**C programming**

[1] Write a C Program to demonstrate the following C programming concepts

(i) C Identifier

(ii) C Variables

(iii) C Comments (Single Line Comments, Multi-Line Comments)

(iv) C Constants (Decimal Constant, Floating-point Constant)

(v) C Literals (Integer literal, Float literal, Character literal, String literal)

#include<stdio.h>

#include<conio.h>

int main(void)

{

//This is are the datatypes

int a;

printf("Enter an integer\n");

scanf("%d",&a);

float b;

printf("Enter a float\n");

scanf("%f",&b);

char w;

printf("Enter a char\n");

scanf("%c",&w);

char d[50];

printf("Enter a string\n");

scanf("%s",&d);

/\*we stored data given by the user in a,b,c,d\*/

printf("%d is an integer\n",a);

printf("%d is a float\n",b);

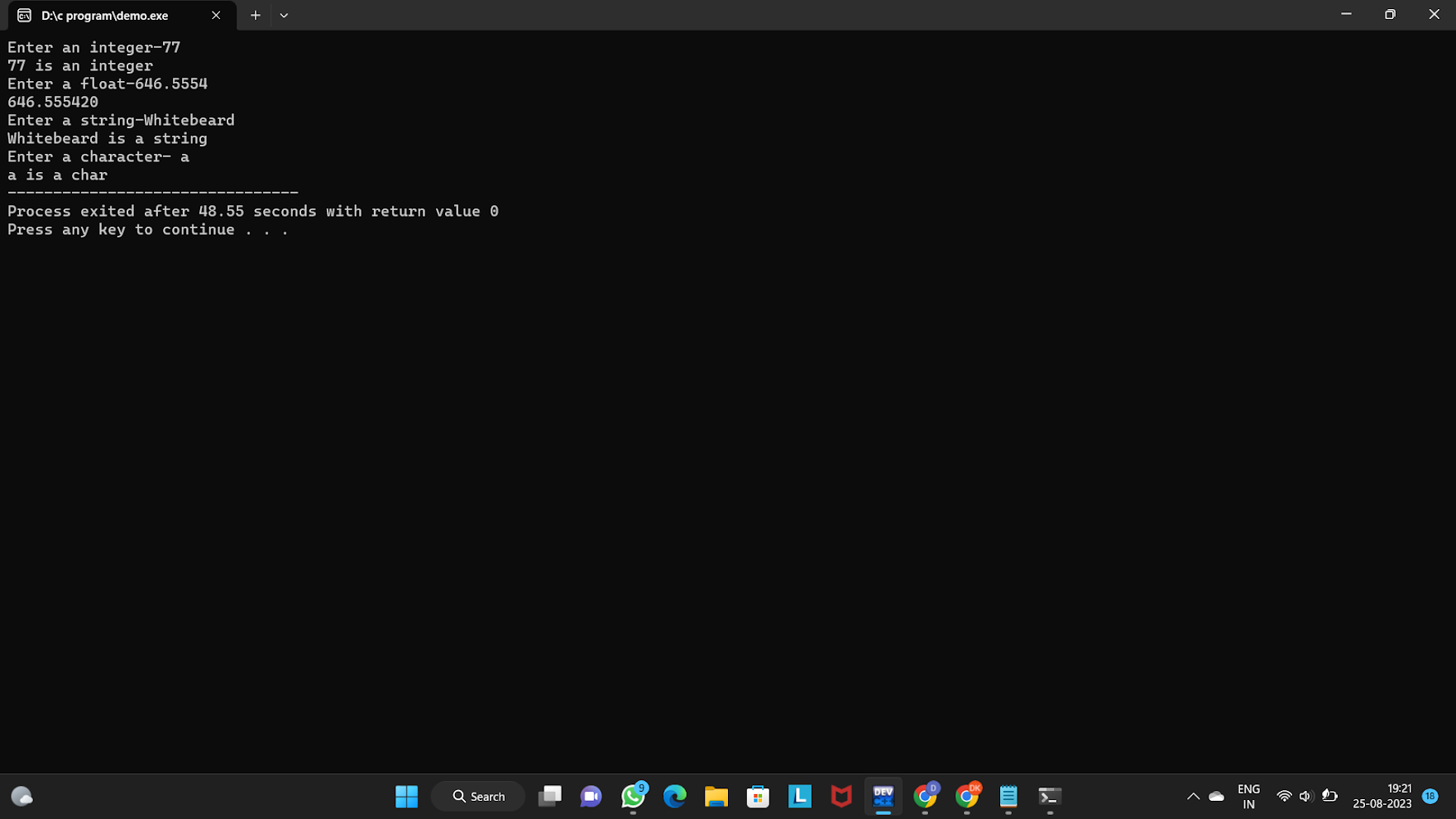
printf("%d is a char\n",w);

printf("%d is a string\n",d);

return 0;

}





2) Write a c program to Demonstrate / Implement conditional statements with the following parameters.

(a) Display the Statement to enter the ANNUAL INCOME of the person and calculate the INCOME TAX Payable based on the following details:

-Use appropriate Variables, Constants, Data types to store the data.

- Use appropriate Control structure (if, If\_Else, If\_else ladder based on the given condition)

If Salary is : ₹0 - ₹2,50,000 Then Income Tax is : Rs 0 (ZERO)

If Salary is : ₹2,50,000 - ₹5,00,000 Then Income Tax is : 5% of the salary

If Salary is : ₹5,00,000 - ₹7,50,000 Then Income Tax is : 10% of the salary

If Salary is : ₹7,50,000 - ₹10,00,000 Then Income Tax is : 15% of the salary

If Salary is : ₹10,00,000 - ₹12,50,000 Then Income Tax is : 20% of the salary

If Salary is : ₹12,50,000 - ₹15,00,000 Then Income Tax is : 25% of the salary

More than ₹15,00,000 Then Income Tax is : 30% of the salary

- Use the appropriate Operators to calculate.

- Display the formatted output upon calculation of the INCOME TAX Deduction (Total annual income and Tax deduction).

#include <stdio.h>

int main() {

int a;

char tax;

printf("Enter your annual income: ");

scanf("%d", &a);

if (a<=250000) {

tax= 0;

} else if (a<=500000) {

tax = a\*0.05;

} else if (a <= 750000) {

tax = a \* 0.10;

} else if (a <= 1000000) {

tax = a \* 0.15;

} else if (a <= 1250000) {

tax = a \* 0.20;

} else if (a <= 1500000) {

tax = a \* 0.25;

} else {

tax = a \* 0.30;

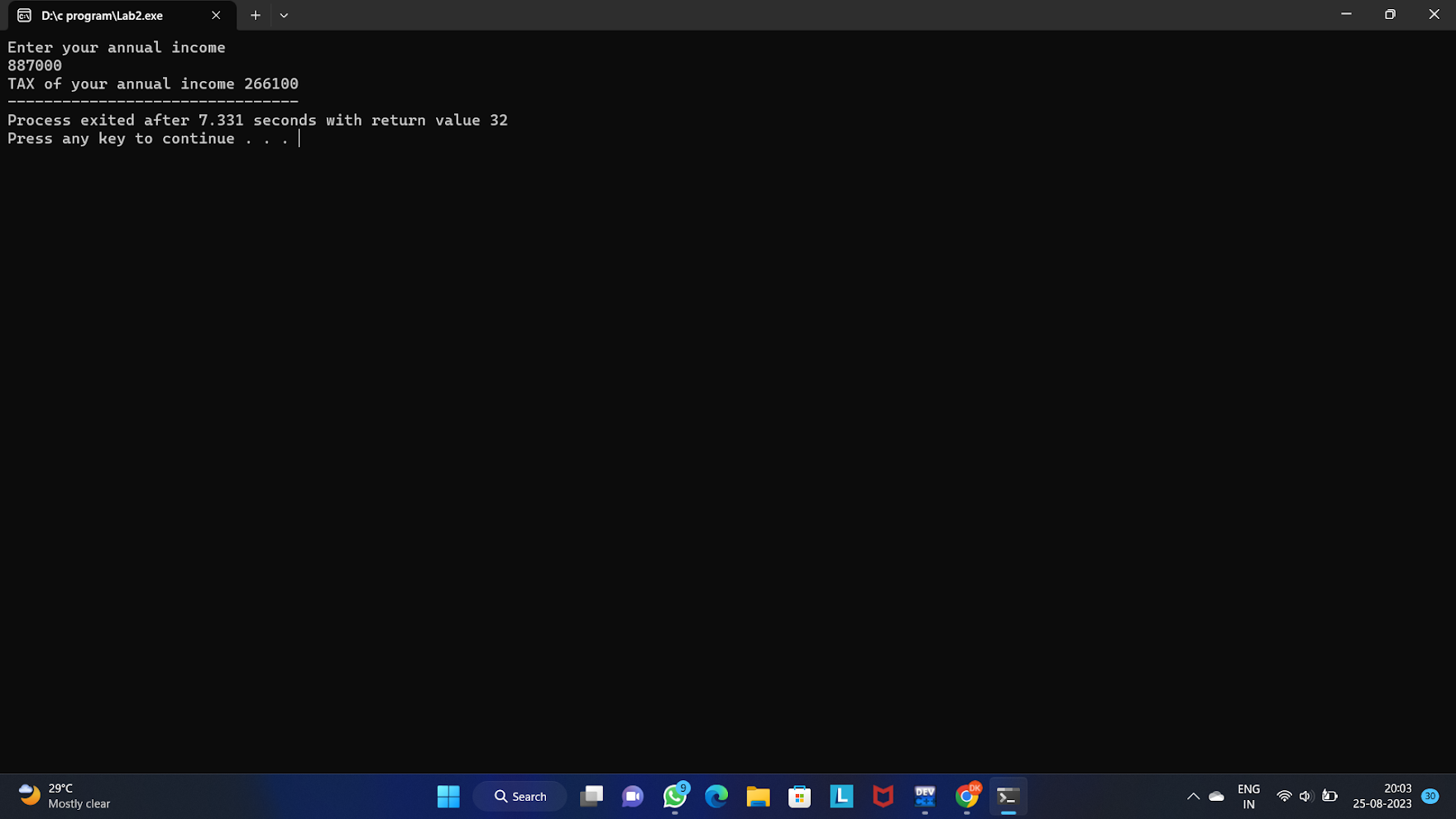
}

printf("Your annual income is Rs. %d\n", a);

printf("Your income tax payable is Rs. %c\n", tax);

return 0;

}



3)Write a C Program to calculate the "attendance shortage fine" until the user enters zero based on the following parameters using any one looping and control structure.

- Display the statement to get the attendance percentage from 0 to 100.

- If the attendance percentage is less than ZERO (0) or grater than HUNDRED (100), it should show an error message.

- Upon giving the Attendance calculate the fine as given below:

(a) If attendance is 90-100% - Fine ZERO (0)

(b) If attendance is 80-89% - Fine Rs.1000

(c) If attendance is 70-79% - Fine Rs.2000

(d) If attendance is 60-69% - Fine Rs. 3000

(e) If attendance is less than 60% - Fine Rs.4000

-Repeat the calculation for various STUDENTS until we give zero for input. Upon giving input ZERO (0) program should end and come out.

 #include<stdio.h>

int main()

{

int a,b;

printf("Enter your Reg No:");

scanf("%d",&a);

if(0>a)

{

printf("Entered Reg No is INVALID!\n",a);

}else

{

printf(" Reg No %d ",a);

}

printf("\nEnter your Attendence percentage:");

scanf("%d",&b);

if(0>b&&100<b)

{

printf("The entered percentage in INVALID!",b);

}

else if(b<60)

{

printf("YOUR ATTENDENCE FINE IS 4000",b);

}

else if(b>=60&&b<=69)

{

printf("YOUR ATTENDENCE FINE IS 3000",b);

}

else if(b>=70&&b<=79)

{

printf("YOUR ATTENDENCE FINE IS 2000",b);

}

else if(b>=80&&b<=89)

{

printf("YOUR ATTENDENCE FINE IS 1000",b);

}

else if(b>=90&&b<=100)

{

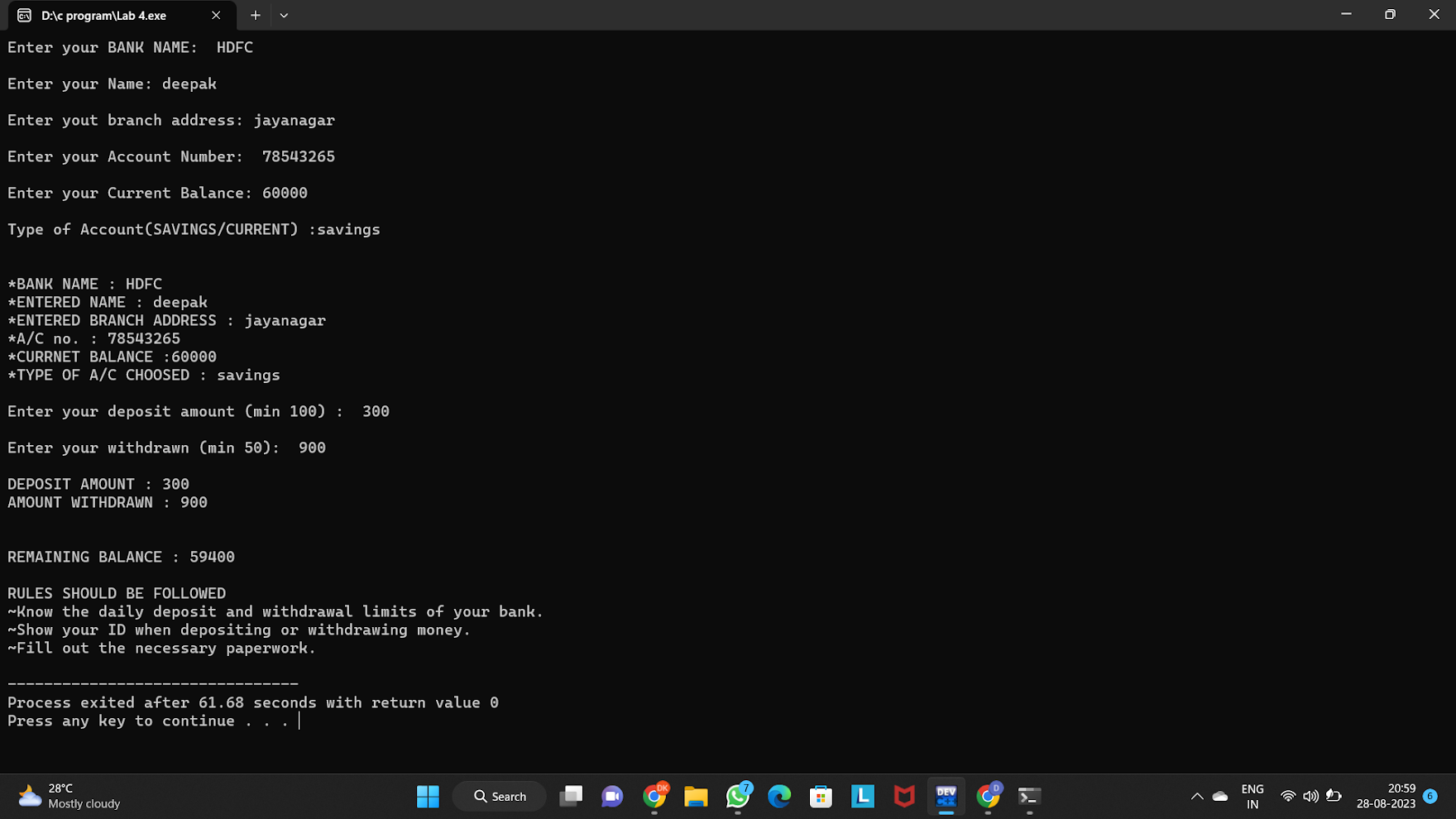
printf("YOUR ATTENDENCE FINE IS 0",b);

}

return 0;

}





4)Write a C Program to implement ONE DIMENSIONAL ARRAY based on the following academic domain parameters.

- Collect the Student REG NO, total Marks out of 600 from the user (minimum 10 individual entry) and store them in respective one dimensional different arrays.

- From the Student total marks calculate the individual average of the same and store it in the array variable AVG[].

- Display the entire ARRAY contents including REG NO, TOTAL MARKS, AVG.

- Use appropriate Variables, Data Types and Operators for manipulation / calculation.

- The output must be properly formatted.

#include<stdio.h>

int main()

{

char a[50],b[50],e[50],i[50];

int c,d;

printf("Enter your BANK NAME: ");

scanf("%s",&a);

printf("\nEnter your Name: ");

scanf("%s",&b);

printf("\nEnter yout branch address: ");

scanf("%s",i);

printf("\nEnter your Account Number: ");

scanf("%d",&c);

printf("\nEnter your Current Balance: ");

scanf("%d",&d);

if(0> d)

{

printf("Entered value in invalid\n");

}

printf("\nType of Account(SAVINGS/CURRENT) :");

scanf("%s",&e);

printf("\n\n\*BANK NAME : %s\n",a);

printf("\*ENTERED NAME : %s\n",b);

printf("\*ENTERED BRANCH ADDRESS : %s\n",i);

printf("\*A/C no. : %d\n",c);

printf("\*CURRNET BALANCE :%d\n",d);

printf("\*TYPE OF A/C CHOOSED : %s\n",e);

char oper;

do

{

printf("\n\n BANKING OPERATIONS\n 1] Deposit\n 2] Withdrawal\n 3] Check Balance\n 4] Guidelines\n 5] Exit\n\n");

printf("Operation :");

int operation;

scanf("%d", &operation);

switch (operation)

{

case 1:

printf("---------Deposit---------\nEnter Amount: ");

int add\_amount;

scanf("%i", &add\_amount);

d += add\_amount;

printf("---------Money Credited---------\n");

break;

case 2:

printf("---------Withdraw---------\nEnter Amount: ");

int debit\_amount;

scanf("%i", &debit\_amount);

d -= debit\_amount;

printf("---------Money Debited---------\n");

break;

case 3:

printf("---------Balance---------\n");

printf("Balance: %i\n", d);

printf("---------Balance---------\n");

break;

case 4:

printf("---------Guidelines---------\n");

printf(" Minimum balance should be 4000");

break;

case 5:

return 0;

default:

break;

}

printf("\nDo You Want to continue Y/N :");

scanf(" %c", &oper);

}

while(oper == 'Y');

}

5)Write a C Program to implement ONE DIMENSIONAL ARRAY based on the following academic domain parameters.

- Collect the Student REG NO, total Marks out of 600 from the user (minimum 10 individual entry) and store them in respective one dimensional different arrays.

- From the Student total marks calculate the individual average of the same and store it in the array variable AVG[].

- Display the entire ARRAY contents including REG NO, TOTAL MARKS, AVG.

- Use appropriate Variables, Data Types and Operators for manipulation / calculation.

- The output must be properly formatted.

#include <stdio.h>

#include<conio.h>

int main(void)

{

int rn[10];

float tm[10],com[10],mat[10],eco[10],eng[10],acc[10],cs[10];

double avg[10];

int i;

for(i=0;i<3;i++)

{

printf("\n\nEnter student register number...........: ");

scanf("%d",&rn[i]);

printf("\nEnter Total marks out of 600............: ");

scanf("%f",&tm[i]);

if(tm[i]>600 || tm[i]<0)

{

do

{

printf("\nEnter Valid marks");

printf("\nEnter Total marks out of 600............: ");

scanf("%f",&tm[i]);

}while(tm[i]>600 || tm[i]<0);

}

printf("Enter ACCOUNTANCY marks out of 100......: ");

scanf("%f",&acc[i]);

if(acc[i]>100 || acc[i]<0)

{

do

{

printf("\nEnter Valid marks");

printf("\nEnter ACCOUNTANCY marks out of 100......: ");

scanf("%f",&acc[i]);

}while(acc[i]>100 || acc[i]<0);

}

printf("Enter ECONOMICS marks out of 100........: ");

scanf("%f",&eco[i]);

if(eco[i]>100 || eco[i]<0)

{

do

{

printf("\nEnter Valid marks");

printf("\nEnter ECONOMICS marks out of 100........: ");

scanf("%f",&eco[i]);

}while(eco[i]>100 || eco[i]<0);

}

printf("Enter COMMERCE marks out of 100.........: ");

scanf("%f",&com[i]);

if(com[i]>100 || com[i]<0)

{

do

{

printf("\nEnter Valid marks");

printf("\nEnter COMMERCE marks out of 100.........: ");

scanf("%f",&com[i]);

}while(com[i]>100 || com[i]<0);

}

printf("Enter MATHEMATICS marks out of 100......: ");

scanf("%f",&mat[i]);

if(mat[i]>100 || mat[i]<0)

{

do

{

printf("\nEnter Valid marks");

printf("\nEnter MATHEMATICS marks out of 100......: ");

scanf("%f",&mat[i]);

}while(mat[i]>100 || mat[i]<0);

}

printf("Enter COMPUTER SCIENCE marks out of 100.: ");

scanf("%f",&cs[i]);

if(cs[i]>100 || cs[i]<0)

{

do

{

printf("\nEnter Valid marks");

printf("\nEnter COMPUTER SCIENCE marks out of 100.: ");

scanf("%f",&cs[i]);

}while(cs[i]>100 || cs[i]<0);

}

printf("Enter ENGLISH marks out of 100..........: ");

scanf("%f",&eng[i]);

if(eng[i]>100 || eng[i]<0)

{

do

{

printf("\nEnter Valid marks");

printf("\nEnter ENGLISH marks out of 100..........: ");

scanf("%f",&eng[i]);

}while(eng[i]>100 || eng[i]<0);

}

}

printf("\n\n");

printf("The Student Details are:\n");

printf("-------------------------------------- -------------------------\n");

printf("REG NO TOTAL\_MARKS AVERAGE\n");

printf("--------------------------------------------------------------\n");

for(i=0;i<3;i++)

{

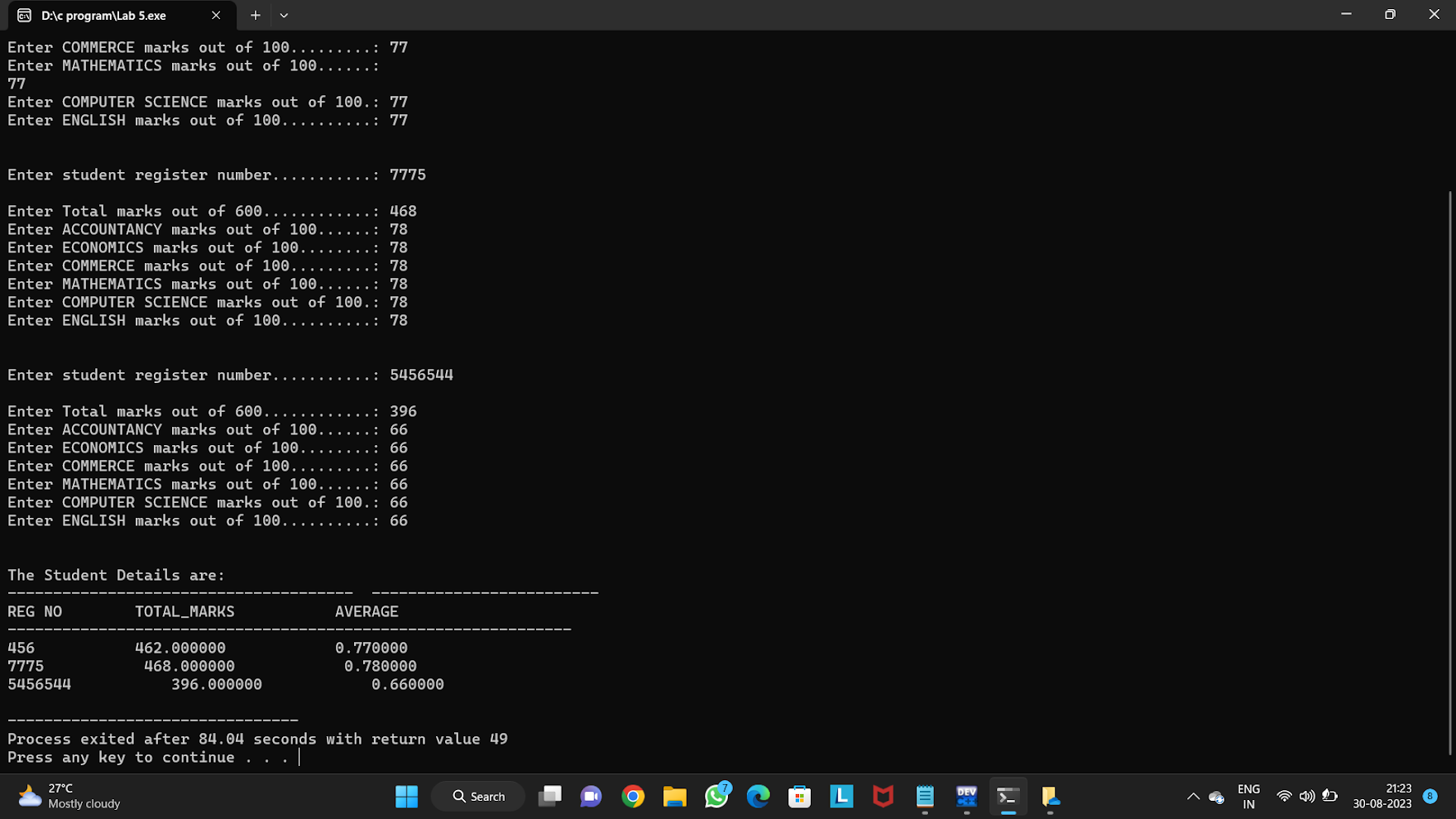
avg[i]=((com[i]+mat[i]+eco[i]+eng[i]+acc[i]+cs[i])/600);

printf("%d\t \t %f \t \t %lf\n", rn[i], tm[i], avg[i]);

}

}





6)Write a C Program to implement storage of temperature of two or more cities of a week / month and display the result using multidimensional arrays

- Capture / collect the input values and store it in VAL\_1 [city] [temperature]. Use appropriate variables and data types further to complete the implementation.

- Use loops to capture and display the MULTI Dimensional array values.

(a) Accessing Elements of Multidimensional Arrays

(b) Initializing Multidimensional Arrays

(c) Iterating Over Multidimensional Arrays

#include <stdio.h>

#include <string.h>

int main(void)

{ int Cities = 2;

int no\_days = 4;

char days[10][100] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};

char cities[10][100] = {"Bangalore", "Chennai"};

char statment[Cities][no\_days][10];

for (int i = 0; i < no\_days; i++)

{

for (int j = 0; j< Cities;j++)

{

char buffer[100];

printf("What the weather at %s on %s : ",cities[j], days[i]);

fgets(statment[j][i], 100, stdin);

}

}

printf("\n\n|Days\t |City\t |Weather |\n");

for (int i = 0; i < no\_days; i++)

{

for (int j = 0; j < Cities ; j++)

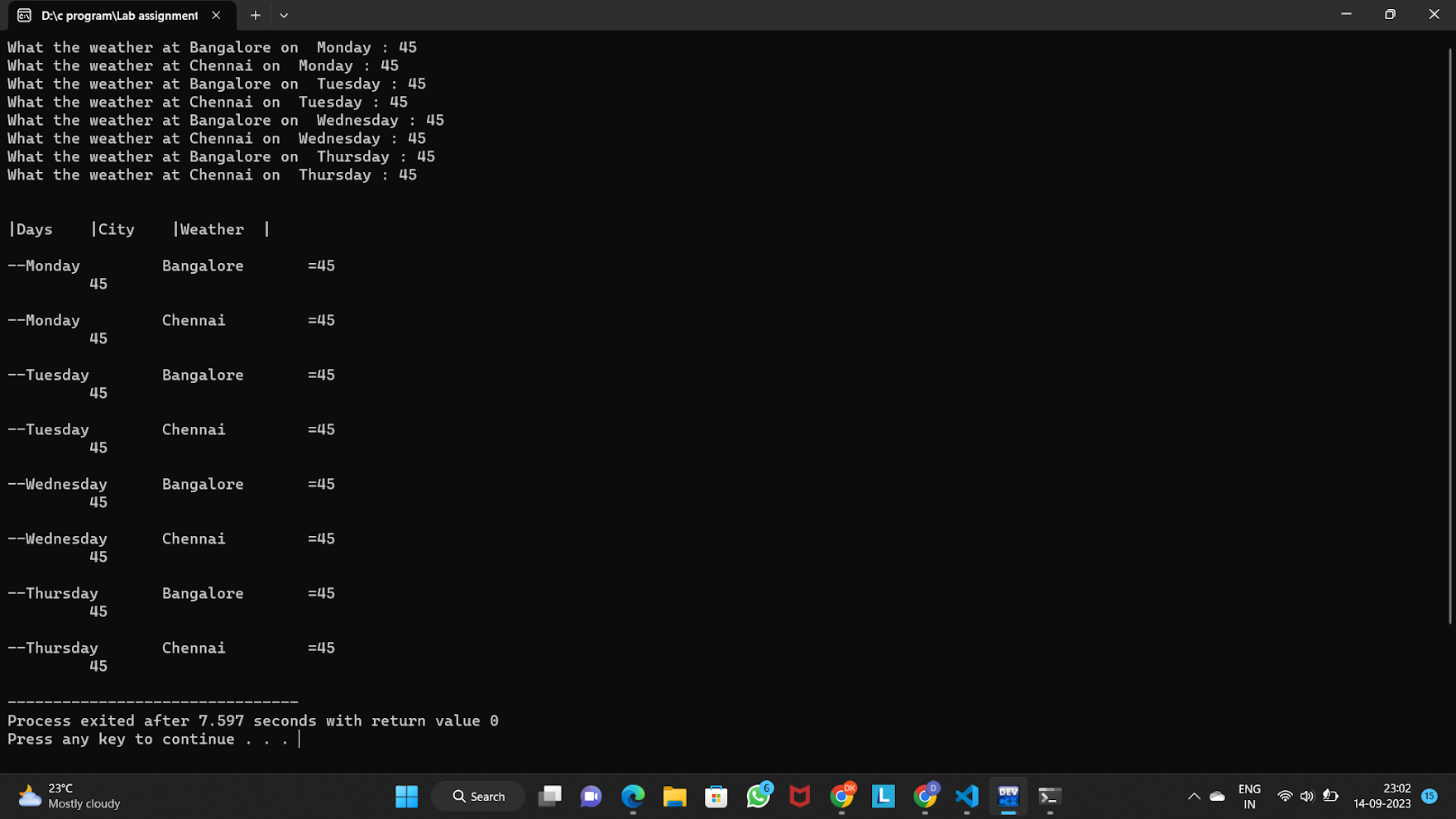
{

printf("\n--%s\t %s\t =%s\t 45\n",days[i], cities[j],statment[j][i]);

}

}

}



7)Demonstrate STRING operations in a C Program by considering the following parameters (STRLEN,STRCMP)

- Prepare a "C Program" to implement a Course enrollment form - get the candidate name (First Name, Last name), age, address and course opted and store them in respective arrays.

- Check if the candidate name is having more than 12 characters - provide an error message and ask the user to re\_enter the NAME and LAST NAME. (use STRLEN)

- If the Student opted for any academic programme provided as input (BSc, BCA, BE, etc..) then display the courses / subjects offered in the respective programme (use STRCMP)

#include <stdio.h>

#include <string.h>

int main() {

char firstName[50], lastName[50], age[10], address[100], course[50];

printf("Enter First Name: ");

scanf("%s", firstName);

printf("Enter Last Name: ");

scanf("%s", lastName);

if (strlen(firstName) + strlen(lastName) > 12)

{

do

{

printf("Error: Candidate name has more than 12 characters. Please re-enter.\n");

printf("Enter First Name: ");

scanf("%s", firstName);

printf("Enter Last Name: ");

scanf("%s", lastName);

}while(strlen(firstName) + strlen(lastName) > 12);

}

printf("Enter Age: ");

scanf("%s", age);

printf("Enter Address: ");

scanf(" %s", address);

printf("Enter Course Opted(Strictly in lower case): ");

scanf(" %s", course);

printf("\nCandidate Information:\n");

printf("Name: %s %s\n", firstName, lastName);

printf("Age: %s\n", age);

printf("Address: %s\n", address);

printf("Course Opted: %s\n", course);

if (strcmp(course, "bsc") == 0) {

printf("\nSubjects offered for BSc:\n");

printf("1. Mathematics\n 2. Eletronics\n 3. English\n 4.Biotechnology\n 5.Computer Science\n 6.Physics");

} else if (strcmp(course, "bca") == 0) {

printf("\nSubjects offered for BCA:\n");

printf("1. Programming languages \n2. Web application Development\n3. Data analysis");

} else if (strcmp(course, "be") == 0) {

printf("\nSubjects offered for BE:\n");

printf("1. Electrical Engineering\n2. Mechanical Engineering\n3. Computer Science\n");

} else {

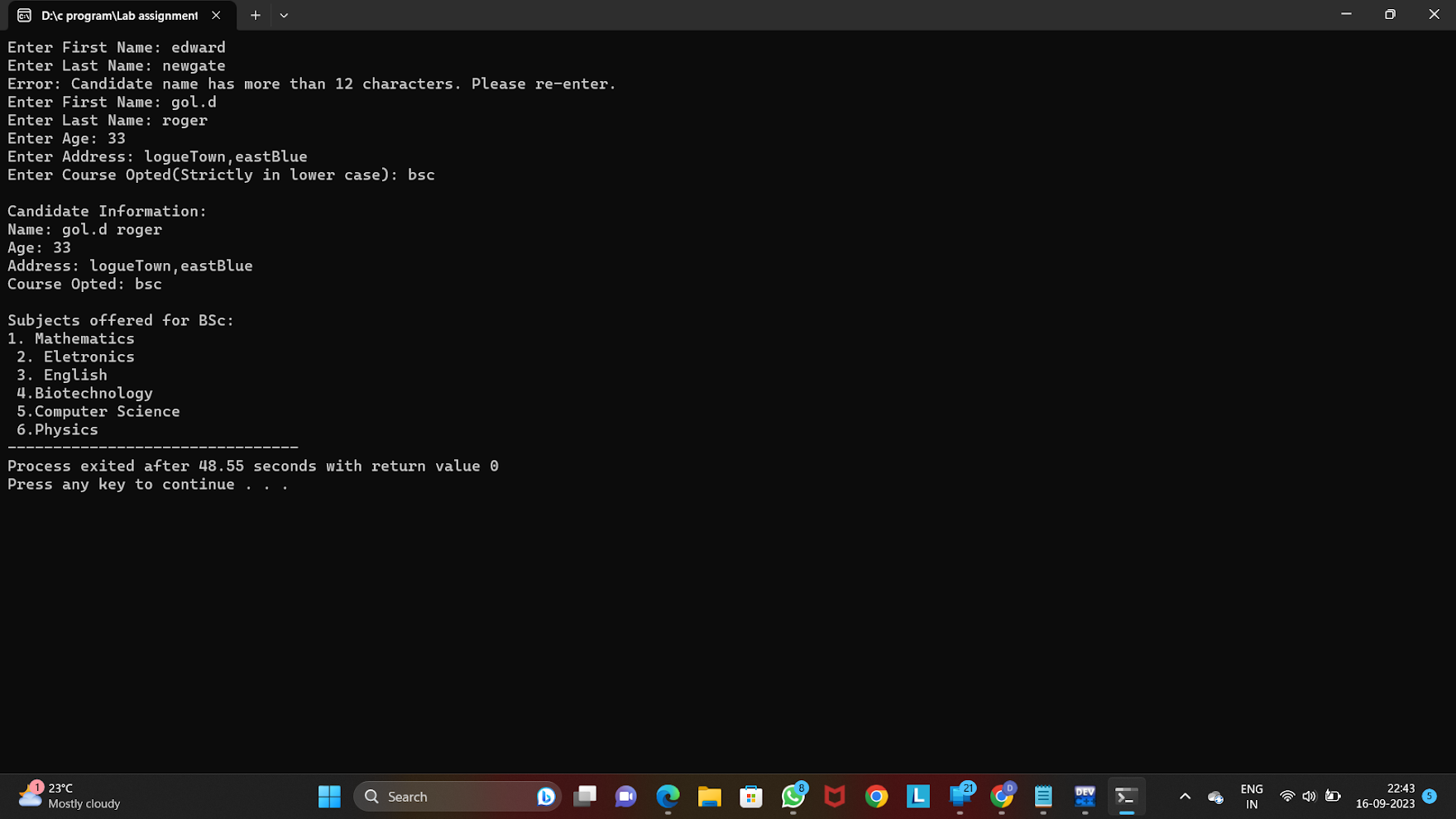
printf("\nSubjects for this course are not available.\n");

}

return 0;

}





8)C Program to Display Prime Numbers Between Intervals Using Function.

Include the following C Concepts :

C for Loop

C break and continue

C Functions

C User-defined functions

Requirements:

(1) Program should ask input values from the user and store them in two variables n1 and n2;

(2) Display prime numbers between n1 and n2; Use a FLAG to check the number is PRIME or NOT.

(3) Develop a User defined function to check the given number is PRIME number or NOT;

#include <stdio.h>

#include <string.h>

//User defined function

int primenumbers(int n)

{

int a;

for(a = 2; a<= n/2; ++a)

{

if( n % a == 0)

return 0;

}

return 1;

}

//Main function

int main()

{

int n1, n2, i,flag ;

printf("\nEnter two numbers (n1 and n2)");

printf("\nEnter n1: ");

scanf("%d",&n1);

printf("Enter n2: ");

scanf("%d",&n2);

printf("Prime numbers between %d and %d are: ",n1 , n2);

for(i = n1; i <= n2; i++)

{

if(i == 1)

continue;

// to skip everthing below that in the loop and goes to the next value of i

flag =primenumbers(i);

if(flag == 1)

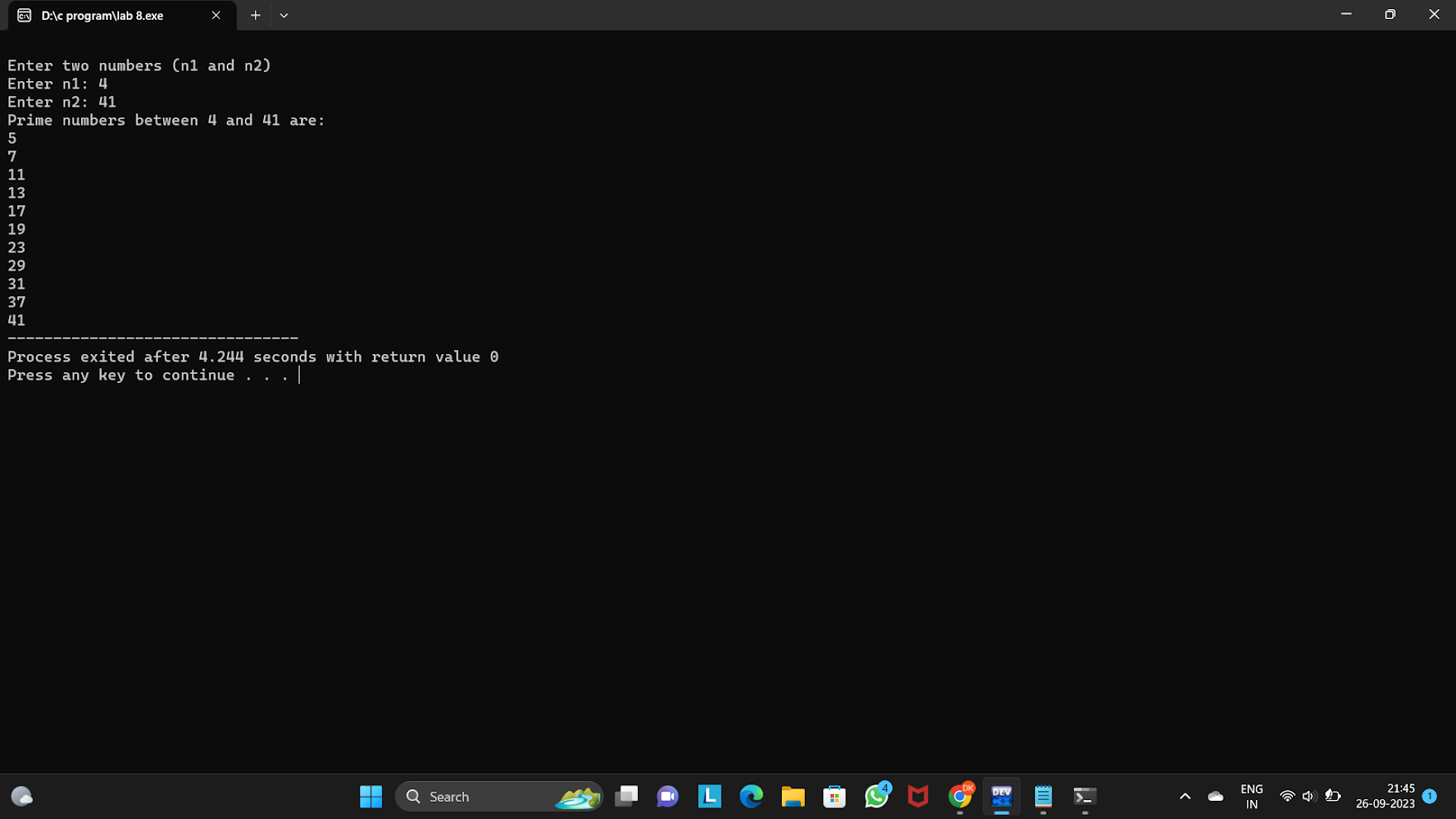
printf("\n%d",i);

}

return 0;

}





9)

Implement a C program on FUNCTIONS with the following specification:

Write a C Program for SUPER\_MARKET STORE Management where it consist of minimum four (04) functions in it with the following parameters:

(1) Function\_01 : Show the MARKET STOCK items of the groceries and other items of the SuperMarket.

(2) Function\_02 : In this function, client should provide the input vales for ORDERING Market\_Store Components / Items and store the ORDERING components appropriately using variables / arrays.

(3) Function\_03 : In this function the calculation of the user ORDER details should execute such as calculating the total cost of the client ORDERS

(4) Function\_04 : Display the result of the Client order details (BILLING).

#include<stdio.h>

#include<string.h>

int function\_1()

{

printf(" --------------------------ITEMS----------------------\n\n");

printf("\t 1]Rice Rs.100 \n ");

printf("\t 2]Wheat Rs.70 \n");

printf("\t 3]MILK Rs.45\n");

printf("\t 4]BREAD Rs.30\n");

printf("\t 5]EGGS Rs.5\n");

}

float function\_2(int quantites[]);

int function\_3(int quantites[]);

void function\_4(float bill);

int main()

{

int quantites[5];

printf("\t\t\t\t-------WELCOME TO STARK SUPER MARKET---------\n\n\n\n");

float bill;

//calling fuction\_1

function\_1();

//calling fuction\_2

function\_2(quantites);

//calling fuction\_3

bill = function\_3(quantites);

//calling fuction\_4

function\_4(bill);

return 0;

}

//declration of functions

//for function\_2

float function\_2(int quantites[])

{

printf("\n-------------------------------------------------------------------------------------\n");

printf("\n\t---------------Enter the quantity of items: -----------------------------\n");

printf("\n-------------------------------------------------------------------------------------\n");

printf("Rice (1 kg) : ");

scanf("%d", &quantites[0]);

printf("Wheat (1 kg): ");

scanf("%d", &quantites[1]);

printf("Milk (1 litre): ");

scanf("%d", &quantites[2]);

printf("Bread (1 loaf): ");

scanf("%d", &quantites[3]);

printf("Eggs (per piece): ");

scanf("%d", &quantites[4]);

printf("\n-------------------------------------------------------------------------------------\n");

}

//for function\_3

int function\_3(int quantites[5])

{

int i;

float cost[] = {100, 70, 45, 30, 5};

float bill = 0;

//using for loop

for(i = 0; i<5; i++)

{

bill += quantites[i] \* cost[i];

}

return bill;

bill = bill + (bill \* 0.12);

}

//for function\_4

void function\_4(float bill)

{

printf("\n\t------------------------\t\n");

printf(" BILL \n");

printf("\t------------------------\n");

printf(" with GST 12%-------- \n");

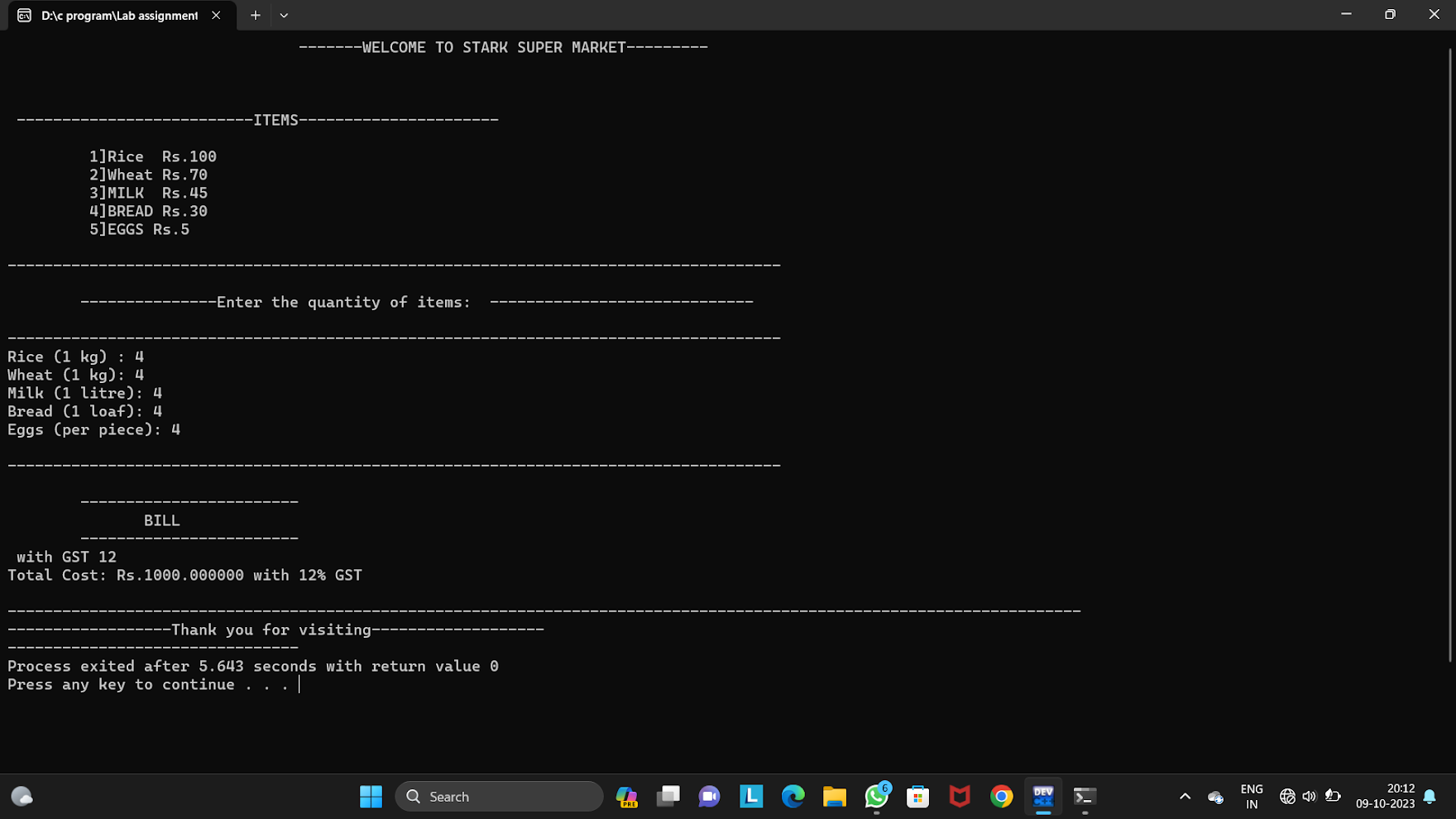
printf("Total Cost: Rs.%f with 12%% GST\n", bill);

printf("\n----------------------------------------------------------------------------------------------------------------------\n");

printf("------------------Thank you for visiting-------------------");

}





10)

The call by value method of passing arguments to a function copies the actual value of an argument into the formal parameter of the function. The call by reference method of passing arguments to a function copies the address of an argument into the formal parameter.

----------------------------------

With respect to the above definitions Write a C Program to demonstrate the CALL BY VALUE and CALL BY REFERENCE with respect to some domain (TOPIC : Swapping, Ascending order, Descending order, etc..)

(a) The program must get the input data from the user and store them in appropriate variables. Use appropriate validations for input value.

(b) Prepare required User Defined functions - which consist of both categories CALL BY VALUE and CALL BY REFERENCE and display the result.

#include<stdio.h>

void swap(int, int);

main(void)

{

int a = 100;

int b = 500;

printf("Swaping of two numbers a = %d b = %d\n",a,b);

swap(a,b);

}

void swap(int a, int b)

{

int c;

c = a;

a = b;

b = c;

printf("After swapping values a = %d, b = %d\n",a,b);

}



#include <stdio.h>

void arrange(int \*a, int \*b, int \*c) {

if (\*a > \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

if (\*a > \*c) {

int temp = \*a;

\*a = \*c;

\*c = temp;

}

if (\*b > \*c) {

int temp = \*b;

\*b = \*c;

\*c = temp;

}

}

int main() {

int a,b,c;

printf("Enter three numbers: ");

scanf("%d %d %d", &a, &b, &c);

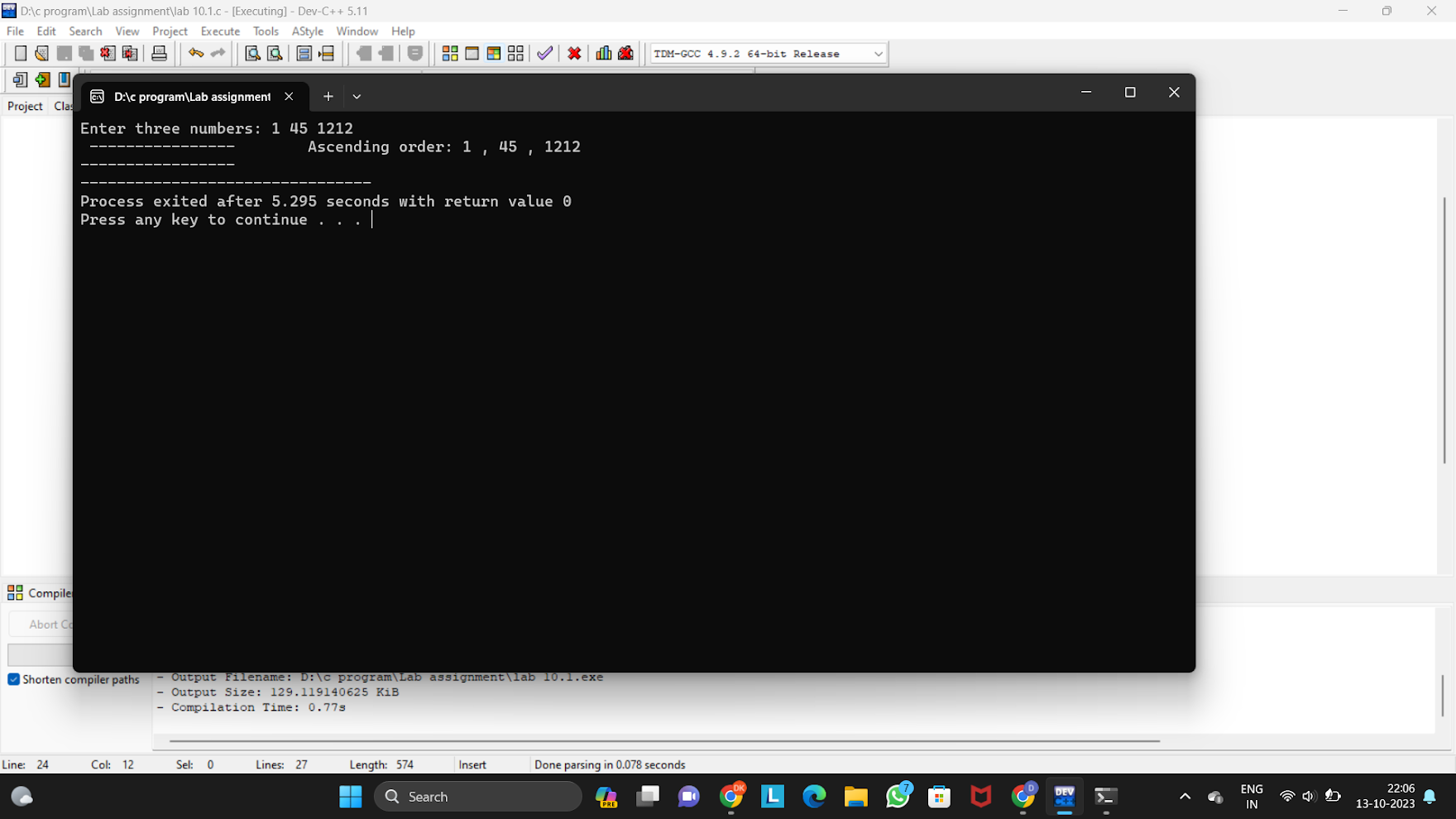
arrange(&a, &b, &c);

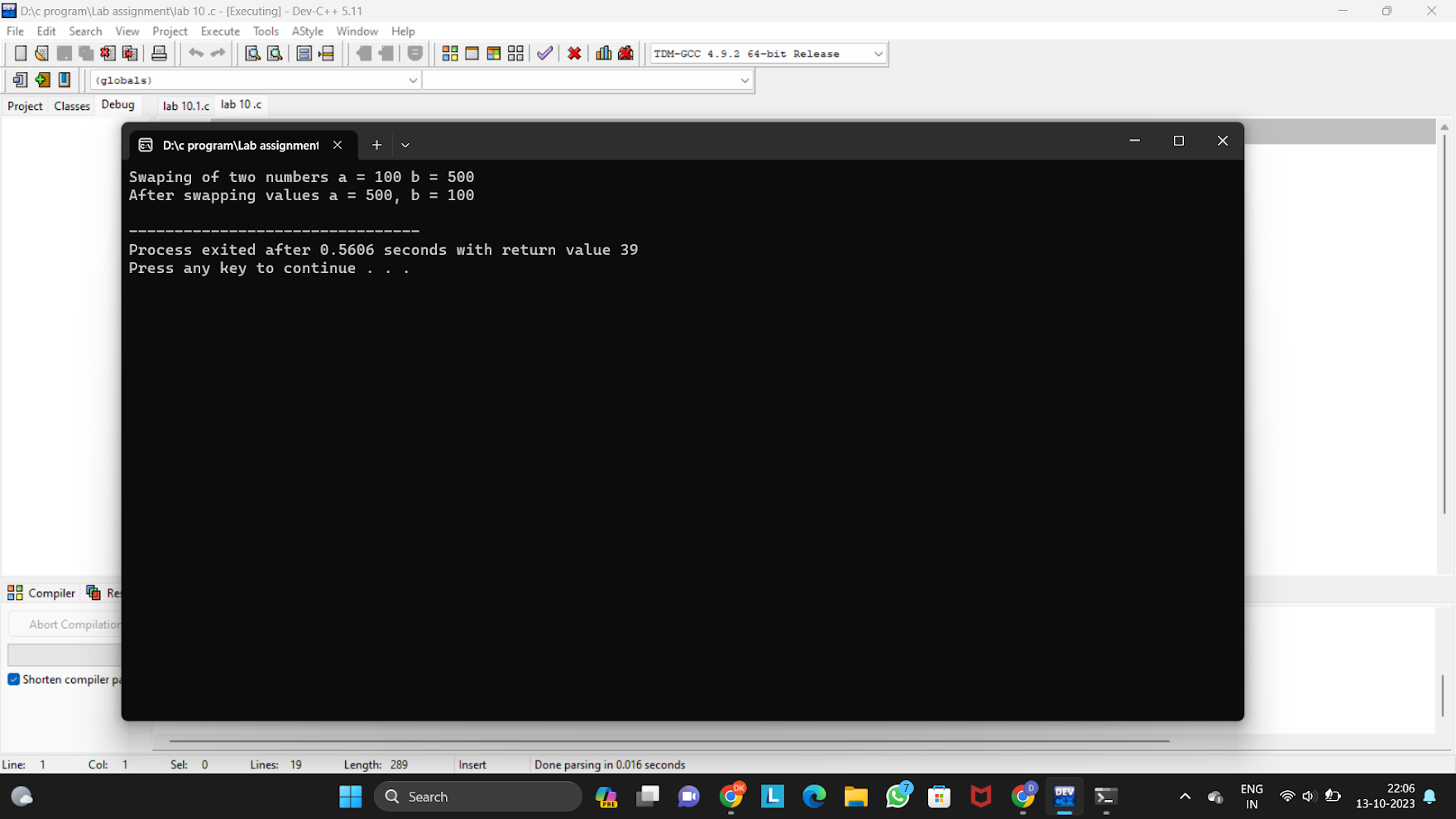
printf(" ----------------\t Ascending order: %d , %d , %d\n-----------------", a, b, c);

return 0;

}







11)

Write a C program to demonstrate "POINTERS" which can be defined as a derived data type that can store addresses of other C variables or a memory location. Program must consist of the following:

(i) Pointer Declaration

(ii) Pointer Initialization

usage of the pointers in the program

#include <stdio.h>

int main()

{

int n;

int\* p;

printf("Enter an Number: ");

while(scanf("%d", &n) != 1 )

{

printf("\nInvalid Entry");

printf("\nEnter an Number: ");

while(getchar() != '\n');

}

p = &n;

int a, s , m;

printf("Enter a value to add: ");

while(scanf("%d", &a) != 1 )

{

printf("Invalid Entry");

printf("\nEnter a value to add: ");

while(getchar() != '\n');

}

printf("Enter a value to subtract: ");

while(scanf("%d", &s) != 1 )

{

printf("Invalid Entry");

printf("\nEnter a value to subtract: ");

while(getchar() != '\n');

}

printf("Enter a value to check remainder: ");

while(scanf("%d", &m) != 1 )

{

printf("Invalid Entry");

printf("\nEnter a value to check remainder: ");

while(getchar() != '\n');

}

int sum = \*p + a;

int diff = \*p - s;

int mod = \*p % m;

printf("\nOriginal Value: %d\n", \*p);

printf("-----------Addition : %d + %d = %d\n--------------------", \*p, a, sum);

printf("-----------Subtraction : %d - %d = %d\n--------------------", \*p, s, diff);

printf("-----------Result after Modulas operation: %d %% %d = %d\n------", \*p, m, mod);

return 0;

}



