DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering) Shahbad Daulatpur,Bawana Road, Delhi 110042

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

 s

Lab Code: CO102

LAB FILE

Submitted to: Submitted by:

Dr. Ashish Girdhar Vipul Sharma

Assistant Professor 2K20/A11/23

Department of Computer B.Tech 1st year

Science and Engineering ME

INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No | Experiment Name | Date | Signature |
| 1. | Program to find sum and average of two numbers | 12/4/21 |  |
| 2. | Program to find greatest of 10 numbers | 12/4/21 |  |
| 3. | Program to find Simple Interest | 24/5/21 |  |
| 4. | Program to print the following pattern.(triangle of stars) | 24/5/21 |  |
| 5. | Program to fine whether the entered number is prime. | 24/5/21 |  |
| 6. | Program to find sum of a 5 digit number. | 31/5/21 |  |
| 7 | Program to reverse a 5 digit number | 7/6/21 |  |
| 8. | Program to convert decimal to binary and vice versa | 7/6/21 |  |
| 9. | Program to implement switch case statement | 7/6/21 |  |
| 10. | Program to generate the Fibonacci sequence. | 7/6/21 |  |
| 11. | Program to find exponential function. | 7/6/21 |  |
| 12. | Program to search a number from an array using linear search. | 7/6/21 |  |
| 13. | Program to search a number from an array using binary search. | 14/6/21 |  |
| 14. | Program to sort an array using Bubble sort | 14/6/21 |  |
| 15. | Program to sort an array using selection sort. | 14/6/21 |  |
| 16. | Program to sort an array using insertion sort. | 14/6/21 |  |
| 17. | Program to find factorial of a number using recursion | 14/6/21 |  |
| 18. | Program to find the length of the string without using strlen and then pass the string to characters | 14/6/21 |  |
| 19. | Program to count the number of vowels in a given string. | 14/6/21 |  |
| 20. | Program to check if a given string is a palindrome or not. | 14/6/21 |  |
| 21. | Program to string concatenation | 14/6/21 |  |
| 22. | Program to string comparison. | 14/6/21 |  |
| 23. | Program to string reverse | 14/6/21 |  |
| 24. | Program to convert a string from lower case to upper case and vice versa. | 14/6/21 |  |
| 25. | Program for the addition of two 3 x 3 matrices. | 21/6/21 |  |
| 26. | Program to multiply two 3 x 3 matrices | 21/6/21 |  |
| 27. | Program to swap two numbers using pointers | 5/7/21 |  |
| 28. | Program to generate the employee details using structure. | 5/7/21 |  |
| 29. | Program to find the area and perimeter of a circle, rectangle, square and triangle using functions. | 5/7/21 |  |
| 30. | Program to pass and return pointer to function hence calculate average of an array | 12/7/21 |  |
| 31. | Program to pass an array as pointer to a function that calculates the sum of all elements of the array | 12/7/21 |  |
| 32. | Program to demonstrate the example of array of pointers. | 12/7/21 |  |
| 33. | Program to create a file called emp.txt and store information about a person, in terms of his name, age and salary. |  |  |
| 34. | Program which copies one file contents to another file |  |  |
| 35. | Program to read a file and after converting all lower case to upper case letters write it to another file |  |  |
| 36. | Program to find the size of a given file. |  |  |
|  |  |  |  |
|  |  |  |  |

**EXPERIMENT 1.**

**OBJECTIVE-** Program to find sum and average of two numbers.

**INTRODUCTION-** In This Program We Input Two Numbers let’s say a and b and take sum of two and with help of sum then calculate average of two.

**PROGRAM CODE-**

// average of two numbers

#include<stdio.h>

int main()

{

int num1,num2;

int sum;

float avg;

printf("enter number 1\n"); //input

printf("enter number 2\n"); //input

scanf("%d",&num1);

scanf("%d",&num2);

sum=num1+num2;

printf("value of sum is %d",sum);

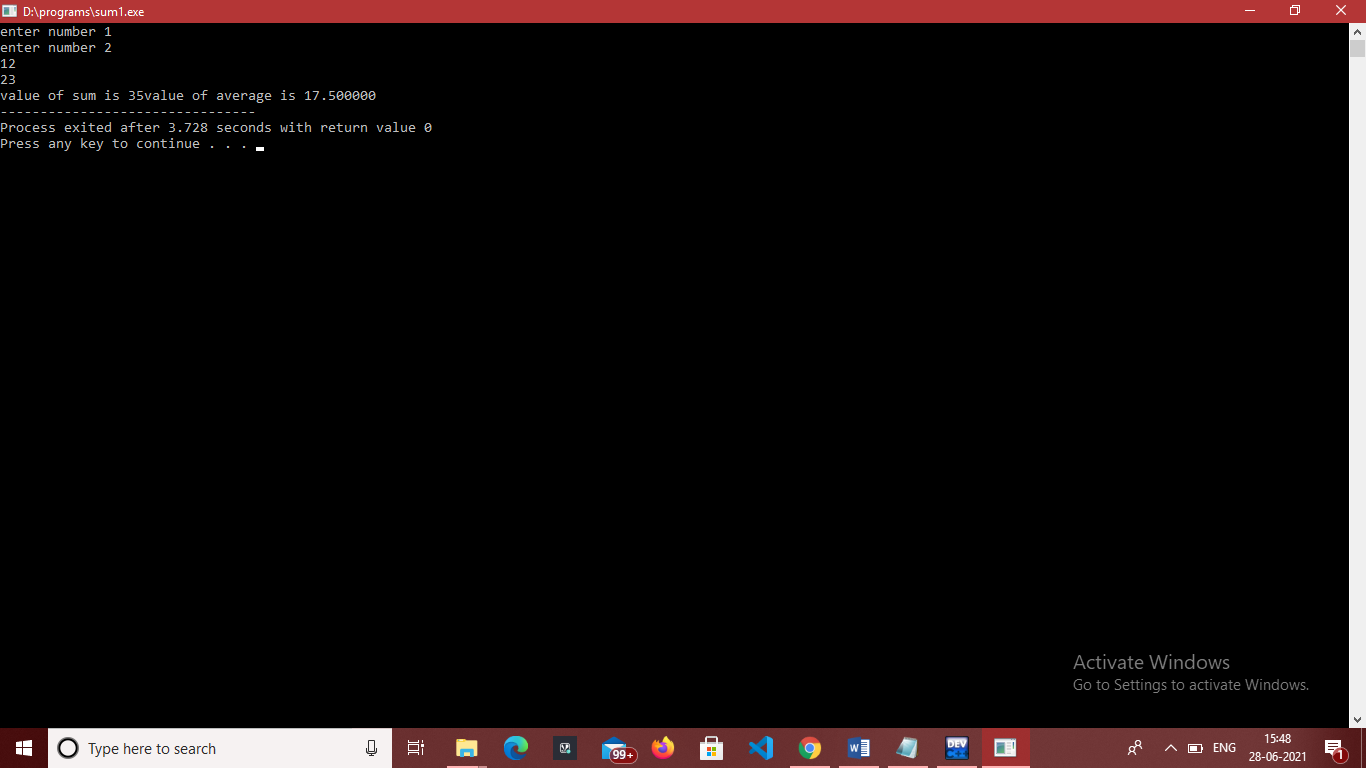
avg=(float)(sum)/2; //formula of average

printf("value of average is %f",avg);

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 2.**

**OBJECTIVE-** Program to find greatest of 10 numbers.

**INTRODUCTION-** In This Program We Input Ten Numbers and initialize a[0] greatest and then compare greatest with others and update the greatest.

**PROGRAM CODE-**

#include <stdio.h>

int main() {

int a[10];

int i;

int greatest;

printf("Enter ten values:"); //Store 10 numbers

for (i = 0; i < 10; i++) {

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

}

greatest = a[0];

for (i = 0; i < 10; i++) {

if (a[i] > greatest) {

greatest = a[i];

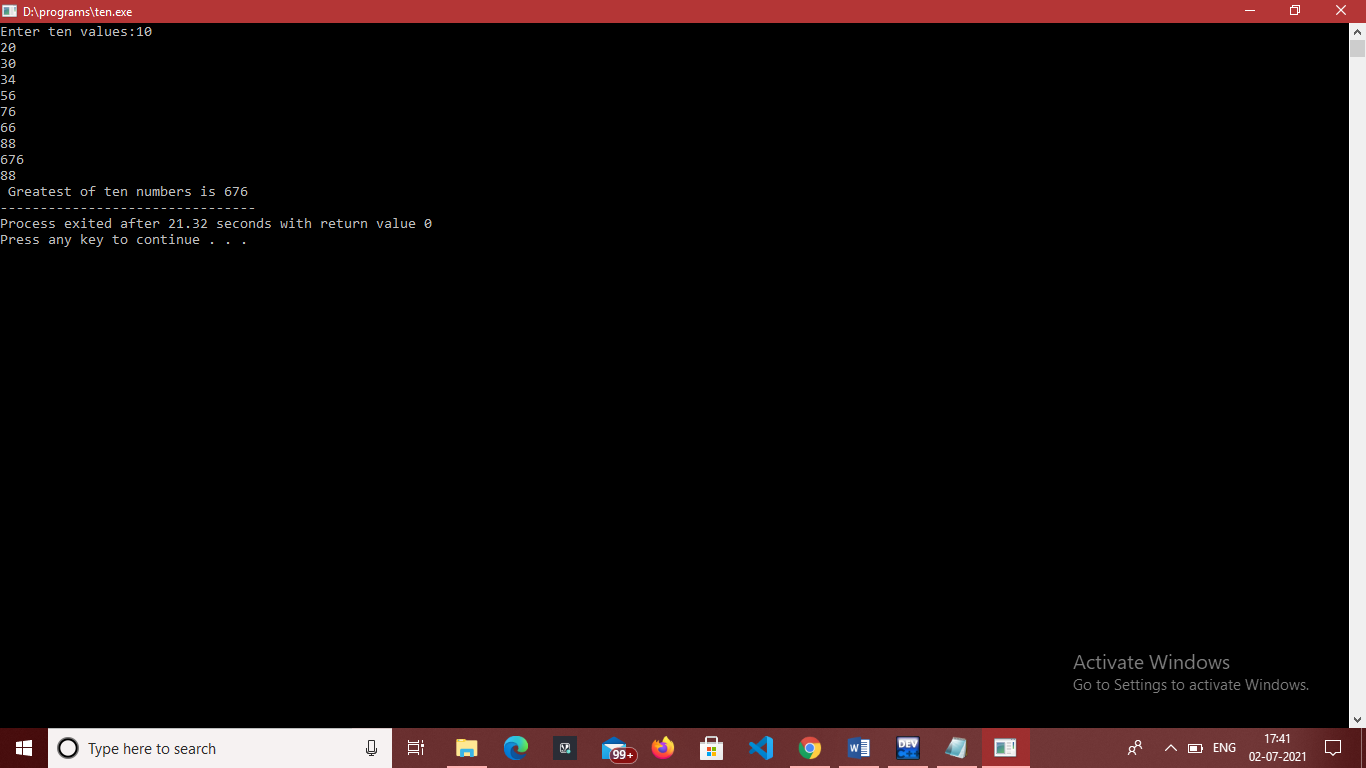
} }

printf(" Greatest of ten numbers is %d", greatest);

return 0;

}

**OUTPUT-**



**EXPERIMENT 3.**

**OBJECTIVE-** Program to find Simple Interest.

**INTRODUCTION-** In This Program We Input Principal, Rate and Time values and by using simple interest formula we calculate simple interest.

**PROGRAM CODE-**

// PROGRAM TO CALCULATE SIMPEL INTEREST

#include<stdio.h>

int main()

{

int p,r,t;

float s;

printf("enter values of principal,rate and time");

scanf("%d%d%d",&p,&r,&t);

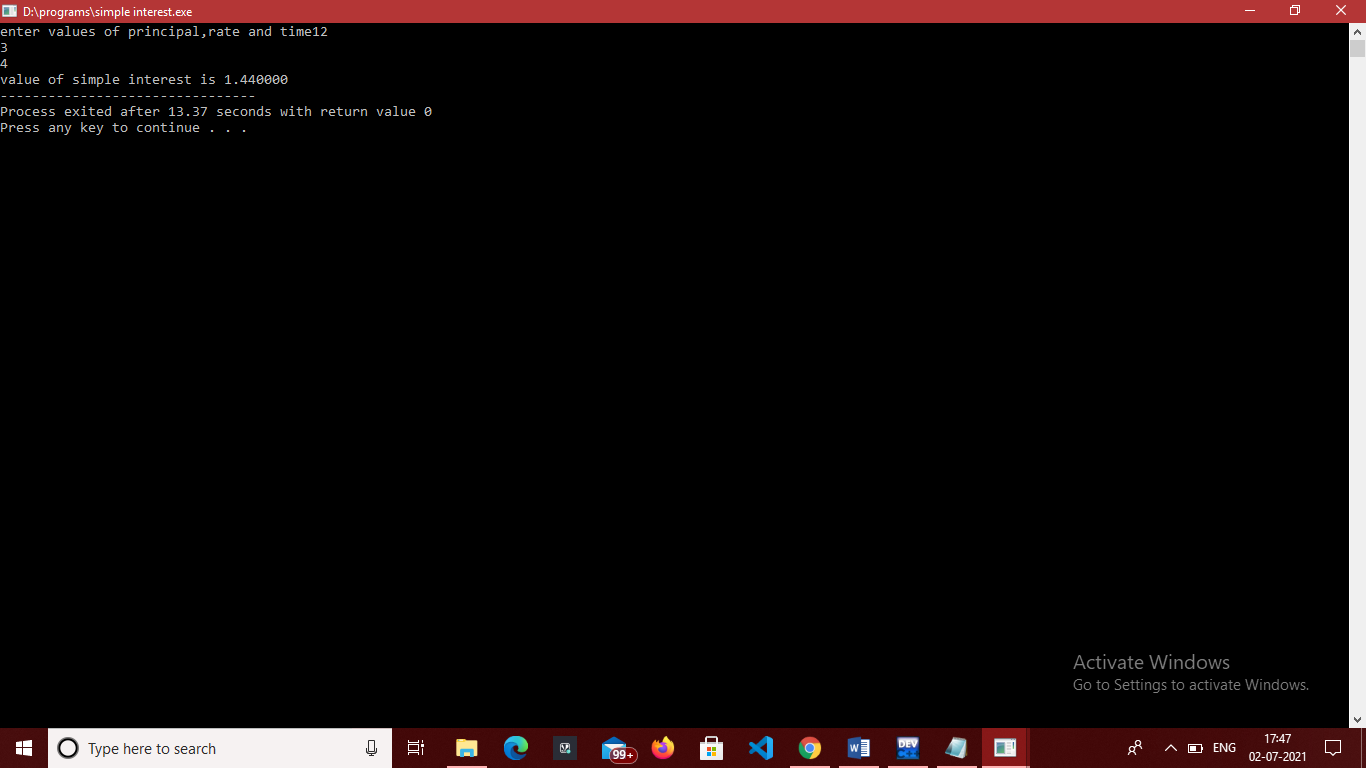
s=(float)p\*r\*t/100; //explicit type casting

printf("value of simple interest is %f",s);

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 4.**

**OBJECTIVE-** Program to print the following pattern.(triangle of stars)

**INTRODUCTION-** In This Program We Print a triangle using star first we input number of rows and by using for loop we print a star triangle.

**PROGRAM CODE-**

#include<stdio.h>

int main()

{

int r;

printf("enter no of rows\n"); //input rows

scanf("%d",&r);

int i;

for(i=1;i<=r;i++)

{ int j;

for(j=1;j<=i;j++)

{

printf("\*"); //star print

}

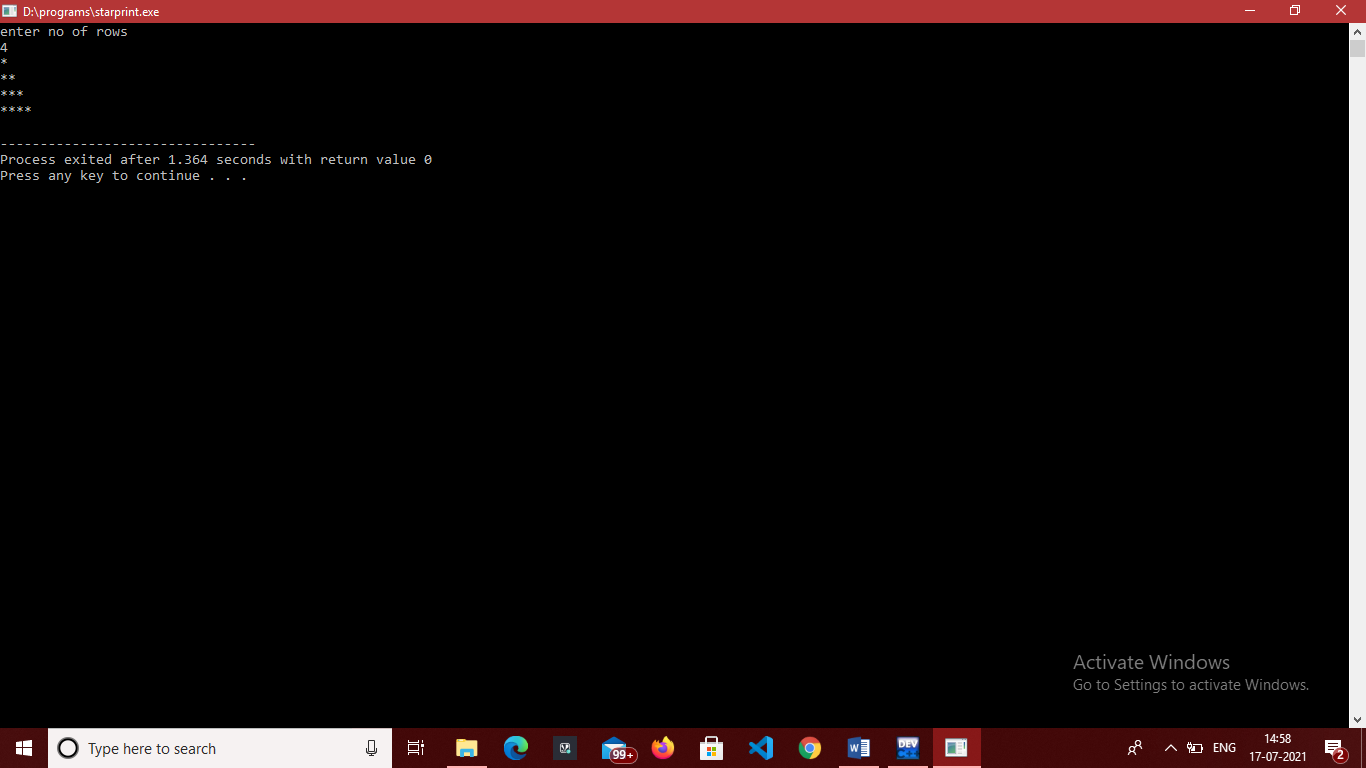
printf("\n");

}

return 0;

}

**OUTPUT-**

**/**

**EXPERIMENT 5.**

**OBJECTIVE-** Program to fine whether the entered number is prime.

**INTRODUCTION-** In This Program We Find a number is prime number or not. If a number is divisible by 1 and itseld then it is a prime number.

**PROGRAM CODE-**

#include<stdio.h>

int main()

{

int n,flag=0;

printf("enter number\n");

scanf("%d",&n);

if(n%2==0){

printf("prime");

}

if(n<=1)

{

printf("neither prime nor composite");

return 0;

}

int i;

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

{

if(n%i==0){

flag=1;

break;

}

}

if(n==1)

{

printf("1 is not composite or prime");

}

else{

if(flag==0)

{

printf("%d is a prime number",n);

}

else{

printf("%d is not a prime number",n);

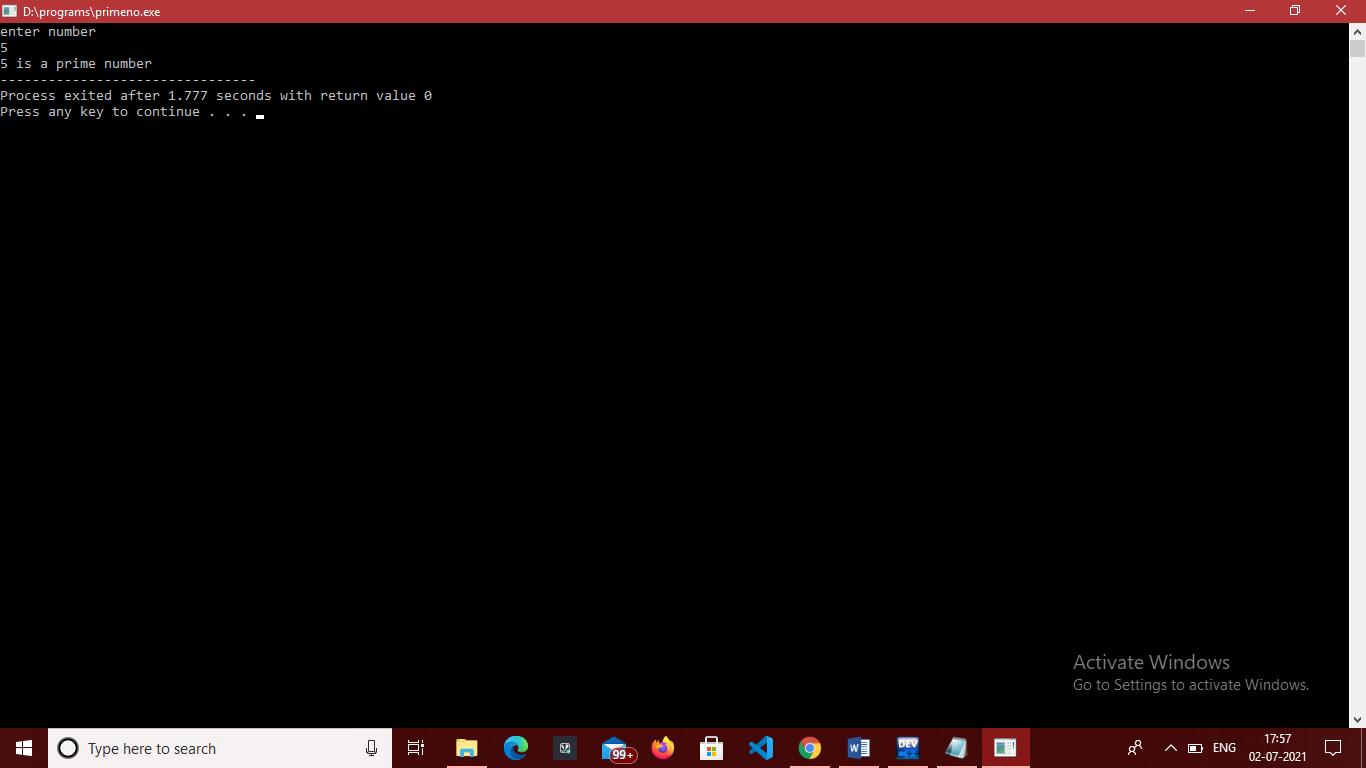
}

}

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 6.**

**OBJECTIVE-** Program to find sum of a 5 digit number.

**INTRODUCTION-** In This Program We enters a number and implement while loop number greater than 0 and in we take remainder of no and then and with sum and last number divided by 10.

**PROGRAM CODE-**

#include<stdio.h>

int main() {

int n;

scanf("%d", &n);

int digit, temp, sum = 0;

temp = n;

while(temp > 0)

{

digit = temp % 10; //remainder

sum = sum + digit;

temp = temp / 10; //short number

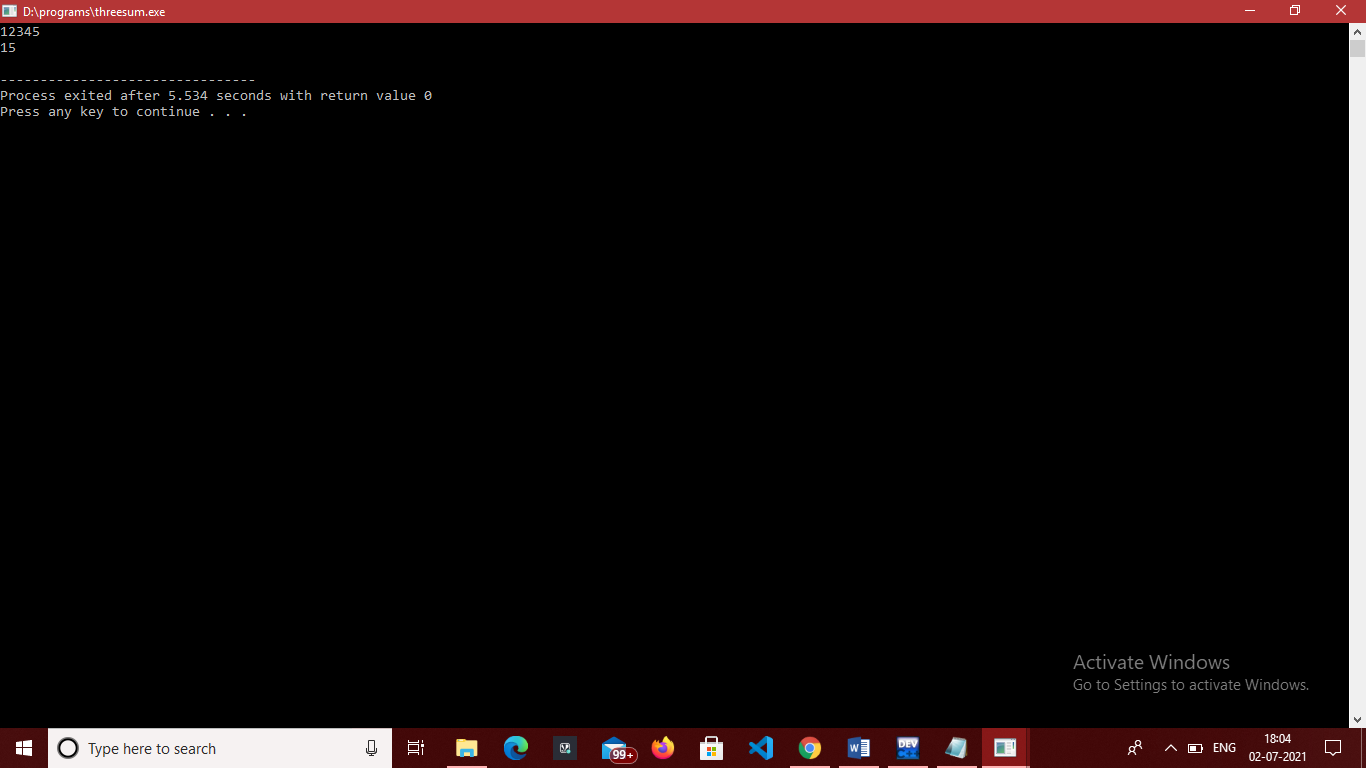
}

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

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 7.**

**OBJECTIVE-** Program to reverse a 5 digit number.

**INTRODUCTION-** In This Program We enter a number then apply while loop. in loop we take remainder then remainder added with rev\*10

Then divide number by 10.

**PROGRAM CODE-**

// REVERSE NUMBER 21-JUN-2021

#include<stdio.h>

int main()

{

int num,rev=0,rem;

printf("enter number");

scanf("%d",&num);

while(num!=0)

{ rem=num%10; //remainder

rev=rev\*10+rem;

num=num/10;

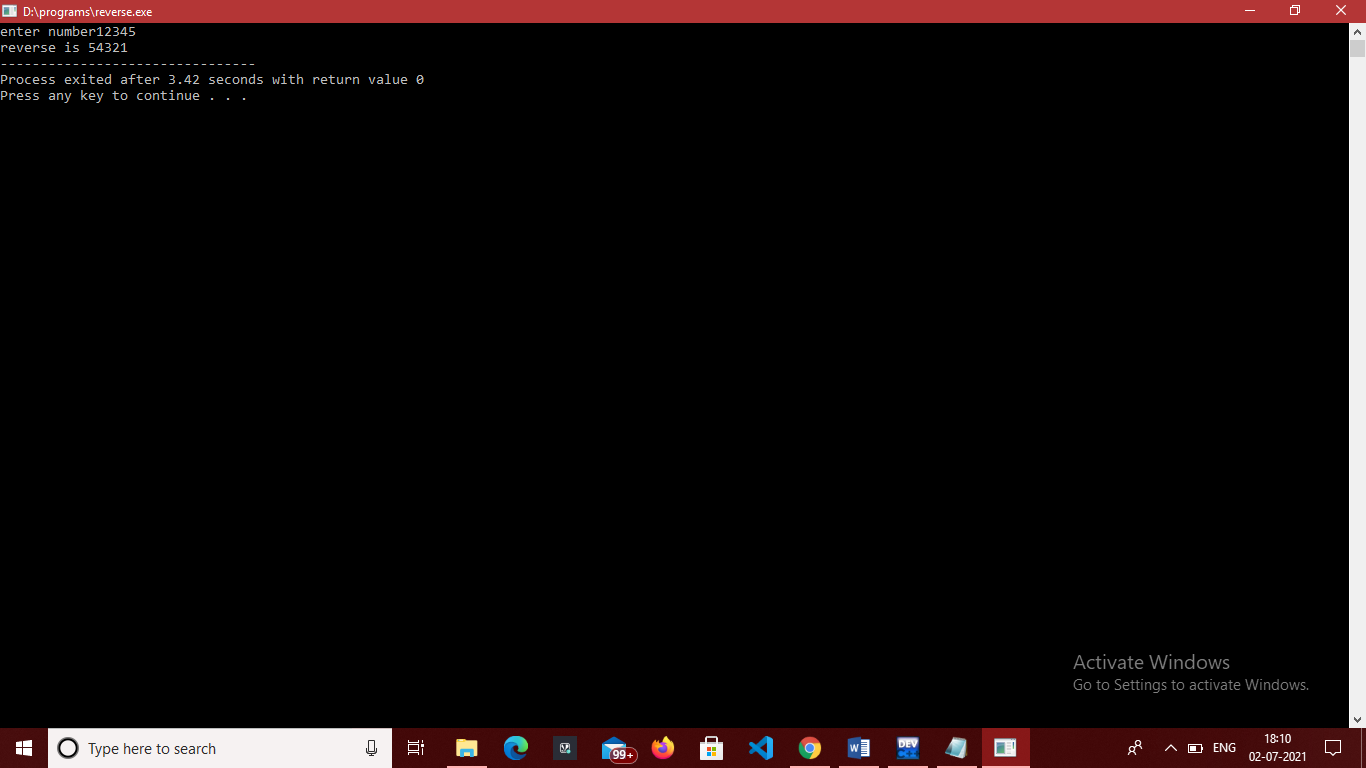
}

printf("reverse is %d",rev);

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 8.**

**OBJECTIVE-** Program to convert decimal to binary and vice versa

**INTRODUCTION-** In This Program We convert binary to decimal and decimal to binary by header file math.h and power function and our as usual remainder method to find last digit.

**PROGRAM CODE-**

//BINARY TO DECIMAL

#include <math.h>

#include <stdio.h>

int convert(long long n);

int main() {

long long n;

printf("Enter a binary number: ");

scanf("%lld", &n);

printf("%lld in binary = %d in decimal", n, convert(n));

return 0;

}

int convert(long long n) {

int dec = 0, i = 0, rem;

while (n != 0) {

rem = n % 10;

n /= 10;

dec += rem \* pow(2, i);

++i;

}

return dec;

}

// DECIMAL TO BINARY

#include <math.h>

#include <stdio.h>

int convert(long long n);

int main() {

long long n;

printf("Enter a binary number: ");

scanf("%lld", &n);

printf("%lld in binary = %d in decimal", n, convert(n));

return 0;

}

int convert(long long n) {

int dec = 0, i = 0, rem;

while (n != 0) {

rem = n % 10;

n /= 10;

dec += rem \* pow(2, i);

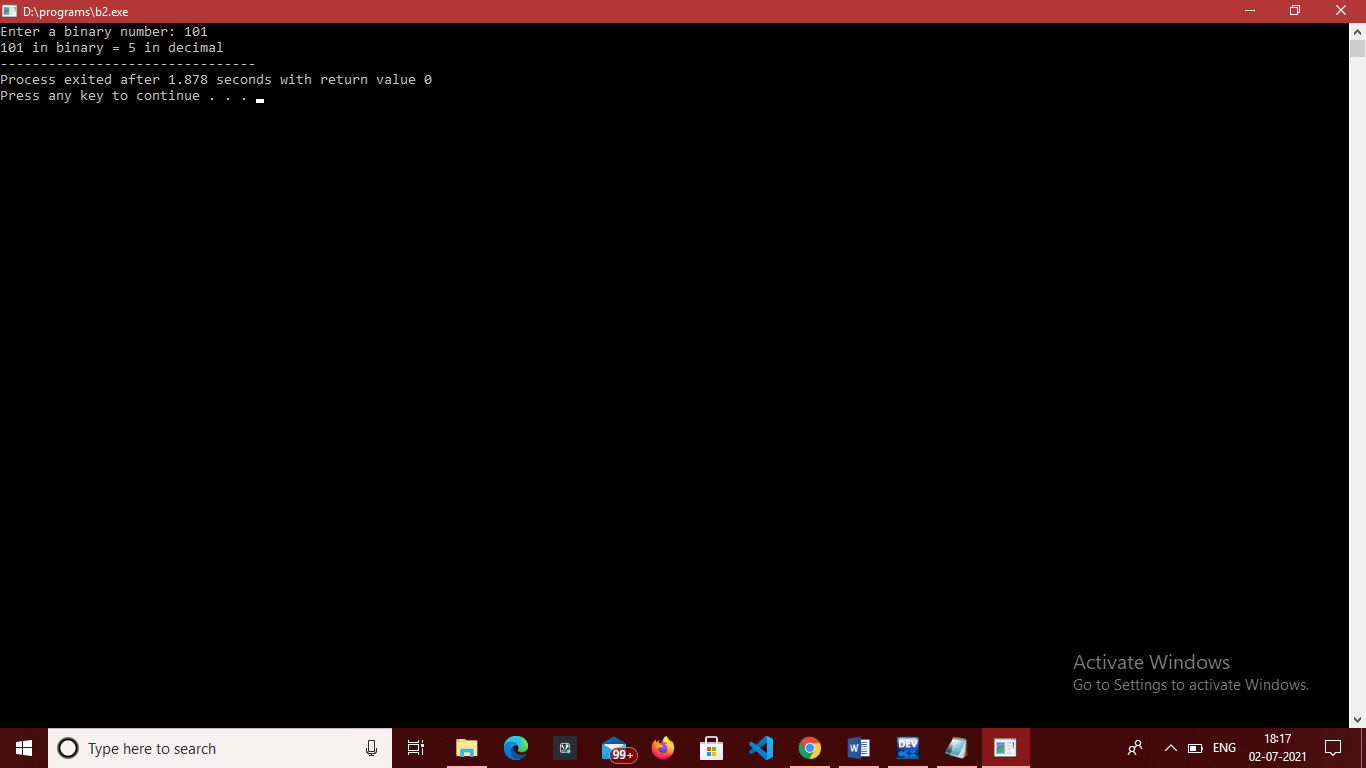
++i;

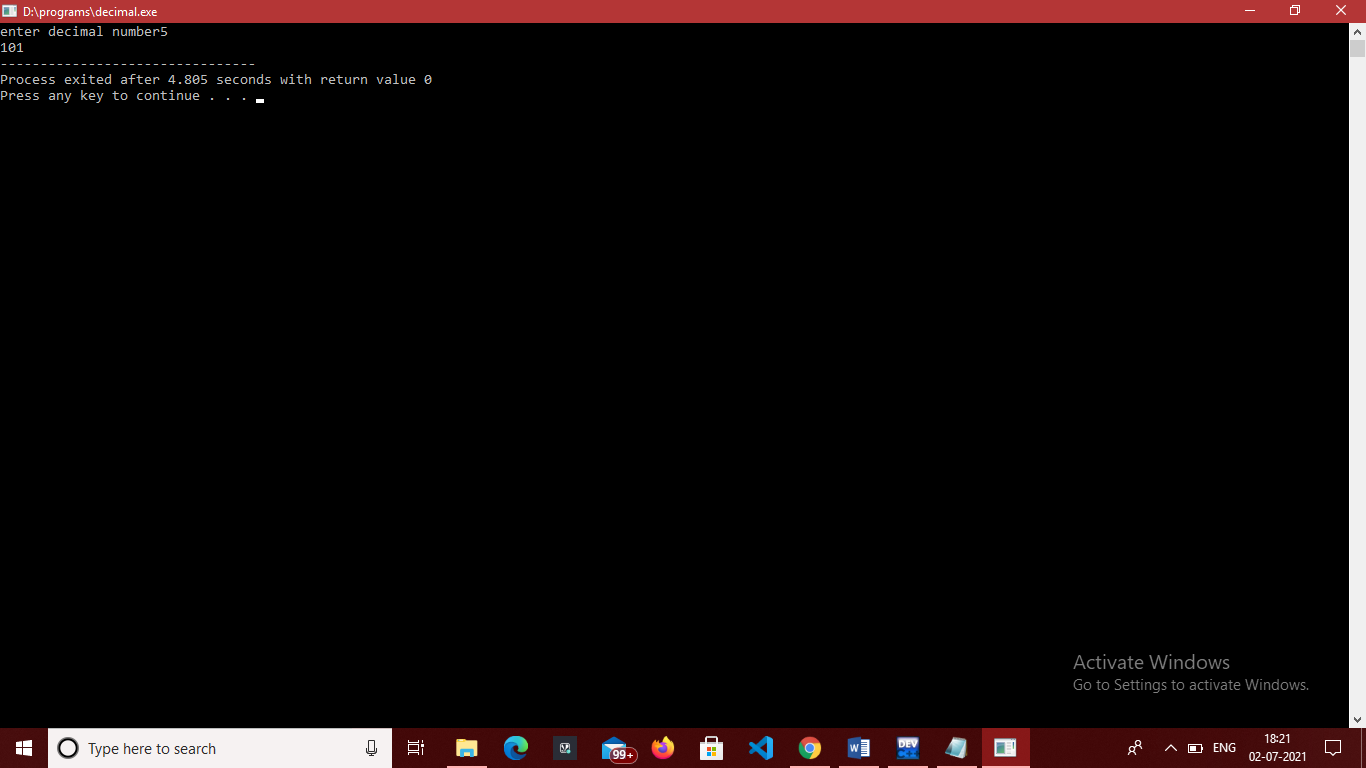
}

return dec;

}

**OUTPUT-**

****

****

**EXPERIMENT 9.**

**OBJECTIVE-** Program to implement switch case statement.

**INTRODUCTION-** In This Program We enter a number and switch case statements and break we print the days according to number for example 1 for monday.

**PROGRAM CODE-**

#include<stdio.h>

int main()

{

int n;

printf("enter number");

scanf("%d",&n);

switch(n)

{

case 1 : printf("day is monday\n");

break;

case 2 : printf("day is tuesday\n");

break;

case 3 : printf("day is wednesday\n");

break;

case 4 : printf("day is Thrusday\n");

break;

case 5 : printf("day is friday\n");

break;

case 6 : printf("day is saturday\n");

break;

case 7 : printf("day is sunday\n");

break;

default: printf("wrong input");

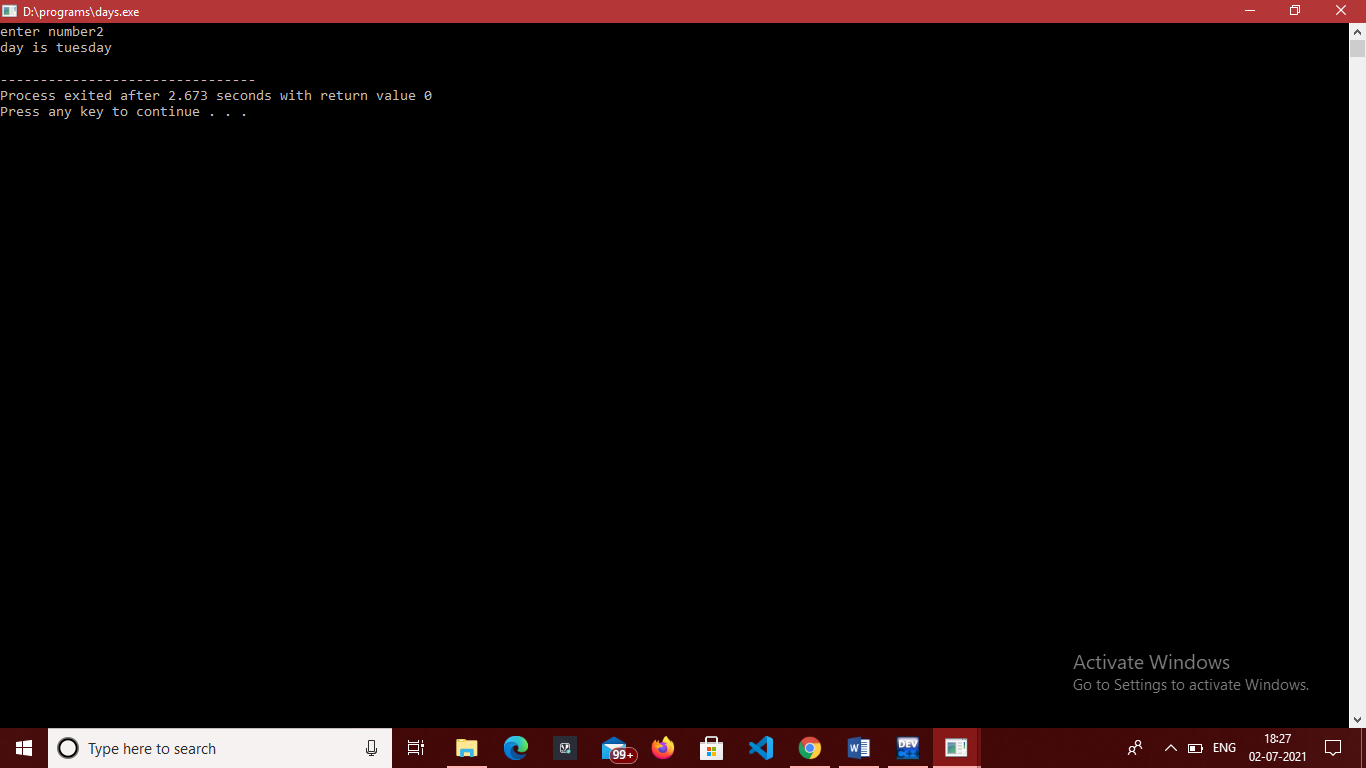
break;

}

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 10.**

**OBJECTIVE-** Program to generate the Fibonacci sequence.

**INTRODUCTION-** In This Program We initialize first with 0 and second with 1 and next=first+second and starts a for loop for i=2.

**PROGRAM CODE-**

#include<stdio.h>

int main()

{

int next,n,first=0,second=1;

next=first+second;

printf("enter number of terms\n");

scanf("%d",&n);

printf("fibonacci series of %d %d\t",first ,second);

int i;

for(i=2;i<=n;i++)

{

printf("%d \t",next);

first=second;

second=next;

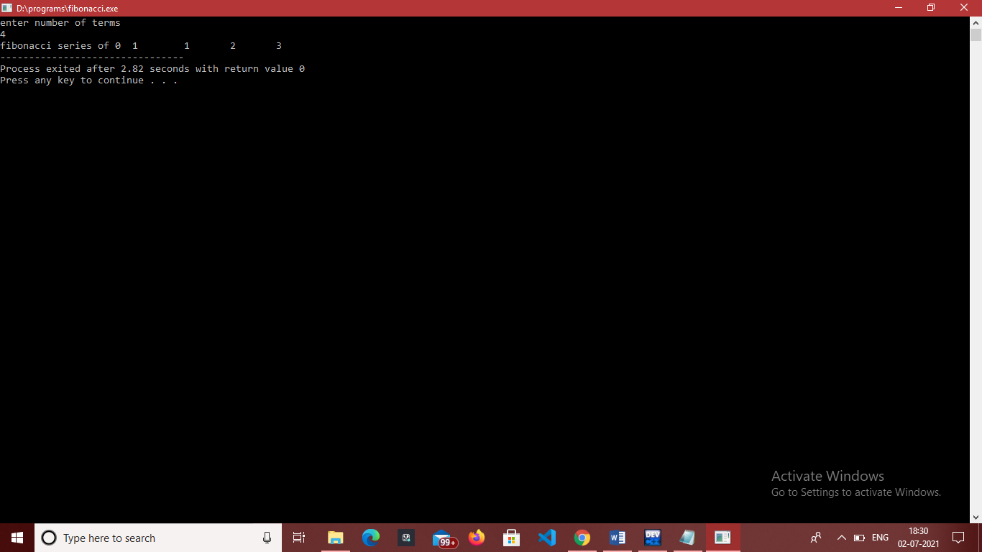
next=first+second;

}

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 11.**

**OBJECTIVE-** Program to find exponential function.

**INTRODUCTION-** In This Program first we a function for calculating factorial of a number and and a function for calculating exponential and in main function we inputs a value for x.

**PROGRAM CODE-**

#include<stdio.h>

#include<math.h>

int fact(int n) //function

{

int factorial=1;

int i;

for(i=1;i<=n;i++)

{

factorial=factorial\*i;

}

return factorial;

}

float expn(int x,int n)

{

float a=((float)pow(x,n)/(float)fact(n)); //function call

return a;

}

int main()

{

int x,ans=0;

printf("enter value of x \n");

scanf("%d ",&x);

int i;

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

{

ans+=expn(x,i);

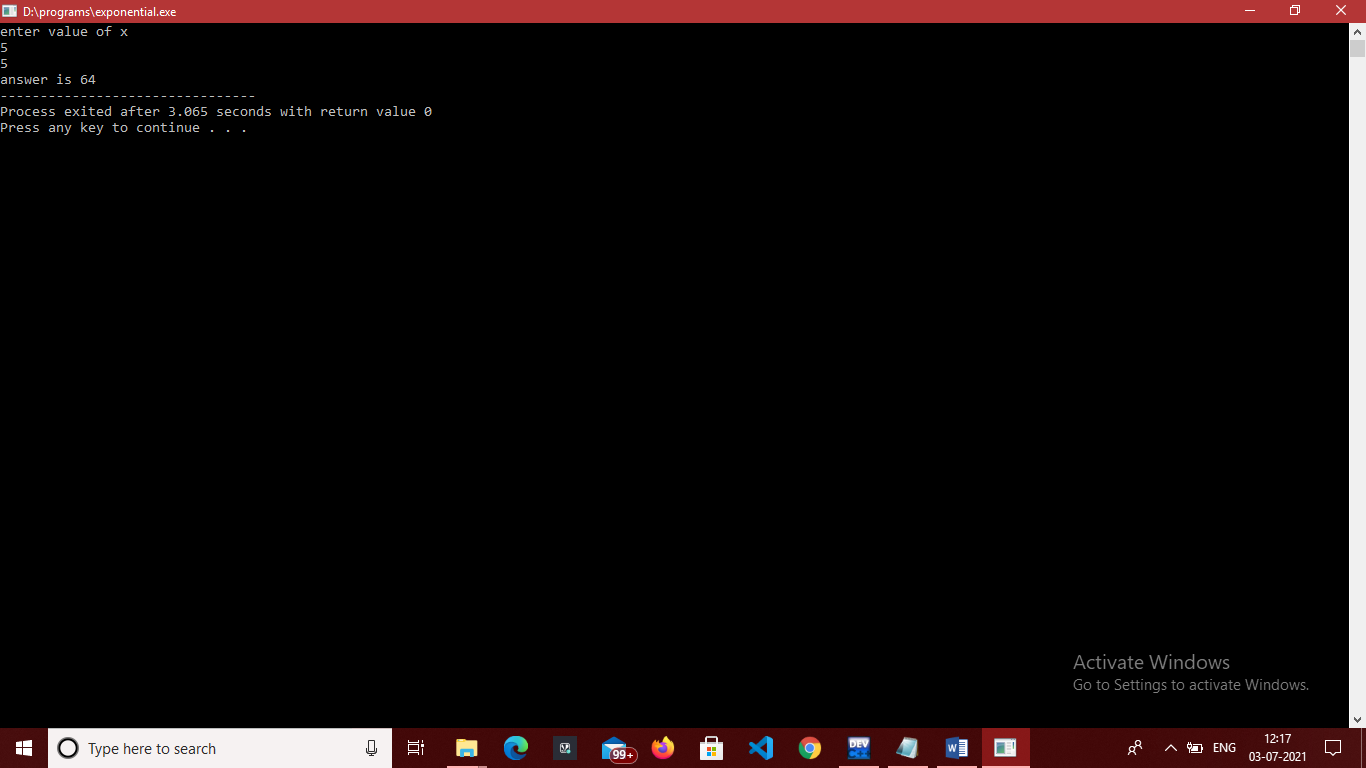
}

printf("answer is %d",ans);

return 0;

}

**OUTPUT-**



**EXPERIMENT 12.**

**OBJECTIVE-** Program to search a number from an array using linear search.

**INTRODUCTION-** In This Program first we a enters a array size then enters array array elements then ask user to enter element he wants to search then using linear search algorithm find an element.

**PROGRAM CODE-**

#include<stdio.h>

int Lsearch(int [],int ,int);

int main()

{

int a[5],item,n,index;

printf("enter array size");

scanf("%d",&n);

printf("enter array elements");

int i;

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

{

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

}

printf("enter eleemnt to be seach");

scanf("%d",&item);

index=Lsearch(a,n,item);

if(index==-1)

{

printf("not found");

}

else

{

printf("element found at index: %d position %d",index,index+1);

}

return 0;

}

int Lsearch(int a[], int size, int item)

{

int i;

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

{

if(a[i]==item)

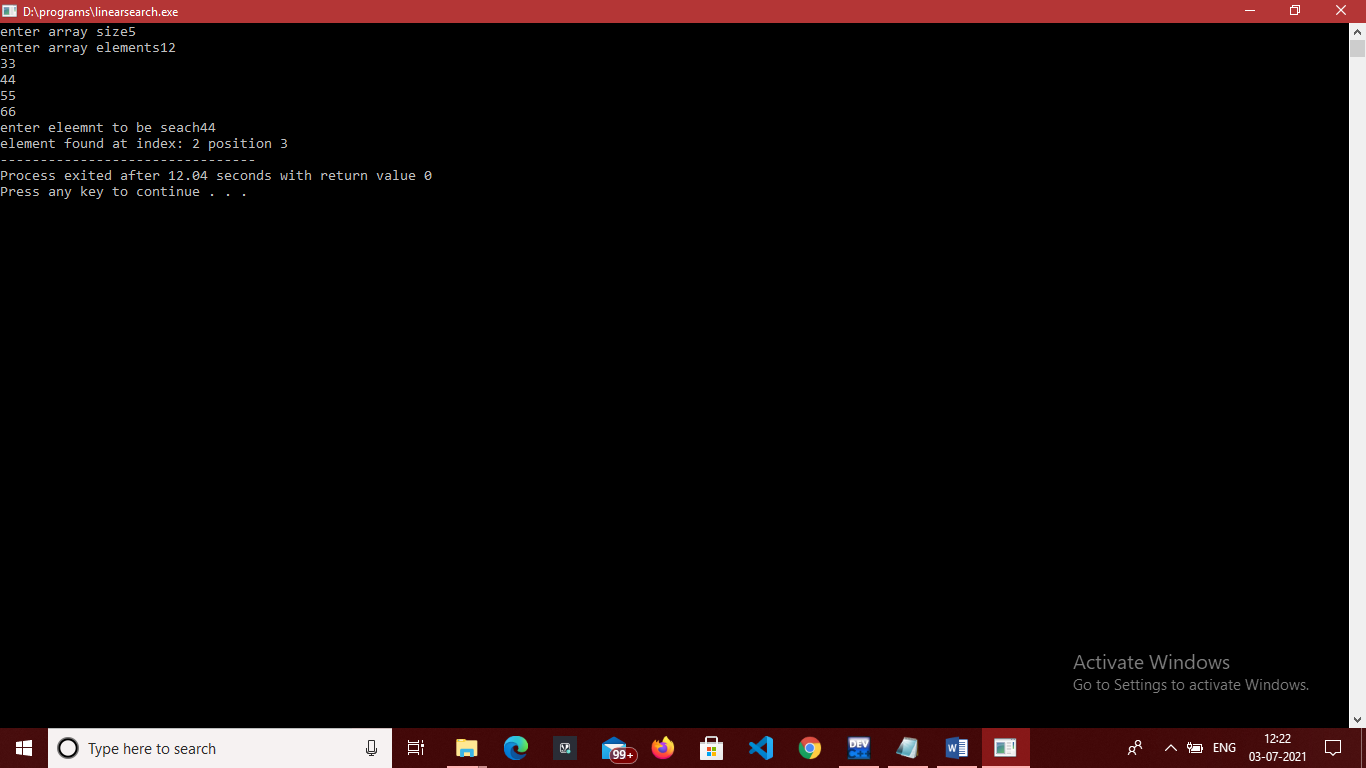
return i;

}

return -1;

}

**OUTPUT-**

****

**EXPERIMENT 13.**

**OBJECTIVE-** Program to search a number from an array using binary search.

**INTRODUCTION-** In This Program first we a function named Bsearch for finding an element with comparing with middle in this method we reduce the time complexity.

**PROGRAM CODE-**

#include<stdio.h>

int Bsearch(int[],int,int);

int main()

{

int a[10],item,n,index;

printf("enter array size");

scanf("%d",&n);

printf("enter array elements");

int i;

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

{

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

}

printf("enter eleemnt to be seach");

scanf("%d",&item);

index=Bsearch(a,n,item);

if(index==-1)

{

printf("not found");

}

else

{

printf("element found at index: %d position %d",index,index+1);

}

return 0;

}

int Bsearch(int a[],int size,int item) //function declare

{

int start,end,mid;

start=0;

end=size-1;

{

while(start<=end)

{

mid=(start+end/2);

if(item==a[mid])

return mid;

else if(item>a[mid])

start=mid+1;

else

{

end=mid-1;

}

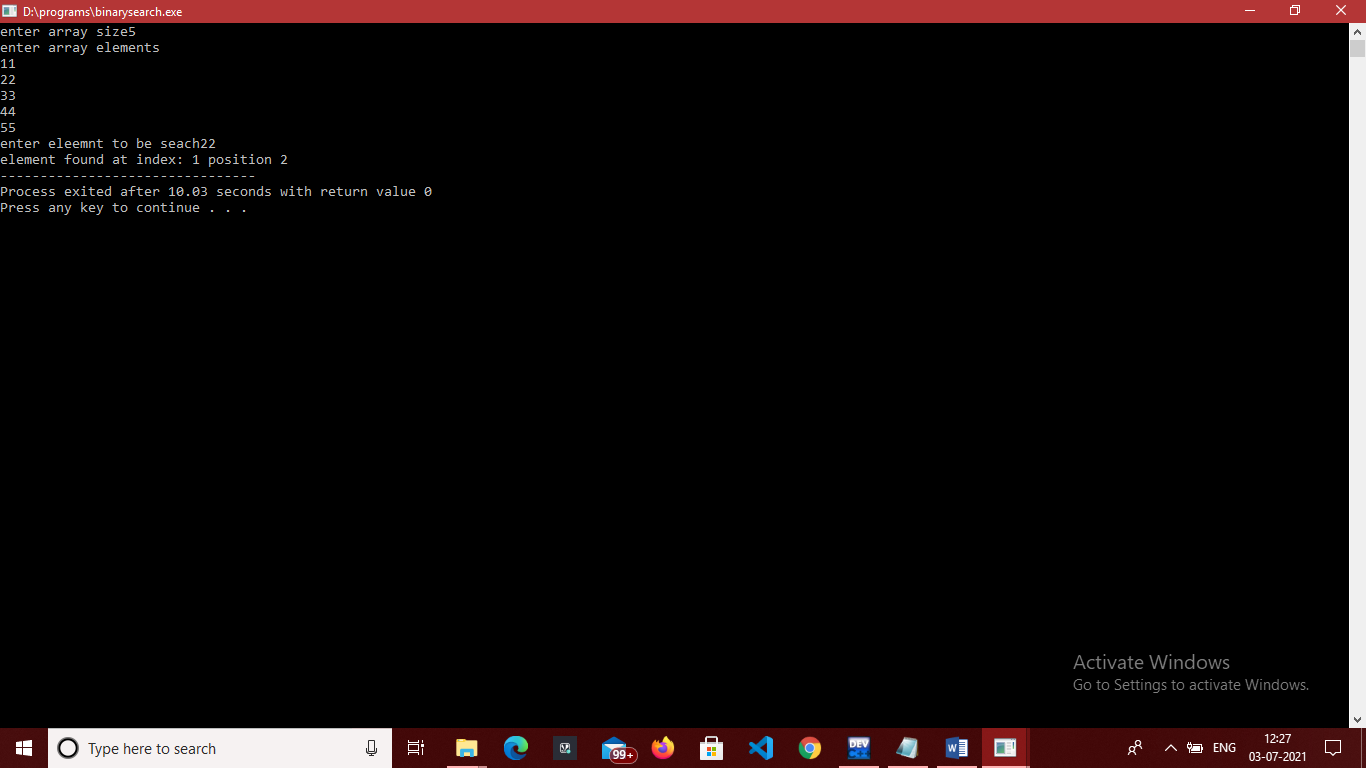
}

return -1;

}

}

**OUTPUT-**



**EXPERIMENT 14.**

**OBJECTIVE-** Program to sort an array using Bubble sort.

**INTRODUCTION-** In This Program first we compare first element with the next one and store it into temp and by using for loop we sort the complete array.

**PROGRAM CODE-**

#include<stdio.h>

void bubblesort(int[],int);

int main()

{

int a[20],item,n,index;

printf("enter array size");

scanf("%d",&n); //input size

printf("enter array elements");

int i;

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

{

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

}

bubblesort(a,n);

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

{

printf("%d \n",a[i]);

}

return 0;

}

void bubblesort(int a[], int size)

{

int i,temp;

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

{

int j;

for(j=0;j<size-i-1;j++)

{

if(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

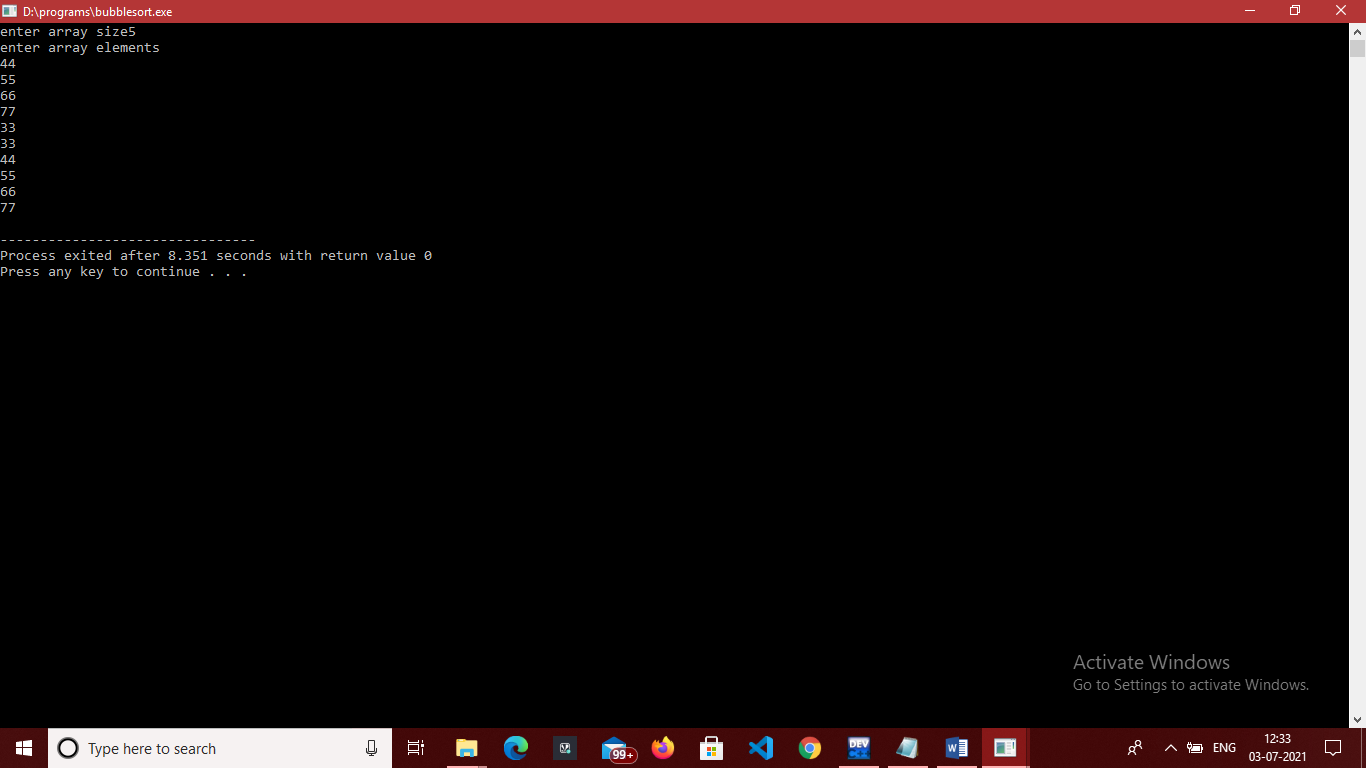
}

}

}

}

**OUTPUT-**

****

**EXPERIMENT 15.**

**OBJECTIVE-** Program to sort an array using selection sort.

**INTRODUCTION-** In This Program first we first initialize position to d and in if we compare all elements with position and sorted the array.

**PROGRAM CODE-**

#include <stdio.h>

int main()

{

int array[100], n, c, d, position, t;

printf("Enter number of elements\n");

scanf("%d", &n);

printf("Enter %d integers\n", n);

for (c = 0; c < n; c++)

scanf("%d", &array[c]);

for (c = 0; c < (n - 1); c++) // finding minimum element (n-1) times

{

position = c;

for (d = c + 1; d < n; d++)

{

if (array[position] > array[d])

position = d;

}

if (position != c)

{

t = array[c];

array[c] = array[position];

array[position] = t;

}

}

printf("Sorted list in ascending order:\n");

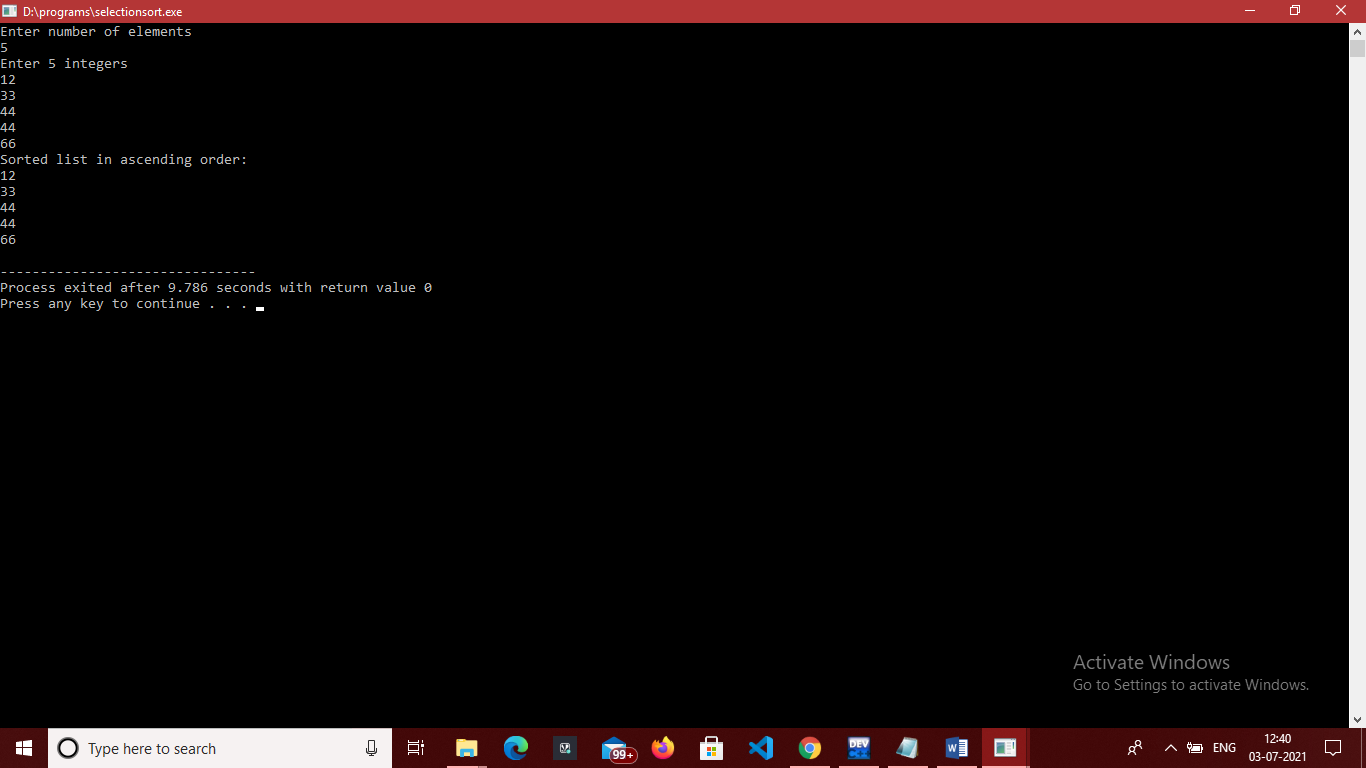
for (c = 0; c < n; c++)

printf("%d\n", array[c]);

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 16.**

**OBJECTIVE-** Program to sort an array using insertion sort.

**INTRODUCTION-** In this program we iterate from arr[1] to arr[n] over the array compare the current element with its predecessor If the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element.

**PROGRAM CODE-**

#include <stdio.h>

int main()

{

int n, array[1000], c, d, t, flag = 0;

printf("Enter number of elements\n");

scanf("%d", &n);

printf("Enter %d integers\n", n);

for (c = 0; c < n; c++)

scanf("%d", &array[c]);

for (c = 1 ; c <= n - 1; c++) {

t = array[c];

for (d = c - 1 ; d >= 0; d--) {

if (array[d] > t) {

array[d+1] = array[d];

flag = 1;

}

else

break;

}

if (flag)

array[d+1] = t;

}

printf("Sorted list in ascending order:\n");

for (c = 0; c <= n - 1; c++) {

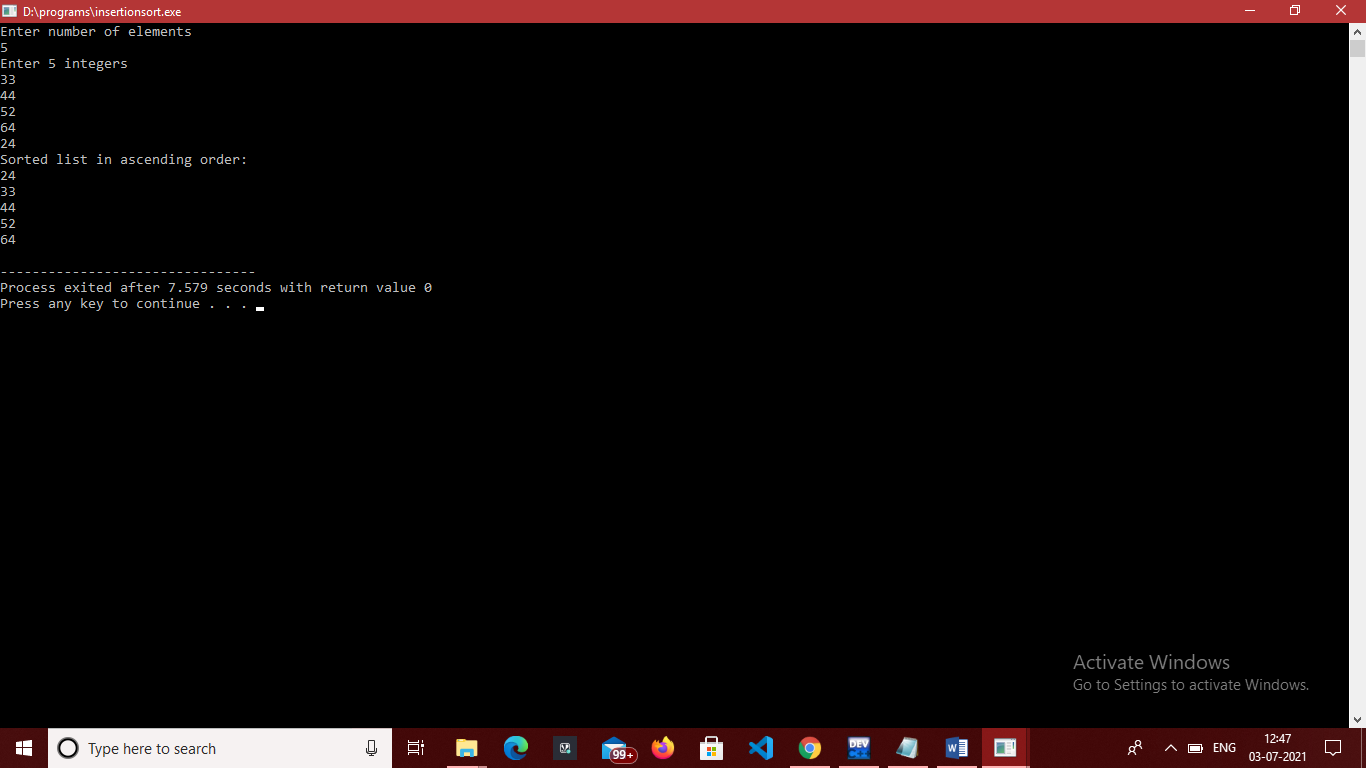
printf("%d\n", array[c]);

}

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 17.**

**OBJECTIVE-** Program to find factorial of a number using recursion.

**INTRODUCTION-** In This Program first we calculate the factorial of a number recursively first make a function fact and use base case as n==0 or n==1 and then again call fact.

**PROGRAM CODE-**

#include<stdio.h>

int fact(int n) // function declare

{

if(n==0) //base case

{

return 1;

}

if(n==1)

{

return 1;

}

int smallans=fact(n-1); //function call

return n\*smallans;

}

int main()

{

int n;

printf("enter number");

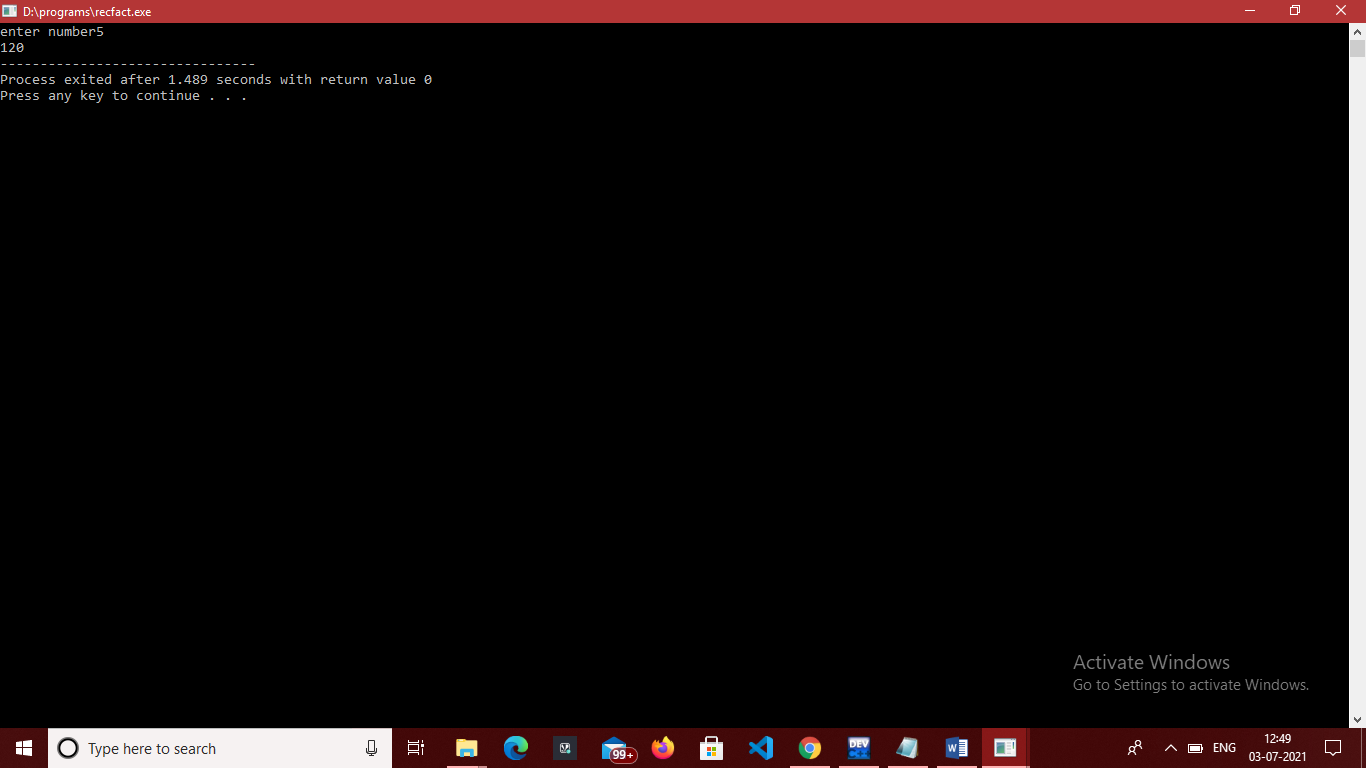
scanf("%d",&n);

printf("%d",fact(n));

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 18.**

**OBJECTIVE-** Program to find the length of the string without using strlen and then pass the string to characters.

**INTRODUCTION-** In This Program first we calculate the length of string firstly we take input of string and then make a function to count the no of characters in it.

**PROGRAM CODE-**

#include<stdio.h>

#include<string.h>

int xstrlen(char str[10])

{

int count=0;

int i=0;

while(str[i]!='\0') //TO READ STRING

{

count++;

i++;

}

printf("Length of string is: %d ",count); //PRINT COUNT

return 0;

}

int main()

{

char str[10];

printf("enter string\n");

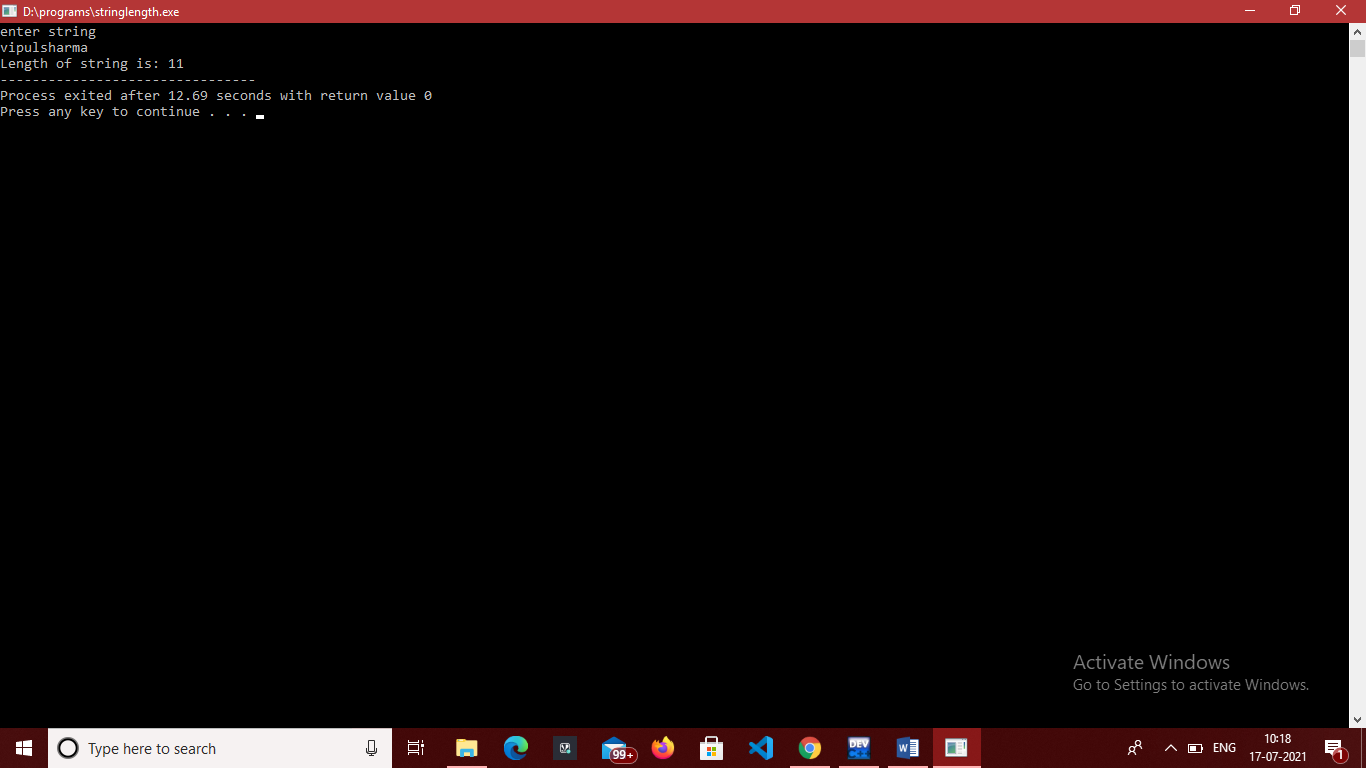
gets(str); //STRING INPUT

xstrlen(str); //FUNCTION CALL

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 19.**

**OBJECTIVE-** Program to count the number of vowels in a given string.

**INTRODUCTION-** In This Program first we calculate the number of vowels in a string firstly we take input and then check the condition using if. If condition is true then count++.

**PROGRAM CODE-**

#include<stdio.h>

#include<string.h>

int main()

{

char str[10];

int count=0;

printf("enter string\n");

gets(str); //input string

int i;

for(i=0;i<strlen(str);i++) //loop to traverse string

{

if(str[i]=='a'||str[i]=='i'||str[i]=='o'||str[i]=='e'||str[i]=='u'||str[i]=='A'||str[i]=='I'||str[i]=='O'||str[i]=='U'||str[i]=='E')

{

count++; //to count vowels

}

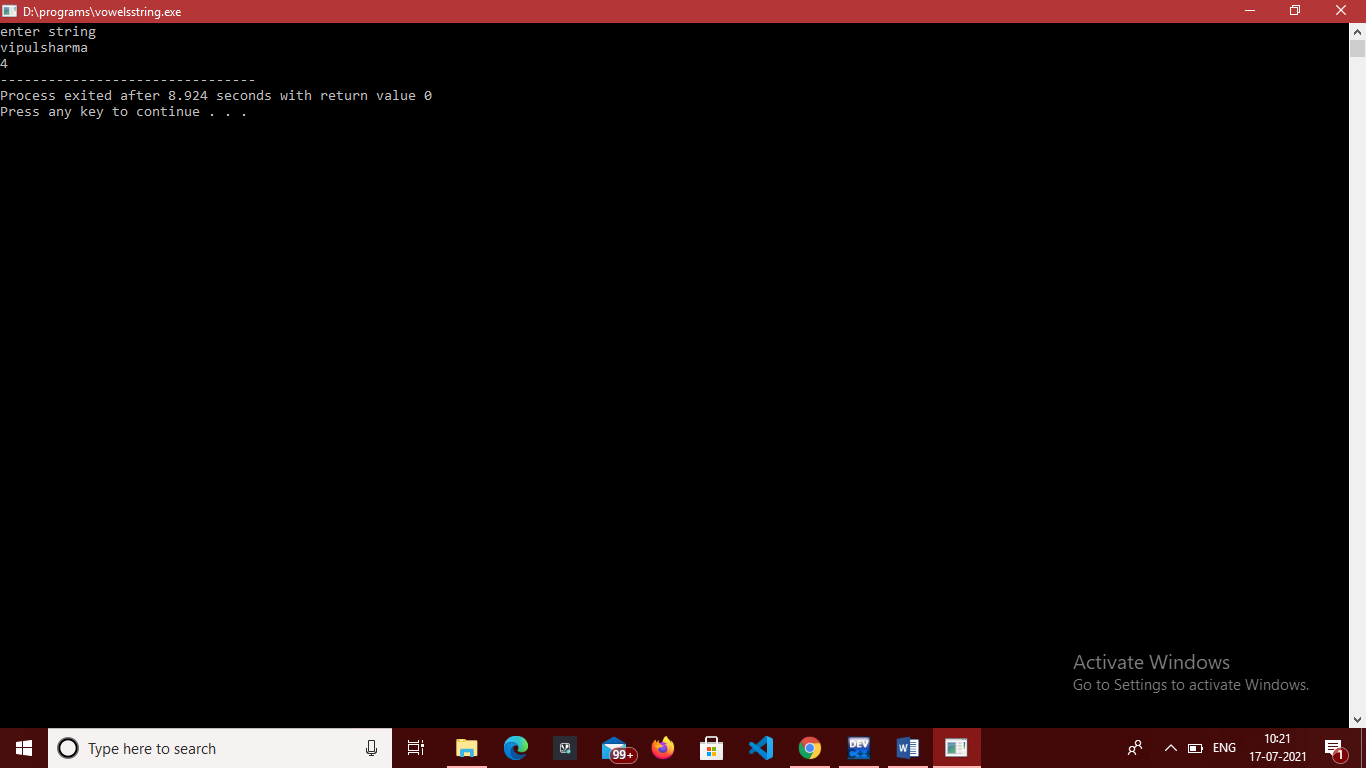
}

printf("%d",count); //print count

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 20.**

**OBJECTIVE-** . Program to check if a given string is a palindrome or not

**INTRODUCTION-** In This Program we want to check a string is palindrome or no. a string is palindrome when first check same as last second same as second last and so on.

**PROGRAM CODE-**

#include<stdio.h>

#include<string.h>

char checkpalin(char str[200]) //function to check palindrome

{

int len=strlen(str);

int flag=0;

int i;

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

{

if(str[i]!=str[len-i-1])

{

flag=1;

break;

}

}

if(flag==0)

{

printf("%s is palindrome",str);

}

else

{

printf("%s not palindrome",str);

}

return 0;

}

int main() //starting of program

{

char str1[200];

printf("enter string\n");

gets(str1); //input string

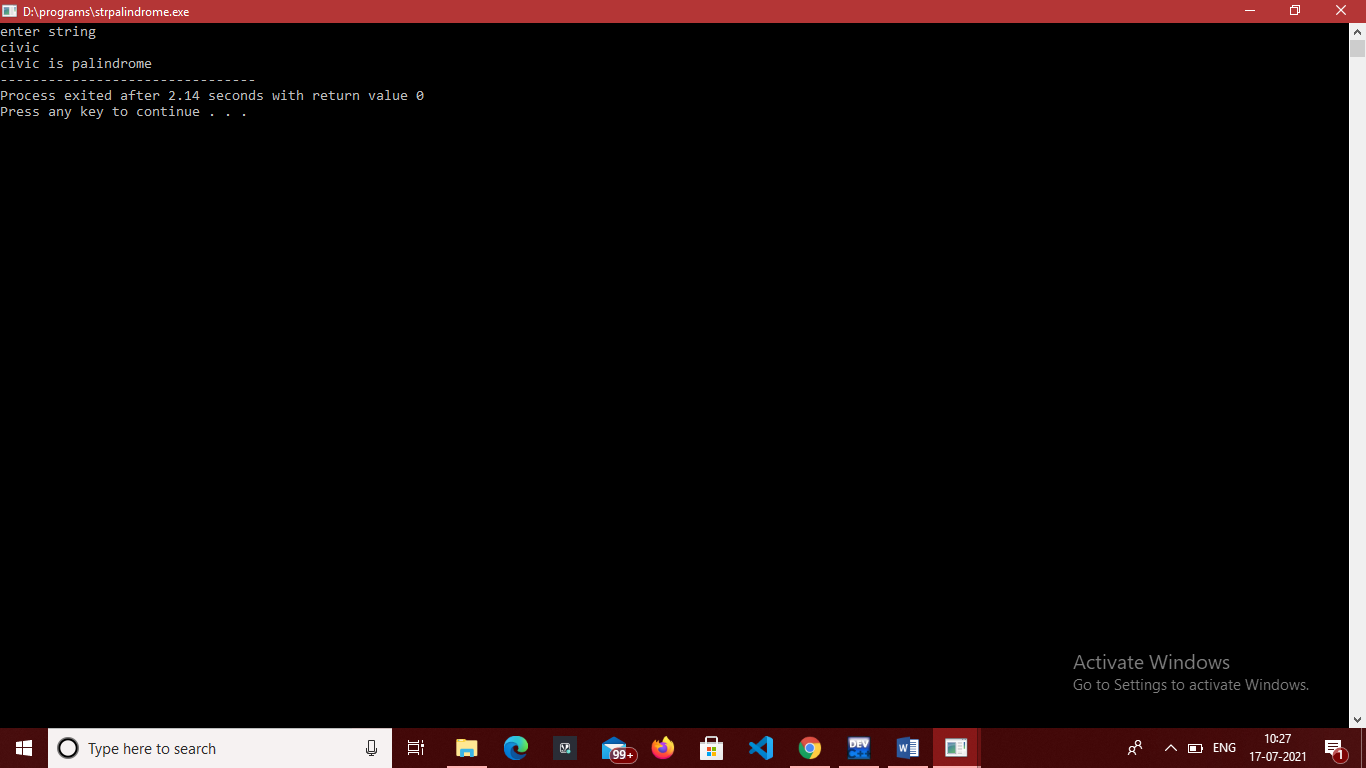
int flag;

checkpalin(str1); //function call

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 21.**

**OBJECTIVE-** . Program to string concatenation.

**INTRODUCTION-** In This Program first we concatenate the two strings. we take input two strings then by using for and while loop concatenate strings.

**PROGRAM CODE-**

#include <stdio.h>

int main() {

char s1[100], s2[100];

printf("enter string\n");

gets(s1); //input string one

printf("enter string\n");

gets(s2); //input string two

int length, j;

length = 0;

while (s1[length] != '\0') {

++length; //length increment

}

for (j = 0; s2[j] != '\0'; ++j, ++length) {

s1[length] = s2[j];

}

s1[length] = '\0';

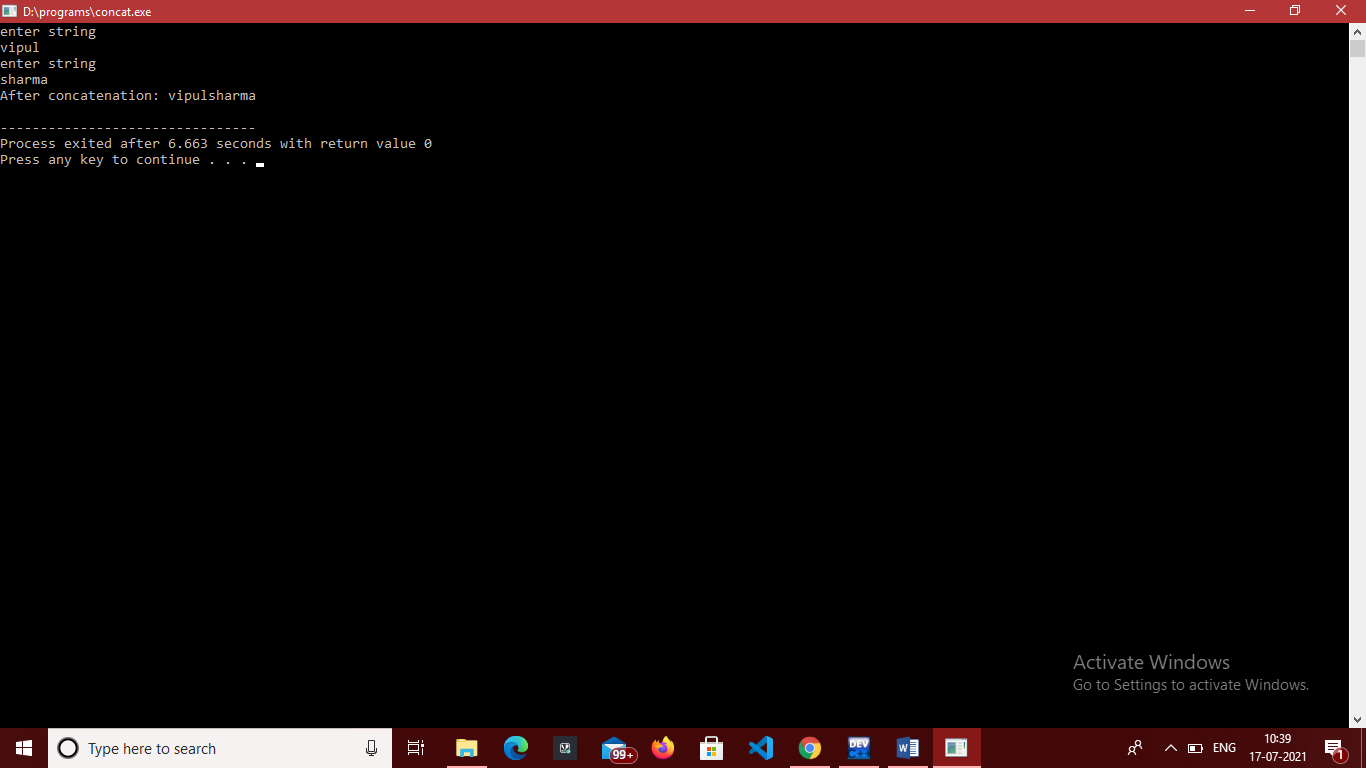
printf("After concatenation: ");

puts(s1); //print after concatenation

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 22.**

**OBJECTIVE-** . Program to string comparison

**INTRODUCTION-** In This Program first we compare two strings are equal of not using loop we check first character is equal to string second first character if not then result equals to 1.

**PROGRAM CODE-**

#include <stdio.h>

#include <string.h>

int main()

{

char str1[100], str2[100];

int result=0, i,j;

printf("\n Please Enter the First String : ");

gets(str1); //input string 1

printf("\n Please Enter the Second String : ");

gets(str2); //input string 2

for(i=0;i<strlen(str1);i++) // for loop tranverse str 1

{

if(str1[i]!=str2[i]) // if condition

{

result=1;

break;

}

}

if(result==0)

{

printf("\n str1 is Equal to str2");

}

else

{

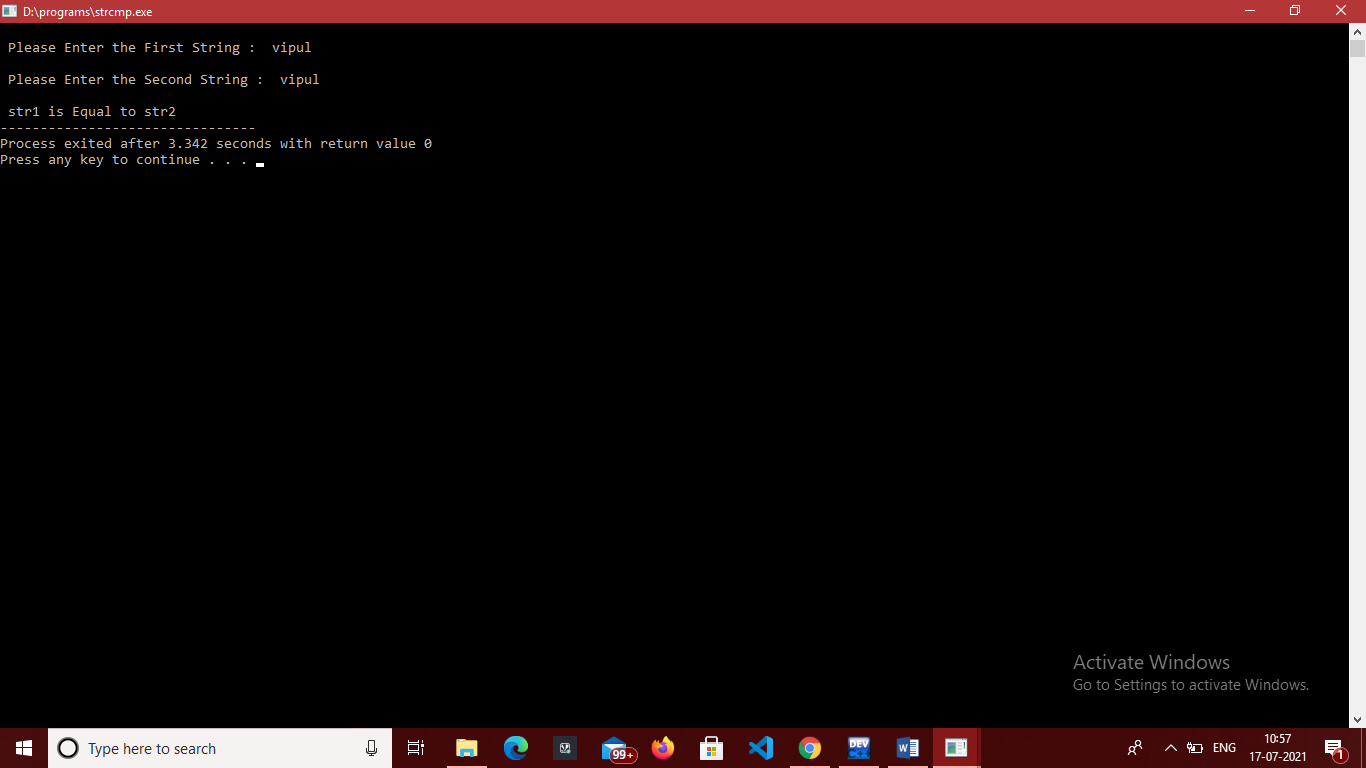
printf("\n str1 is not Equal to str2");

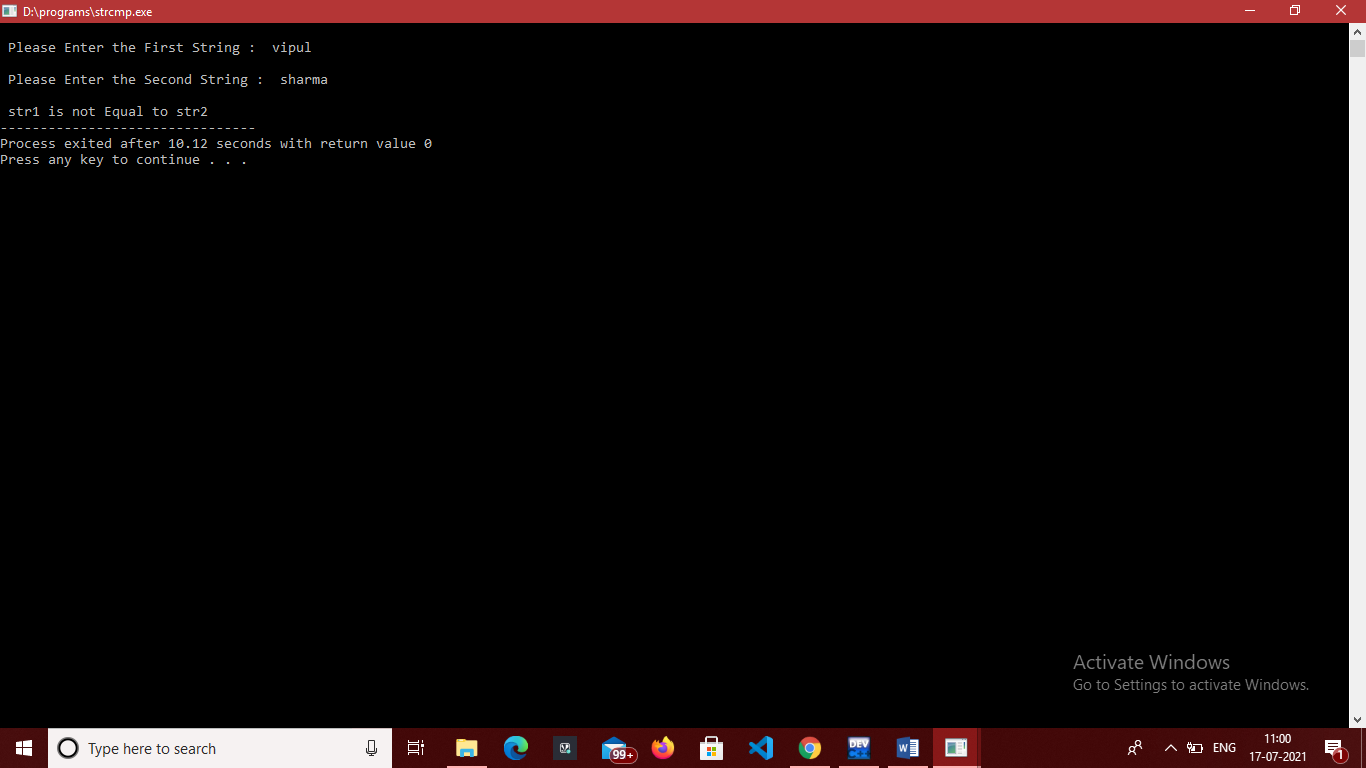
}

return 0;

}

**OUTPUT-**

****

****

**EXPERIMENT 23.**

**OBJECTIVE-** . Program to string reverse.

**INTRODUCTION-** In This Program reverse a string by using temporary variable I make a function named rev to reverse a string in which I use for loop.

**PROGRAM CODE-**

#include<stdio.h>

#include<string.h>

char rev(char str[10]) //function to reverse a string

{

int len=strlen(str);

char temp;

int i;

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

{

temp=str[i];

str[i]=str[len-1];

str[len-1]=temp;

len--;

}

}

int main()

{

char str[10];

printf("enter string: \n");

gets(str); //string input

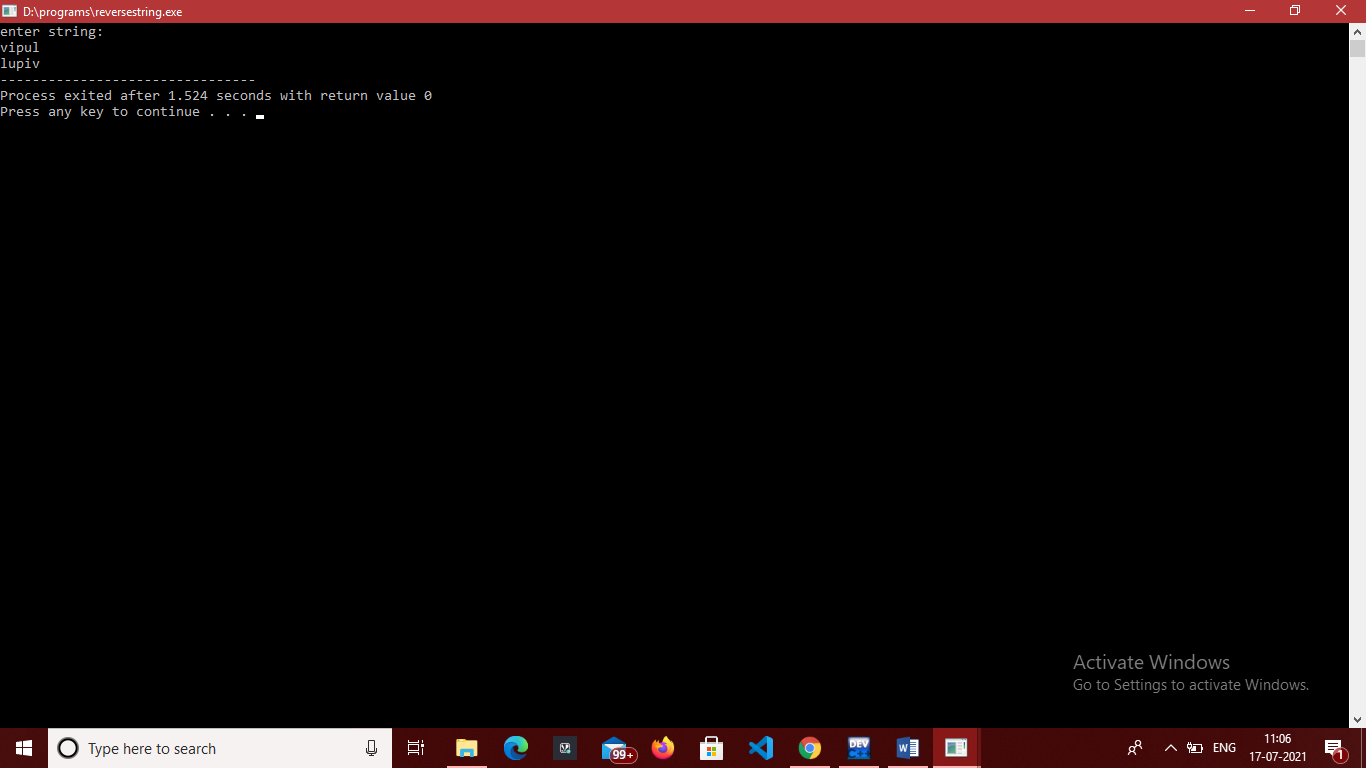
rev(str); //function call

printf("%s",str);

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 24.**

**OBJECTIVE-** Program to convert a string from lower case to upper case and vice versa

**INTRODUCTION-** In This Program we convert lowercase into uppercase and viva-versa we subtract and add 32 to convert string.

**PROGRAM CODE-**

#include<stdio.h>

#include<string.h>

int main()

{

char str[20];

printf("enter string\n");

gets(str); //input string

int i;

for(i=0;str[i]!='\0';i++)

{

if(str[i]>='a'&& str[i]<='z')

str[i]=str[i]-32; //subtract 32 value to get uppercase

}

printf("%s",str);

return 0;

}

//TO CONVERT INTO LOWER CASE

#include<stdio.h>

#include<string.h>

int main()

{

char str[20];

printf("enter string\n");

gets(str); //input string

int i;

for(i=0;str[i]!='\0';i++)

{

if(str[i]>='A'&& str[i]<='Z')

str[i]=str[i]+32; //Add 32 value to get lowercase

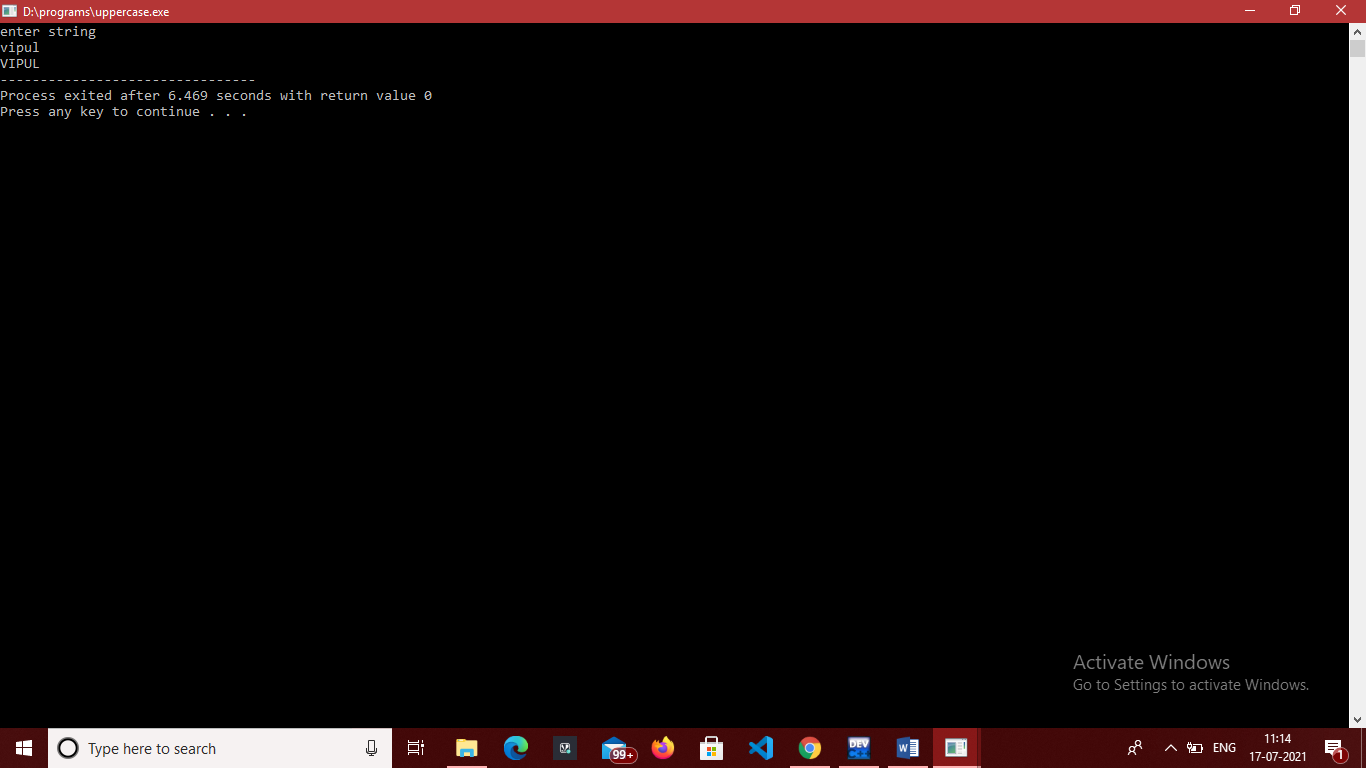
}

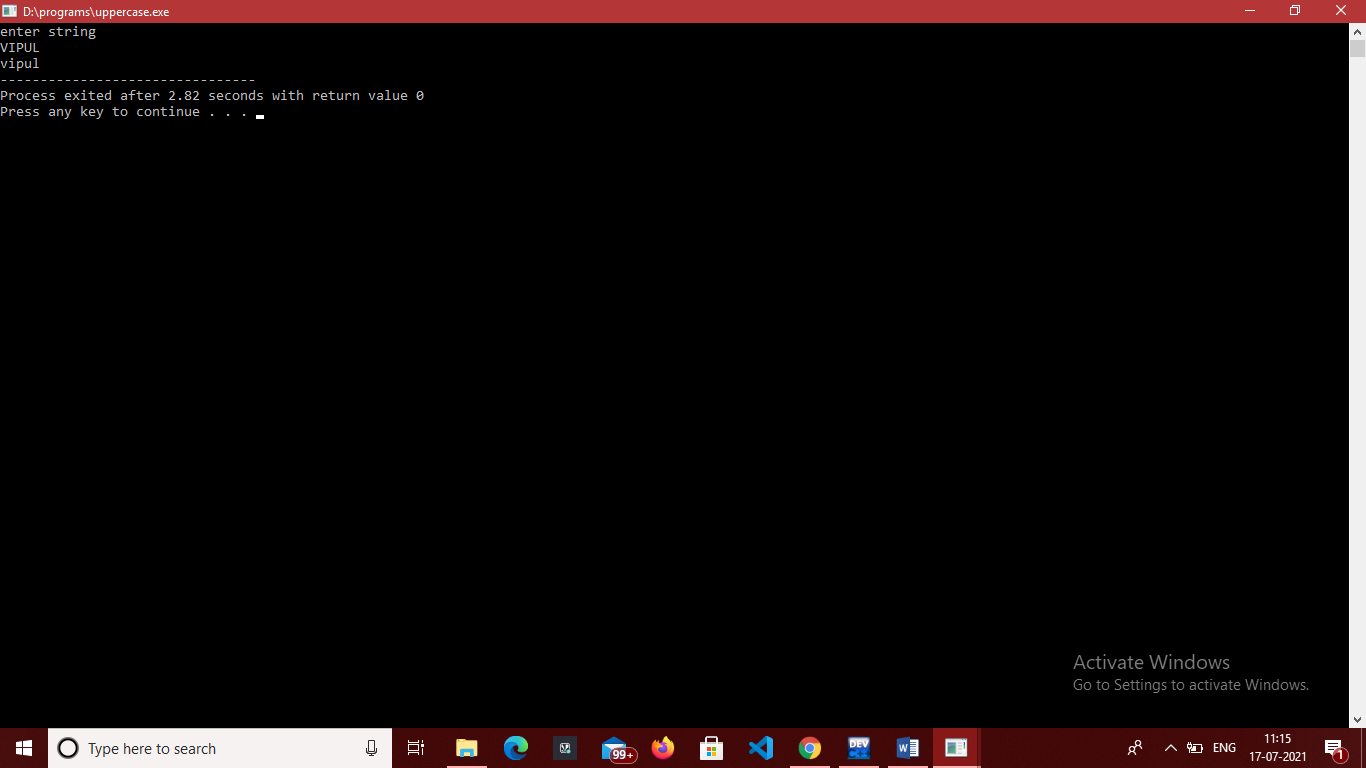
printf("%s",str);

return 0;

}

**OUTPUT-**

****

****

**EXPERIMENT 25.**

**OBJECTIVE-** Program for the addition of two 3 x 3 matrices.

**INTRODUCTION-** In This Program we input two matrices a[i][j] ,b[i][j] and add both matrices into c[i][j].

**PROGRAM CODE-**

#include<stdio.h>

int main()

{

int a[10][10], b[10][10], c[10][10], i, j, k;

printf("\nEnter First Matrix : \n"); //input first matrix

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

{

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

{

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

}

}

printf("\nEnter Second Matrix:\n"); //input second matrix

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

{

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

{

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

}

}

printf("The First Matrix is: \n"); //print first matrix

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

{

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

{

printf(" %d ", a[i][j]);

}

printf("\n");

}

printf("The Second Matrix is : \n"); //print second matrix

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

{

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

{

printf(" %d ", b[i][j]);

}

printf("\n");

}

for (i = 0; i <= 2; i++) //loop for adding

{

for (j = 0; j <= 2; j++)

{

c[i][j]=0;

for (k = 0; k <= 2; k++)

{

c[i][j] = a[i][j] + b[i][j];

}

}

}

printf("\nAddition Of Two Matrices : \n");

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

{

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

{

printf(" %d ", c[i][j]);

}

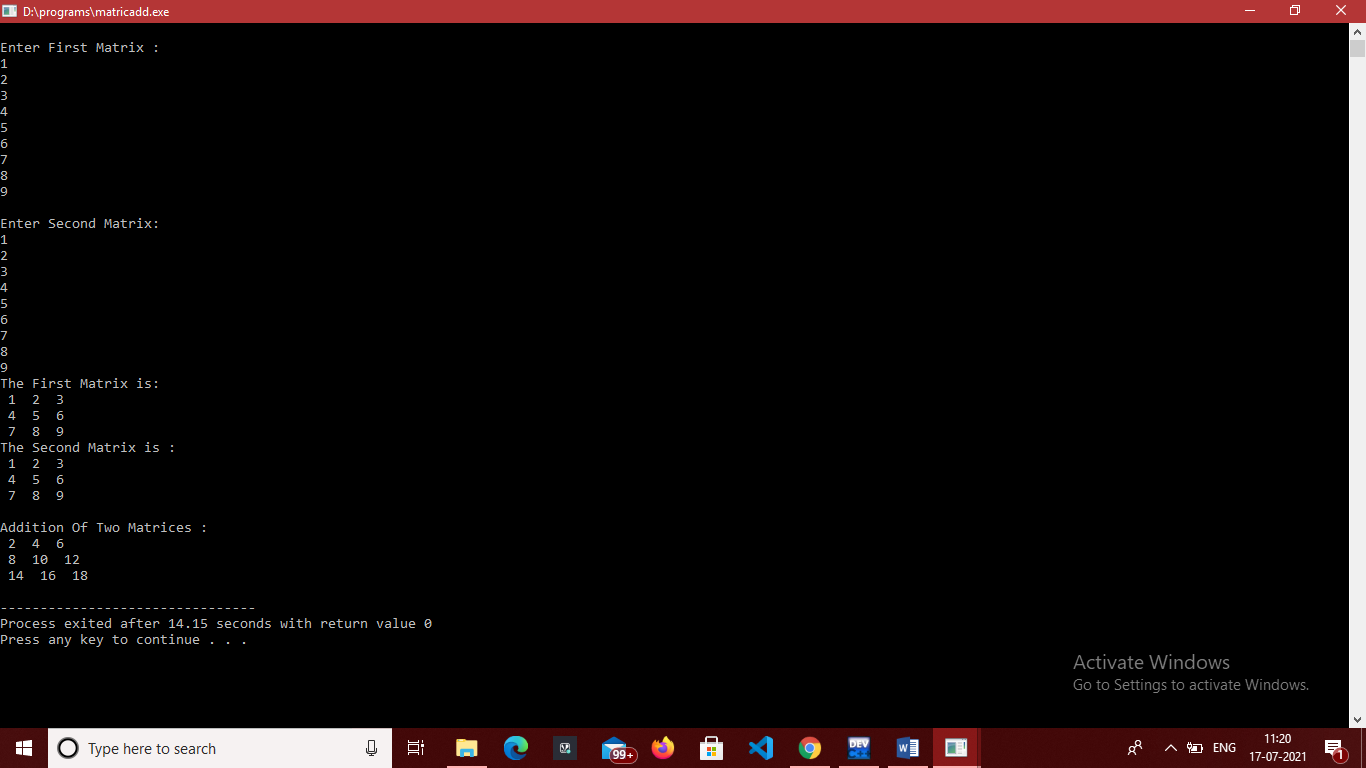
printf("\n");

}

return 0;

}

**OUTPUT-**



**EXPERIMENT 26.**

**OBJECTIVE-** Program to multiply two 3 x 3 matrices

**INTRODUCTION-** In This Program we multiply two matrices a[i][j] , b[i][j] and declare sum equals to 0 and sum in loop condition with c[i][j].

**PROGRAM CODE-**

#include<stdio.h>

int main() {

int a[10][10], b[10][10], c[10][10], i, j, k;

int sum = 0;

printf("\nEnter First Matrix : \n"); //input first matrix

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

for (j = 0; j < 3; j++) {

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

}

}

printf("\nEnter Second Matrix: \n"); //input second matrix

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

for (j = 0; j < 3; j++) {

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

}

}

printf("The First Matrix is: \n"); //print first matrix

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

for (j = 0; j < 3; j++) {

printf(" %d ", a[i][j]);

}

printf("\n");

}

printf("The Second Matrix is : \n"); //print second matrix

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

for (j = 0; j < 3; j++) {

printf(" %d ", b[i][j]);

}

printf("\n");

}

//Multiplication Logic

for (i = 0; i <= 2; i++) {

for (j = 0; j <= 2; j++) {

sum = 0;

for (k = 0; k <= 2; k++) {

sum = sum + a[i][k] \* b[k][j];

}

c[i][j] = sum;

}

}

printf("\nMultiplication Of Two Matrices : \n");

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

for (j = 0; j < 3; j++) {

printf(" %d ", c[i][j]);

}

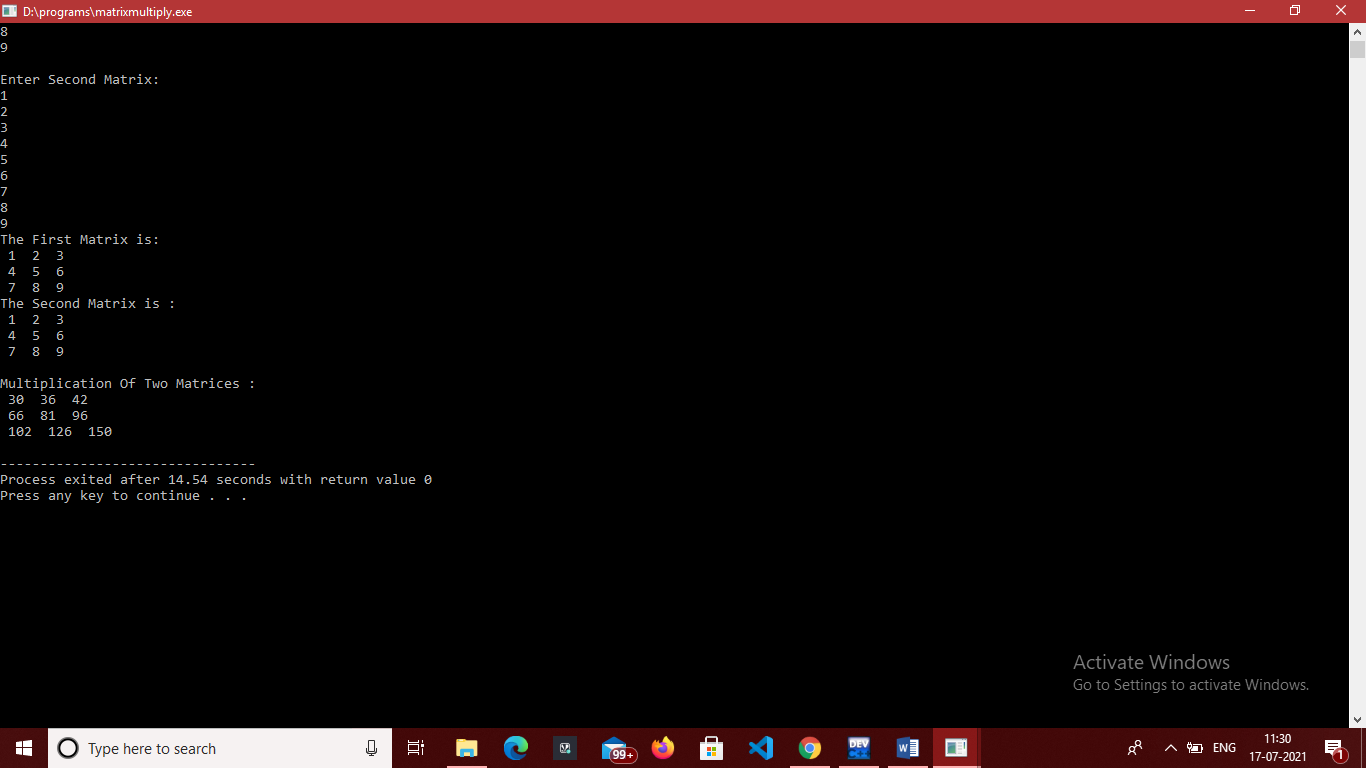
printf("\n");

}

return (0);

}

**OUTPUT-**



**EXPERIMENT 27.**

**OBJECTIVE-** Program to swap two numbers using pointers.

**INTRODUCTION-** In This Program we swap two numbers using pointer we take a function named swap and with the help of temporary variable we swap the numbers.

**PROGRAM CODE-**

// SWAP USING POINTERS

#include<stdio.h>

int swap(int \*x,int \*y) //function for swapping

{

int temp; //declare temporary variable

temp=\*x;

\*x=\*y;

\*y=temp;

}

int main()

{

int num1,num2;

printf("enter first number: \n"); // input number 1

scanf("%d",&num1);

printf("enter second number: \n"); //input number 1

scanf("%d",&num2);

printf("Numbers before swapping: num1 is %d , num2 is %d \n",num1,num2);

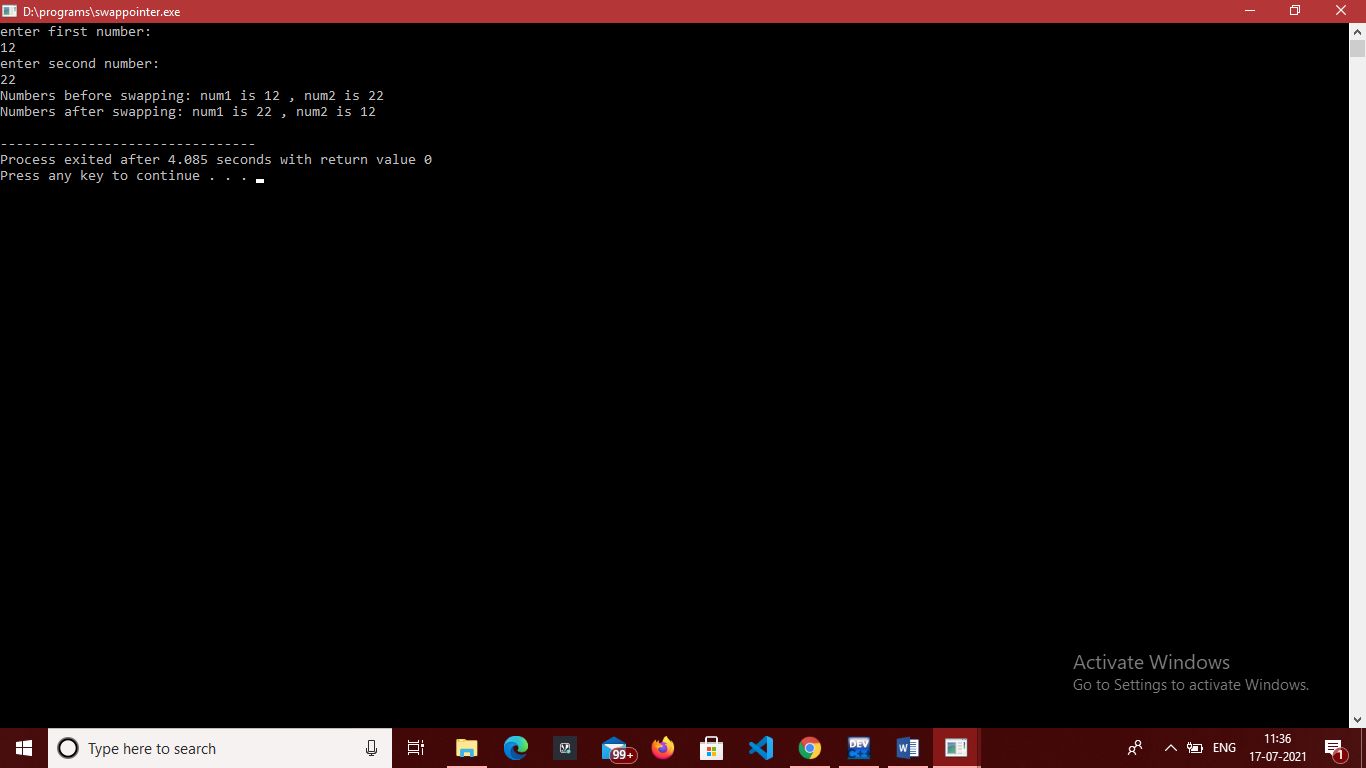
swap(&num1,&num2);

printf("Numbers after swapping: num1 is %d , num2 is %d \n",num1,num2);

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 28.**

**OBJECTIVE-** Program to generate the employee details using structure.

**INTRODUCTION-** In This Program we make a structure of employee details and print it.

**PROGRAM CODE-**

// EMPLOYEE DETAILS USING STRUCTURE

#include<stdio.h>

struct employee

{

int id;

char name[20];

float salary;

int age;

};

int main()

{ struct employee e1; /\*declare structure variable\*/

int i,n;

/\*read employee details\*/

printf("\nEnter details :\n");

printf("Name ?: ");

gets(e1.name);

printf("ID ?: ");

scanf("%d",&e1.id);

printf("Salary ?: ");

scanf("%f",&e1.salary);

printf("Age ?: ");

scanf("%d",&e1.age);

/\*print employee details\*/

printf("\nEntered detail is: \n");

printf("Name: %s \n",e1.name);

printf("Id: %d \n",e1.id);

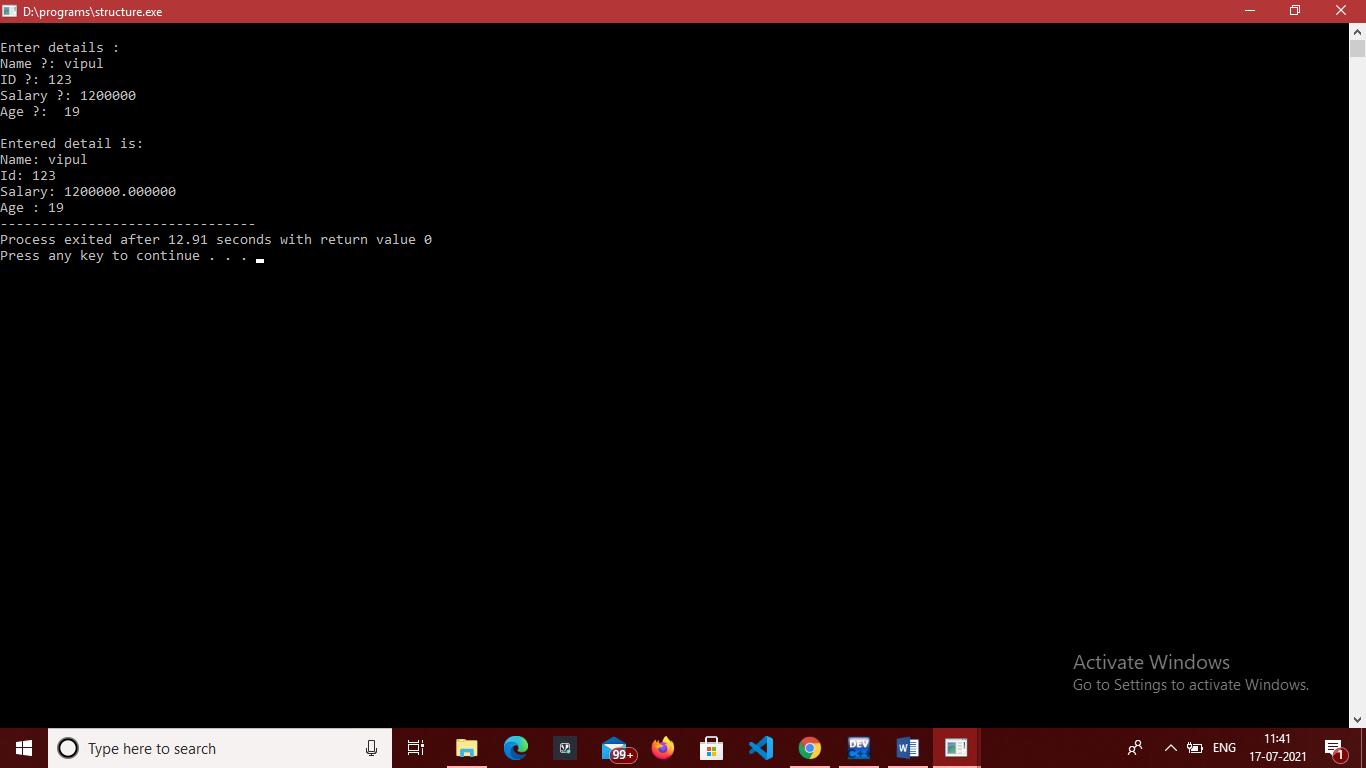
printf("Salary: %f\n",e1.salary);

printf("Age : %d",e1.age);

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 29.**

**OBJECTIVE-** Program to find the area and perimeter of a circle, rectangle, square and triangle using functions.

**INTRODUCTION-** In This Program we calculate area and perimeter of circle , rectangle, square ,triangle by using their formulas in functions.

**PROGRAM CODE-**

#include<stdio.h>

#include<process.h>

#include<math.h>

//function to calculate area of rectangle

float recarea(float a, float b) {

float area;

area = a \* b;

return area;

}

//function to calculate perimeter of rectangle

float recperimeter(float a, float b){

float perimeter;

perimeter = 2\*(a + b);

return perimeter;

}

float circlearea(float rad)

{

float area;

area=3.14\*rad\*rad;

return area;

}

float circleperi(float rad)

{ float perm;

perm=2\*3.14\*rad;

return perm;

}

float triArea(float a, float b, float c)

{

// Length of sides must be positive

// and sum of any two sides

// must be smaller than third side.

if (a < 0 || b < 0 || c < 0 ||(a + b <= c) || a + c <= b ||b + c <= a)

{

printf("Not a valid triangle");

exit(0);

}

float s = (a + b + c) / 2;

return sqrt(s \* (s - a) \*(s - b) \* (s - c));

}

float perimtri(float a,float b,float c)

{

float sum;

sum=a+b+c;

return sum;

}

float areasq(float a)

{

float area;

area=a\*a;

return area;

}

float perisq(float a)

{

float peri;

peri= 4\*a;

return peri;

}

int main()

{ int choice;

float r;

float a,b,s,c;

printf("Calculate Area and Perimeter of :\n");

printf("1. Circle: \n");

printf("2. Rectangle: \n");

printf("3. Square: \n");

printf("4. Triangle: \n");

printf("Enter choice: \n");

scanf("%d",&choice);

switch(choice)

{

case 1: printf("enter radius: ");

scanf("%f",&r);

printf("Area of circle is: %f \n", circlearea(r));

printf("Perimeter of circle is: %f \n ",circleperi(r));

break;

case 2: printf("Enter sides\n");

scanf("%f %f",&a,&b);

printf("Area of Rectangle is: %f \n ", recarea(a,b));

printf("Perimeter of rectangle is: %f \n",recperimeter(a,b));

break;

case 3: printf("enter side\n");

scanf("%f",&s);

printf("Area of square is: %f \n ",areasq(s));

printf("Perimeter of square is: %f \n",perisq(s));

break;

case 4: printf("enter sides: \n");

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

printf("Area of triangle is: %f \n",triArea(a,b,c));

printf("Perimeter of triangle is: %d \n",perimtri(a,b,c));

perimtri(a,b,c);

break;

default: printf("wrong input");

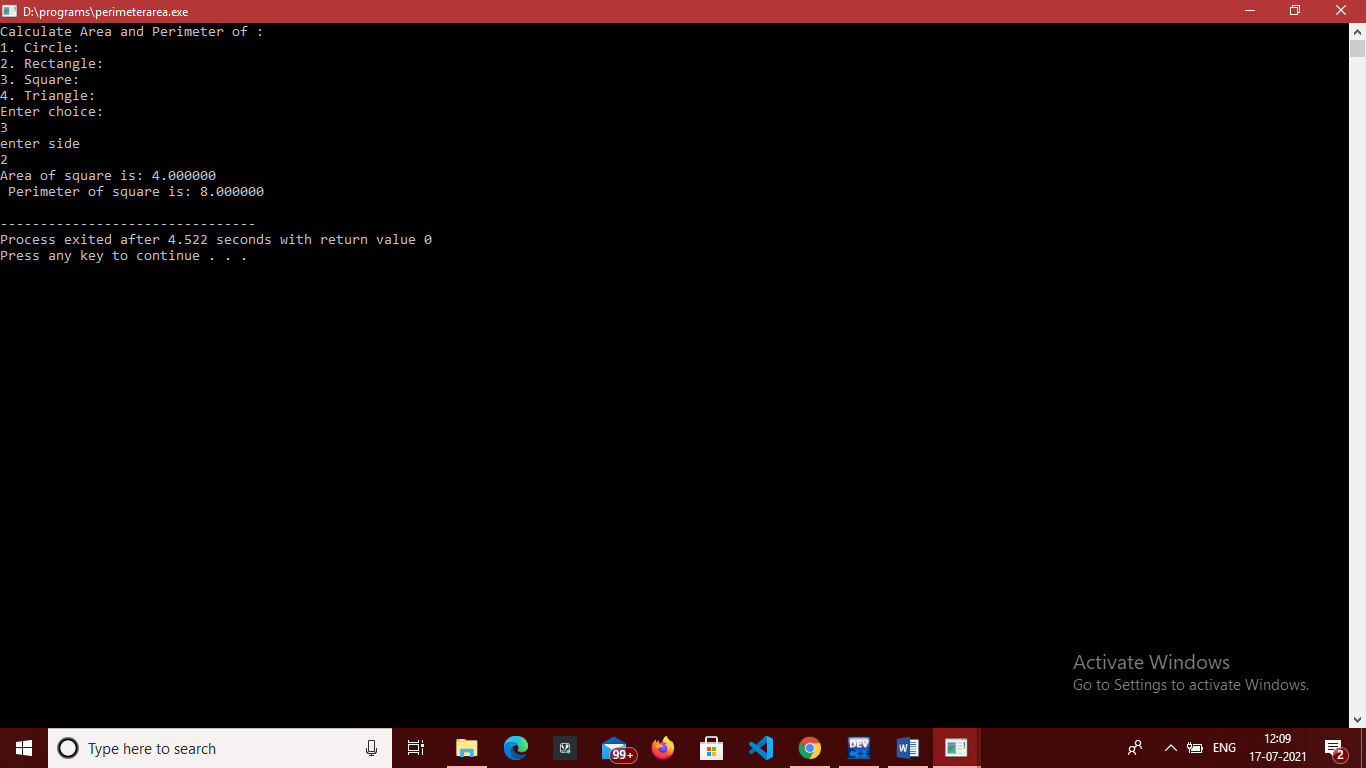
break;

}

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 30.**

**OBJECTIVE-** Program to pass and return pointer to function hence calculate average of an array.

**INTRODUCTION-** In This Program we calculate average by returning the value stores on &avg.

**PROGRAM CODE-**

#include <stdio.h>

float \*average(int \*a,int n) //function to calculate average

{

int i;

float avg=0;

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

{

avg +=\*a;

a++;

}

//printf("sum is %d \n" ,sum);

avg=avg/n;

printf("average is %f",avg);

return &avg;

}

int main()

{

int a[20],n,i;

printf("enter number of elements: \n"); //size of array you want

scanf("%d",&n);

for(i=0;i<n;i++){

printf("enter %d element :",i+1);

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

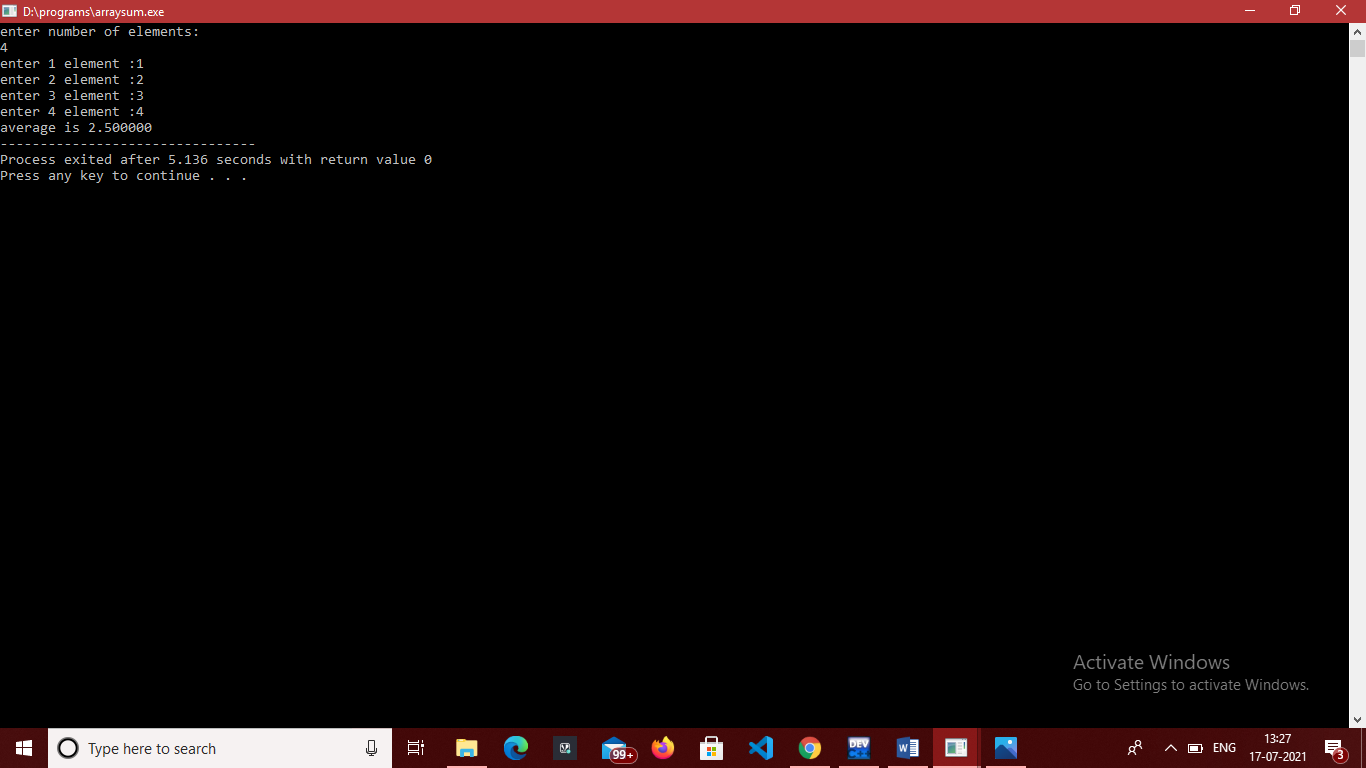
}

float \*arr=average(a,n); //fuction call

return 0;

}

**OUTPUT-**



**EXPERIMENT 31.**

**OBJECTIVE-** Program to pass an array as pointer to a function that calculates the sum of all elements of the array

**INTRODUCTION-** In This Program we calculate sum by array as pointer in it. We make function name sum to calculate sum and then it in main function.

**PROGRAM CODE-**

#include <stdio.h>

float \*sum(int \*a,int n) //function to sum average

{

int sum=0, i;

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

{

sum +=\*a;

a++;

}

printf("sum is %d \n" ,sum);

}

int main()

{

int a[20],n,i;

printf("enter number of elements: \n"); //size of array you want

scanf("%d",&n);

for(i=0;i<n;i++){

printf("enter %d element :",i+1);

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

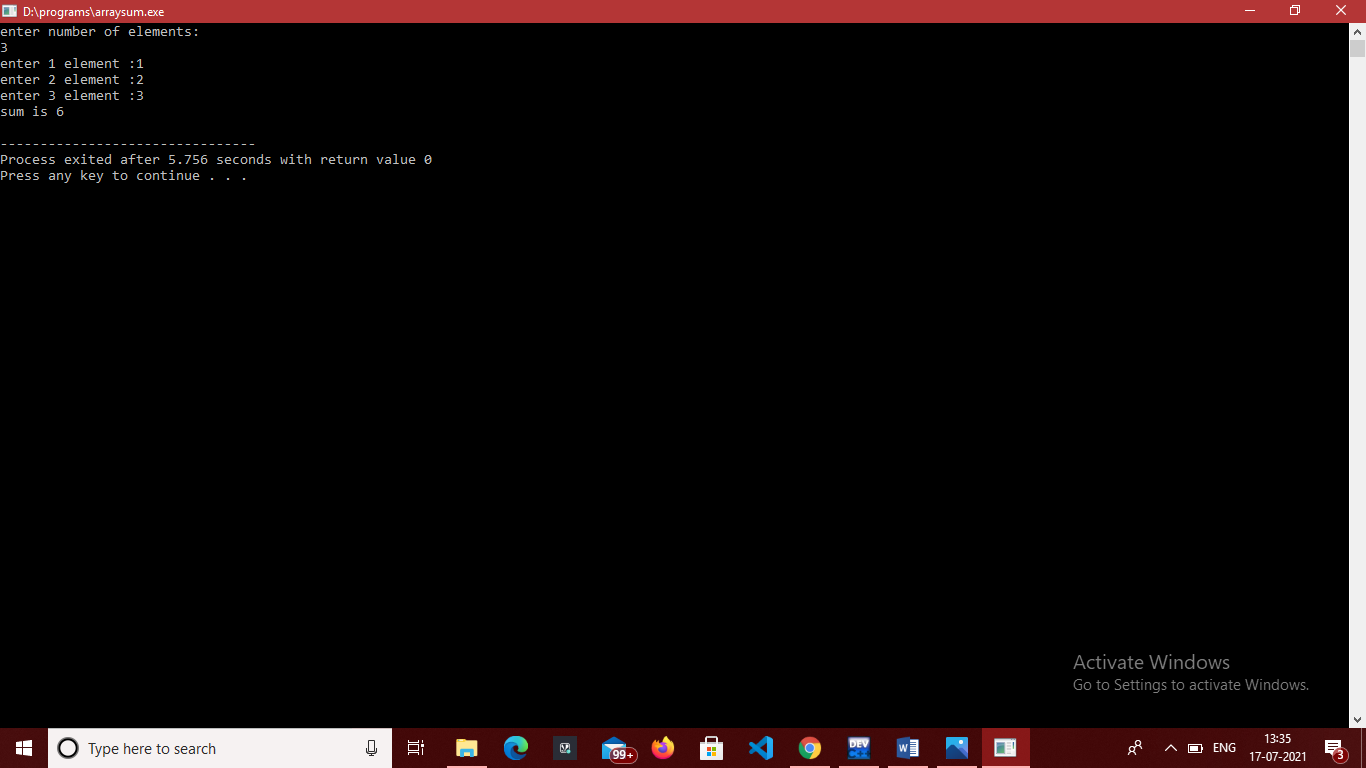
}

float \*arr=sum(a,n); //fuction call

return 0;

}

**OUTPUT-**

****

**EXPERIMENT 32.**

**OBJECTIVE-** Program to demonstrate the example of array of pointers

**INTRODUCTION-** In This Program we make array of pointer and stores values in it and display them.

**PROGRAM CODE-**

#include<stdio.h>

int main()

{

int a,b,c;

int \*ptr[3];

ptr[0]= &a;

ptr[1]= &b;

ptr[2]= &c;

a=100;

b=200;

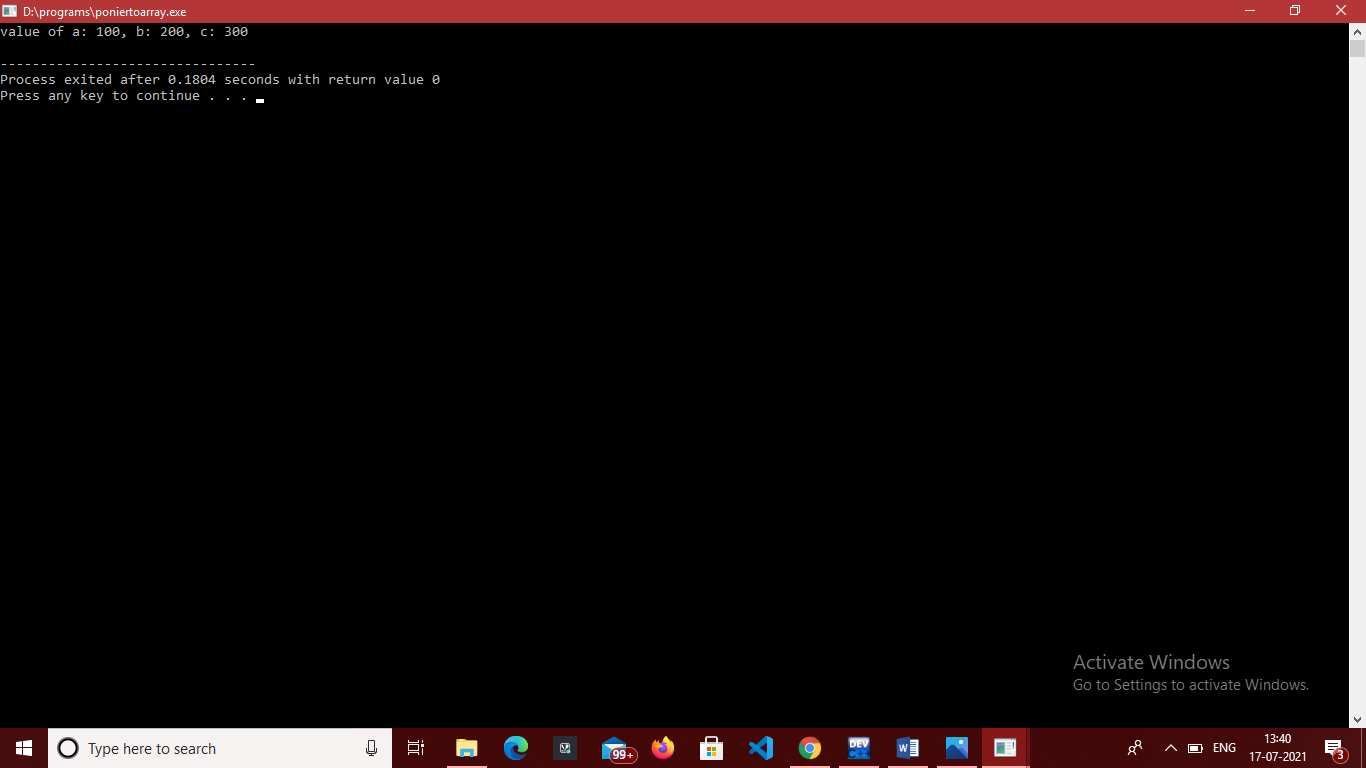
c=300;

printf("value of a: %d, b: %d, c: %d\n",\*ptr[0],\*ptr[1],\*ptr[2]);

return 0;

}

**OUTPUT-**

****