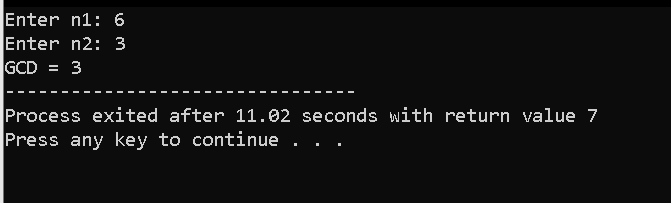
printf("Enter n2: ");

int num2 = 0;

scanf("%d", &num2);

printf("GCD = %d", GCD(num1, num2));

}



//7.Example to show pass by value and pass by reference

#include <stdio.h>

//Pass by reference: the actual values of the variables gets swapped.

void swapRef(int \*a, int \*b) {

  \*a ^= \*b;

  \*b ^= \*a;

  \*a ^= \*b; }

//Pass by value: the actual value of variables remains unchanged.

void swapVal(int a, int b) {

 printf("Before swap :  a = %d , b = %d\n", a, b);

 a ^= b;

 b ^= a;

 a ^= b;

 printf("After swap:  a = %d , b = %d\n", a, b); }

void main() {

printf("Enter num1 = "); //input for number\_1

int num1 = 0;

scanf("%d", &num1);

printf("Enter num2 = "); //input for number\_2

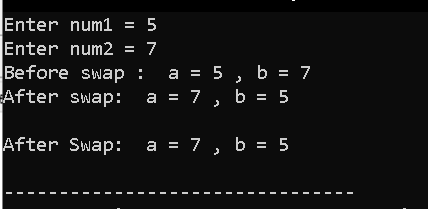
int num2 = 0;

scanf("%d", &num2);

swapVal(num1, num2); //pass by value

swapRef(&num1, &num2); //pass by reference

printf("\nAfter Swap:  a = %d , b = %d\n", num1, num2); }



//8. program to differentiate between static, auto and

 global.

#include <stdio.h>

//global variable declared globally

int g = 56784567;

void autoStatic() {

 auto int a = 0;

 static s = 0;

 printf("auto a = %d , static s = %d", a++, s++);

}

void main() {

int i = 0;

for (; i < 5; ++i)

  autoStatic();

printf("global g = %d\n", g);

}

/\*

An auto variable is called each time when the function is called and destroyed when the program's execution leaves the function.

Static is declared once and destroys once when the program's

execution finishes.

\*/

