**EXPERIMENT** – **1**

**AIM** -To check whether number is prime or not.

**PROGRAM** -

#include <iostream>

using namespace std;

int main()

{int num,i,count=0;

cout<<"Enter a number:";

cin>>num;

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

{if(num%i==0)

{count++;

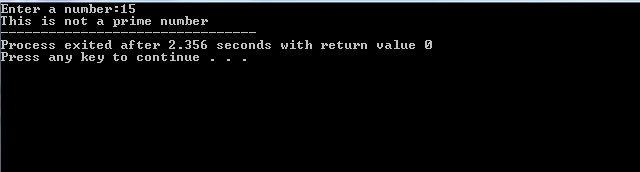
break;}}

if(count==0)

{cout<<"This is a prime number";}

else{cout<<"This is not a prime number";}}

**OUTPUT** –



**EXPERIMENT – 2**

**AIM –** Find greatest of three numbers.

**PROGRAM** -

#include <iostream>

using namespace std;

int main()

{ int a,b,c;

cout<<"Enter the three numbers: "<<endl;

cin>>a>>b>>c;

if(a>=b && a>=c)

{cout<<"\n"<<a<<" is the greatest"<<endl;}

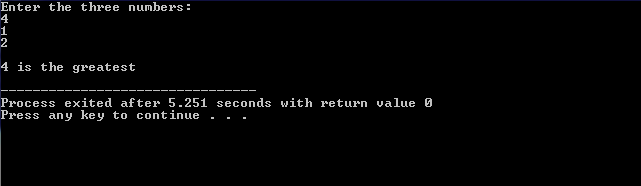
else if(b>=c)

{ cout<<b<<" is the greatest number"<<endl;}

else{ cout<<c<<" is the greatest number"<<endl;}

}

**OUTPUT-**



**EXPERIMENT – 3**

**AIM-**Swap two numbers without using third variable.

**PROGRAM** -

#include <iostream>

using namespace std;

int main()

{

int a,b;

cout<<"Enter two numbers: "<<endl;

cin>>a;

cin>>b;

a=a+b;

b=a-b;

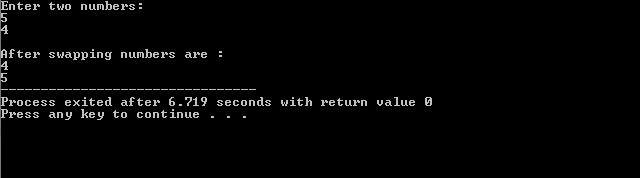
a=a-b;

cout<<"\nAfter swapping numbers are : ";

cout<<"\n"<<a<<"\n"<<b;

}

**OUTPUT-**



**EXPERIMENT – 4**

**AIM­-**Calculate factorial of a number.

**PROGRAM** -

#include <iostream>

using namespace std;

int main()

{int num, i, fact=1;

cout<<"Enter a number : ";

cin>>num;

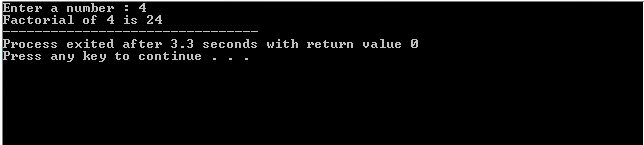
for(i=num; i>0; i--)

{fact=fact\*i;}

cout<<"Factorial of "<<num<<" is "<<fact;

}

**OUTPUT-**



**EXPERIMENT-5**

**AIM-**To find factors of a number.

**PROGRAM** -

#include <iostream>

using namespace std;

int main()

{

int n, i;

cout << "Enter a positive integer: ";

cin >> n;

cout << "Factors of " << n << " are: " << endl;

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

{

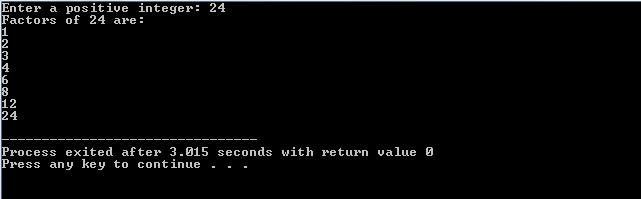
if(n % i == 0)

cout << i << endl;

}

}

**OUTPUT-**



**EXPERIMENT – 6**

**AIM-**To reverse elements of an array.

**PROGRAM** -

#include<iostream>

using namespace std;

int main(){

int Arr[100],n,temp,i,j;

cout<<"Enter number of elements you want to insert ";

cin>>n;

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

cout<<"Enter element "<<i+1<<":";

cin>>Arr[i];}

for(i=0,j=n-1;i<n/2;i++,j--){

temp=Arr[i];

Arr[i]=Arr[j];

Arr[j]=temp;}

cout<<"\nReverse array"<<endl;

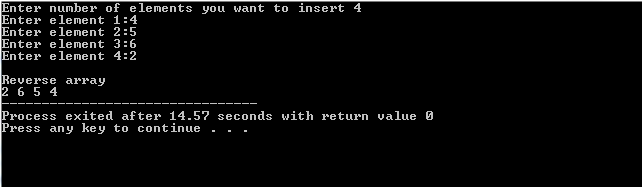
for(i=0;i<n;i++)

cout<<Arr[i]<<" ";

return 0;

}

**OUTPUT-**



**EXPERIMENT -7**

**AIM -** Check whether the given number is palindrome or not.

**PROGRAM** -

#include<iostream>

using namespace std;

int main(){

int num, rem, orig\_no, rev\_no=0;

cout<<"Enter a number : ";

cin>>num;

orig\_no=num;

while(num!=0)

{rem=num%10;

rev\_no=rev\_no\*10 + rem;

num=num/10;}

if(rev\_no==orig\_no)

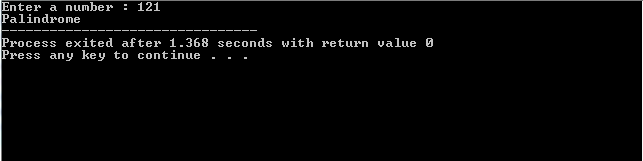
{cout<<"Palindrome";}

else

{cout<<"Not Palindrome";}

}

**OUTPUT-**



**EXPERIMENT** – **8**

**AIM** -To find the GCD of given numbers.

**PROGRAM** -

#include<iostream>

using namespace std;

int main() {

int first;

cout<<"Enter First Number : ";

cin>>first;

int second;

cout<<"Enter Second Number: ";cin>>second;

int gcd;

for(int i=1;i<=first&&i<=second;i++){

if(first%i==0 && second%i == 0 ){

gcd=i

}

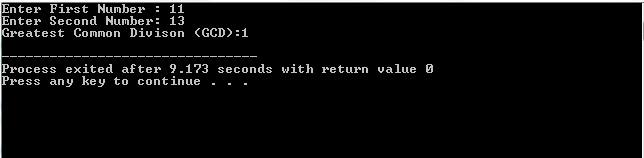
}

cout<<"Greatest Common Divison (GCD):"<<gcd<<endl;

return 0;

}

**OUTPUT** –



**EXPERIMENT 9**

**Aim:** To find the median of the given array

**Program:**

#include<iostream>

using namespace std;

int main()

{

int n,a[10];

cout<<"\nEnter the number of element ";

cin>>n;

cout<<"\nEnter the elements ";

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

{

cin>>a[i];

}

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

{

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

{

if(a[i]>a[j])

{

int temp = a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

int t;

if(n%2==0)

{

t=(n/2)-1;

}

else

{

t=n/2;

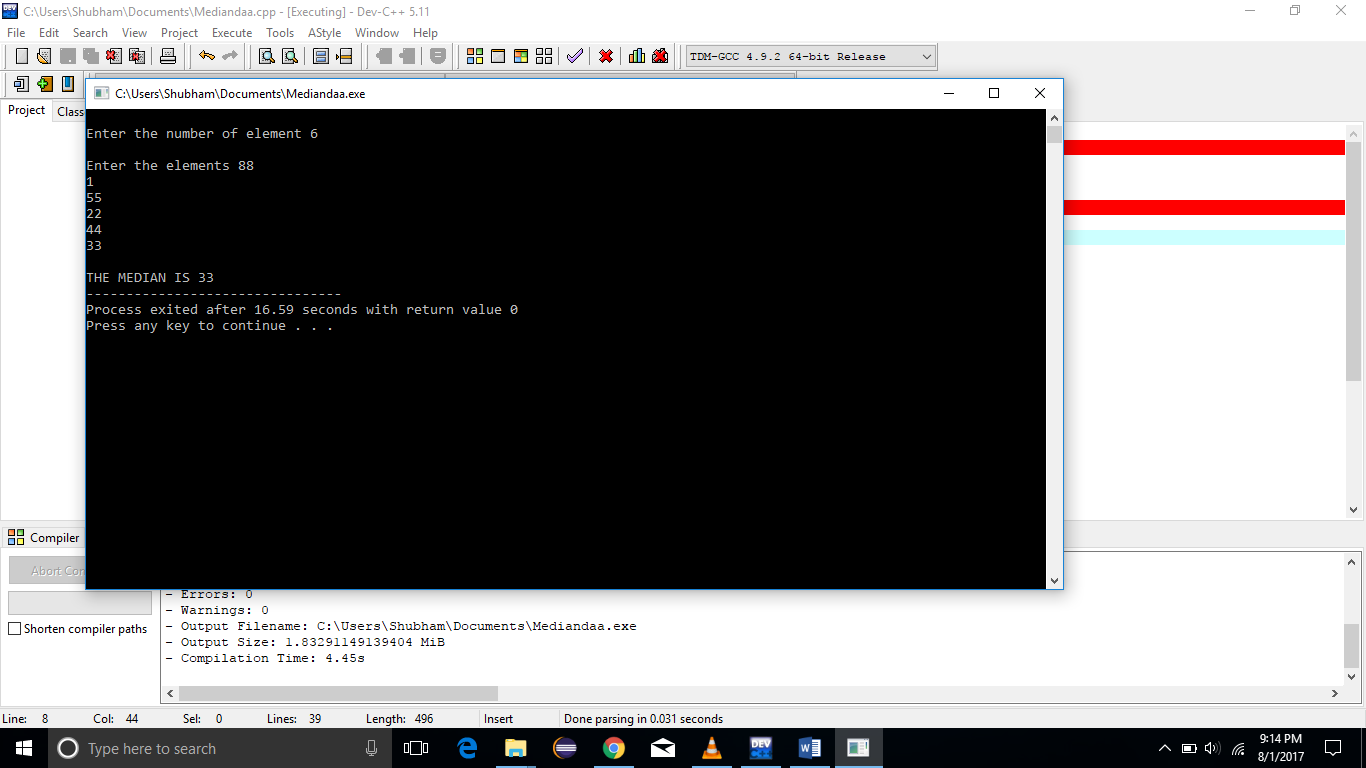
}

cout<<"\nTHE MEDIAN IS "<<a[t];

return 0;

}

**OUTPUT:**



**EXPERIMENT**–**13**

**AIM**– To implement binary search using divide and conquer strategy.

**CODE**–

#include<iostream>

using namespace std;

int main(){

int n, i, arr[50], search, first, last, middle;

cout<<"Enter total number of elements :";

cin>>n;

cout<<"Enter "<<n<<" number :\n";

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

cin>>arr[i];

cout<<"Enter a number to find :";

cin>>search;

first = 0;

last = n-1;

middle = (first+last)/2;

while (first <= last){

if(arr[middle] < search){

first = middle + 1;}

else if(arr[middle] == search)

{

cout<<"The number found at location :"<<middle+1<<"\n";

break;

}

else

{

last = middle - 1;

}

middle=first+last/2;

}

if(first > last)

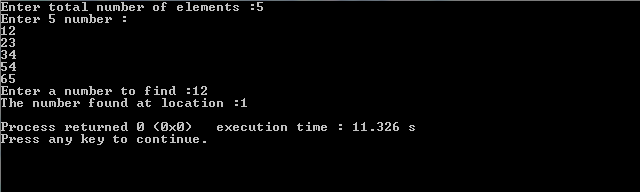
{

cout<<"Not found! the number is not present in the list.";

}

}

**OUTPUT** –



**EXPERIMENT**–**14**

**AIM**– To find the LCS (Longest Common Subsequence) by using dynamic programming.

**CODE**–

public class LCS{

int lcs( char[] X, char[] Y, int m, int n ) {

if (m == 0 || n == 0)

return 0;

if (X[m-1] == Y[n-1])

return 1 + lcs(X, Y, m-1, n-1);

else

return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));

}

int max(int a, int b){

return (a > b)? a : b;

}

public static void main(String[] args){

LCS lcs = new LCS();

String s1 = "AGGTABSADLKS";

String s2 = "GXTXAYBGFJKA";

char[] X=s1.toCharArray();

char[] Y=s2.toCharArray();

int m = X.length;

int n = Y.length;

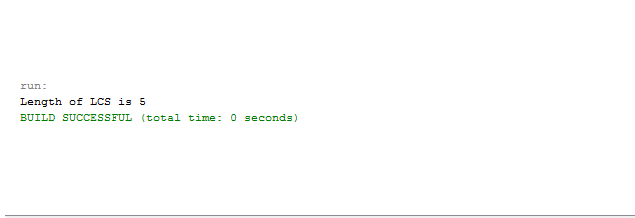
System.out.println("Length of LCS is" + " " +

lcs.lcs( X, Y, m, n ) );

}

}

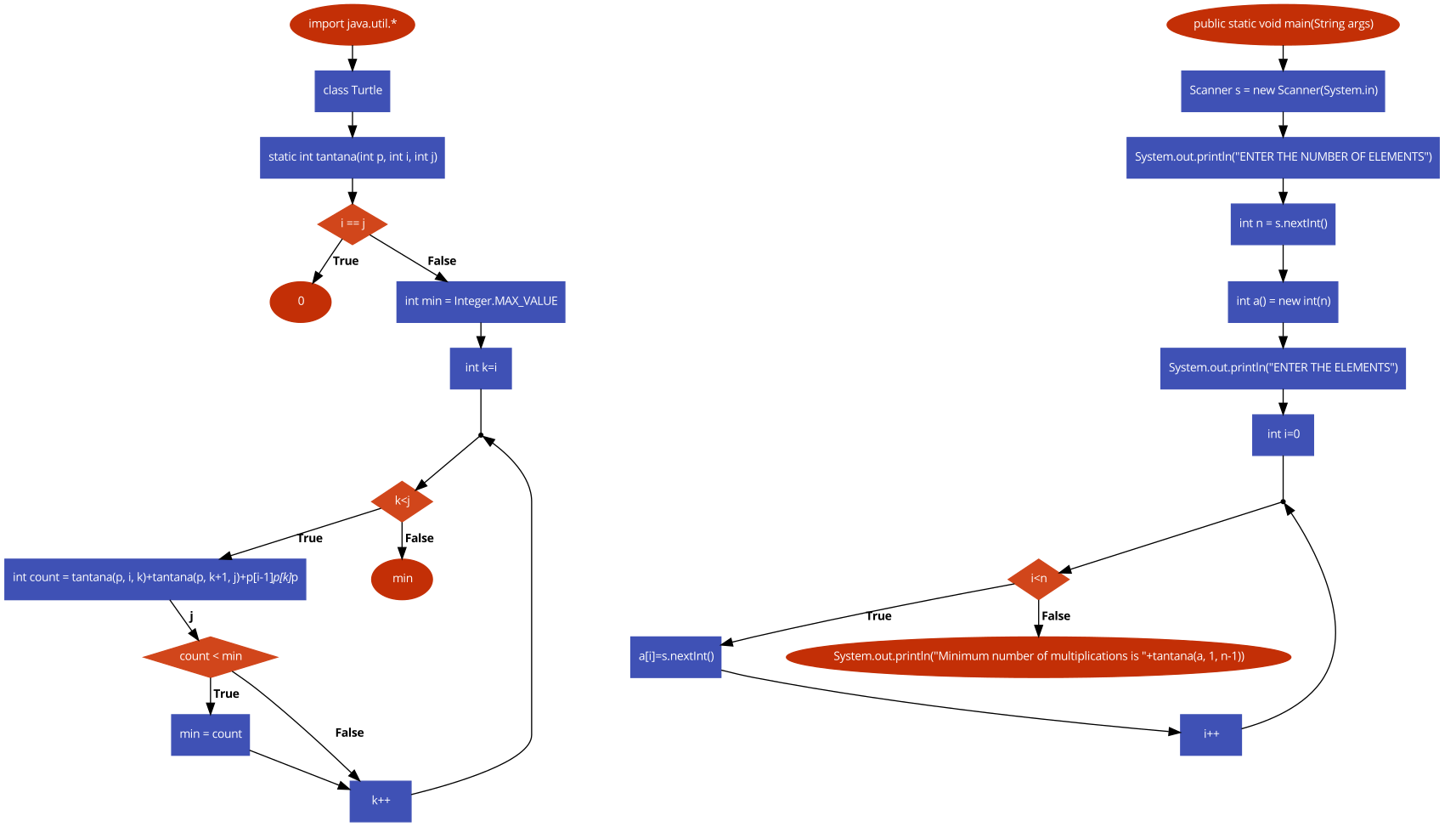
**OUTPUT** –



**EXPERIMENT NO: -15**

**AIM: -** To implement matrix chain multiplication.

**FLOW CHART: -**

****

**CODE:-**

import java.util.\*;

class Turtle

{

static int tantana(int p[], int i, int j)

{

if (i == j)

return 0;

int min = Integer.MAX\_VALUE;

for (int k=i; k<j; k++)

{

int count = tantana(p, i, k)+tantana(p, k+1, j)+p[i-1]\*p[k]\*p[j];

if (count < min)

min = count;

}

return min;

}

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

System.out.println("ENTER THE NUMBER OF ELEMENTS");

int n = s.nextInt();

int a[] = new int[n];

System.out.println("ENTER THE ELEMENTS");

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

{

a[i]=s.nextInt();

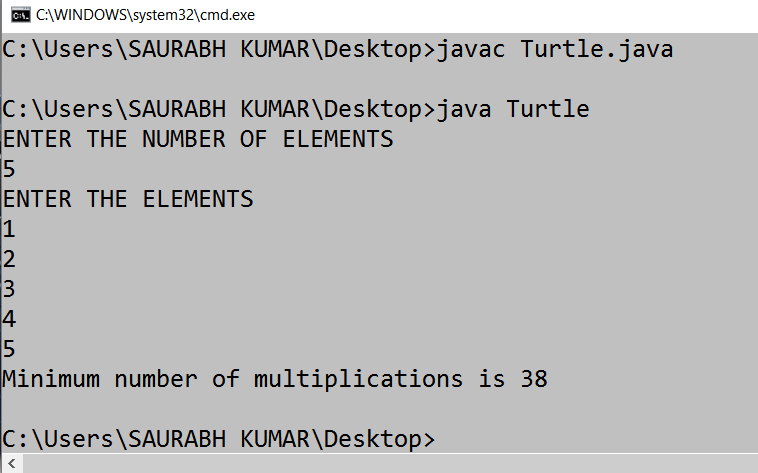
}

System.out.println("Minimum number of multiplications is "+tantana(a, 1, n-1));

}

}

**OUTPUT:-**

****