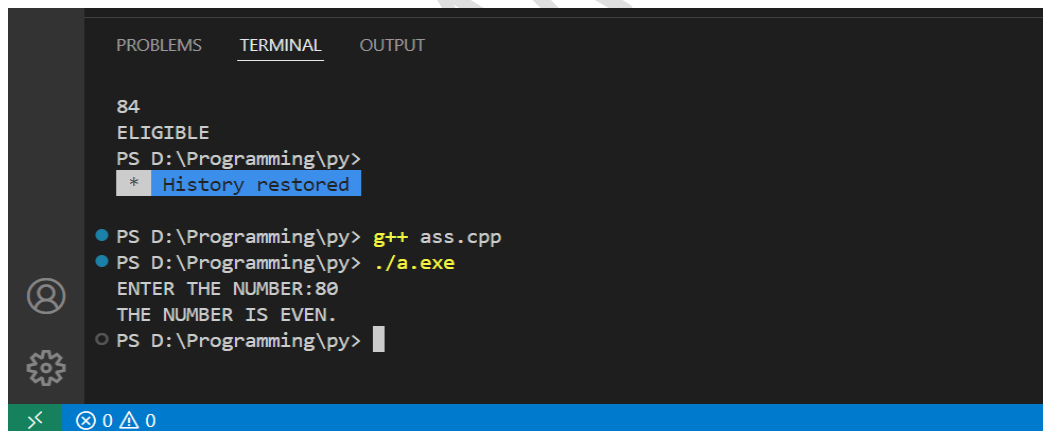


**QUE 1:** WAP to find given number is even or odd.

**SOLUTION:**

```
// to find if the number is even or odd
#include<iostream>
using namespace std;
int main ( )
{
    int n;
    cout<<" ENTER THE NUMBER: ";
    cin>n;
    if (n%2==0)
        cout<<" THE NUMBER IS EVEN.";
    else
        cout<<" THE NUMBER IS ODD.";
    return 0;
}
```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT
84
ELIGIBLE
PS D:\Programming\py>
* History restored
• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
ENTER THE NUMBER:80
THE NUMBER IS EVEN.
○ PS D:\Programming\py> |
```

**QUE 2:**

**SOLUTION:**

```
//to find out if the given number is prime or composite.
#include <iostream>
using namespace std;
int main ( )
{
```

```

int n;

cout<< "ENTER THE NUMBER": ;

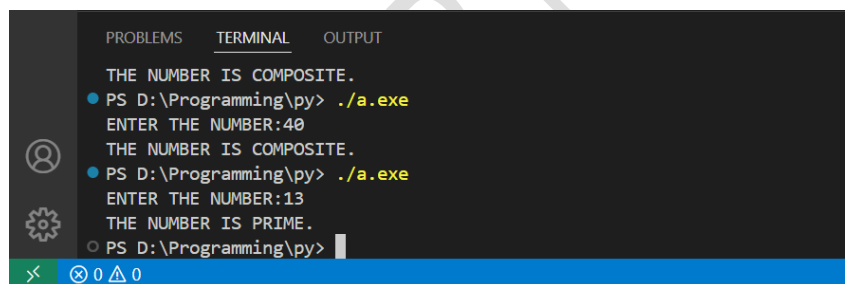
cin>>n;

for (int i = 2; i <= n; i++)
{
    if(n % i != 0)
        continue;
    else if ( n % i == 0 && n == i)
        cout<< "THE NUMBER IS PRIME";
    else
        cout<< "THE NUMBER IS COMPOSITE";
    break;
}

return 0;
}

```

#### OUTPUT:



```

PROBLEMS  TERMINAL  OUTPUT
THE NUMBER IS COMPOSITE.
• PS D:\Programming\py> ./a.exe
ENTER THE NUMBER:40
THE NUMBER IS COMPOSITE.
• PS D:\Programming\py> ./a.exe
ENTER THE NUMBER:13
THE NUMBER IS PRIME.
○ PS D:\Programming\py>

```

**QUE 3:**WAP to print the table of a given number up to 'n' multiples.

#### SOLUTION:

```

// to display the table of a given number
#include <iostream>
using namespace std;
int main ( )
{
    int n, num;

    cout << "enter the number";

    cin >> num;

    cout << "enter the value of the last multiple: ";

    cin >> n;
}

```

```

        for( int i = 1; i <= n; i++)
        {
            int a = num*i;

            cout <<num << "x" <<i<< "=" <<a<<endl;

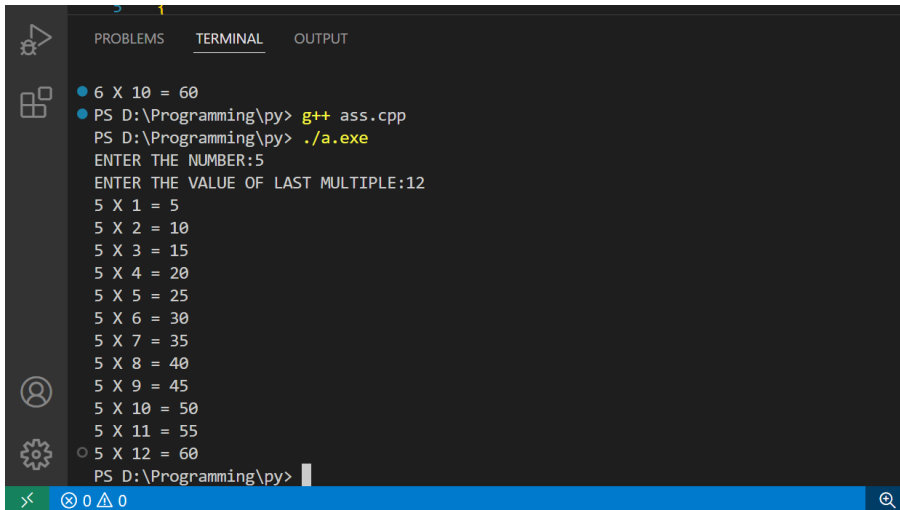
        }

        return 0;

    }

```

### OUTPUT:



```

6 X 10 = 60
PS D:\Programming\py> g++ ass.cpp
PS D:\Programming\py> ./a.exe
ENTER THE NUMBER:5
ENTER THE VALUE OF LAST MULTIPLE:12
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
5 X 11 = 55
5 X 12 = 60
PS D:\Programming\py>

```

**QUE 4:** WAP to find the greater number between:

- (a) two numbers
- (b) three numbers

**SOLUTION:**

```

// to find the greater number between two numbers.
#include<iostream>
using namespace std;
int main ( )
{
    int a1, a2;

    cout << "ENTER THE FIRST NUMBER: ";

    cin <<a1;

    cout << "ENTER THE SECOND NUMBER: ";

    cin <<a2;

    if (a1>a2)

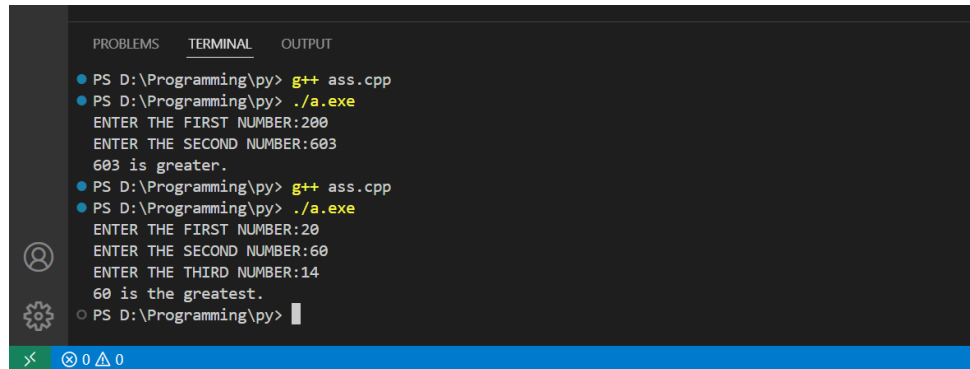
```

```

        cout << a1 << "IS GREATER.";
    else
        cout << a2 << "IS GREATER.";
    return 0;
}

```

## OUTPUT:



```

PROBLEMS  TERMINAL  OUTPUT
• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
ENTER THE FIRST NUMBER:200
ENTER THE SECOND NUMBER:603
603 is greater.
• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
ENTER THE FIRST NUMBER:20
ENTER THE SECOND NUMBER:60
ENTER THE THIRD NUMBER:14
60 is the greatest.
○ PS D:\Programming\py>

```

// to find the greater number between **three** numbers.

```
#include <iostream>
```

```
using namespace std;
```

```
int main ( )
```

```
{
```

```
    int a1, a2, a3;
```

```
    cout << "enter the first number: ";
```

```
    cin >> a1;
```

```
    cout << "enter the second number: ";
```

```
    cin >> a2;
```

```
    cout << "enter the third number: ";
```

```
    cin >> a3;
```

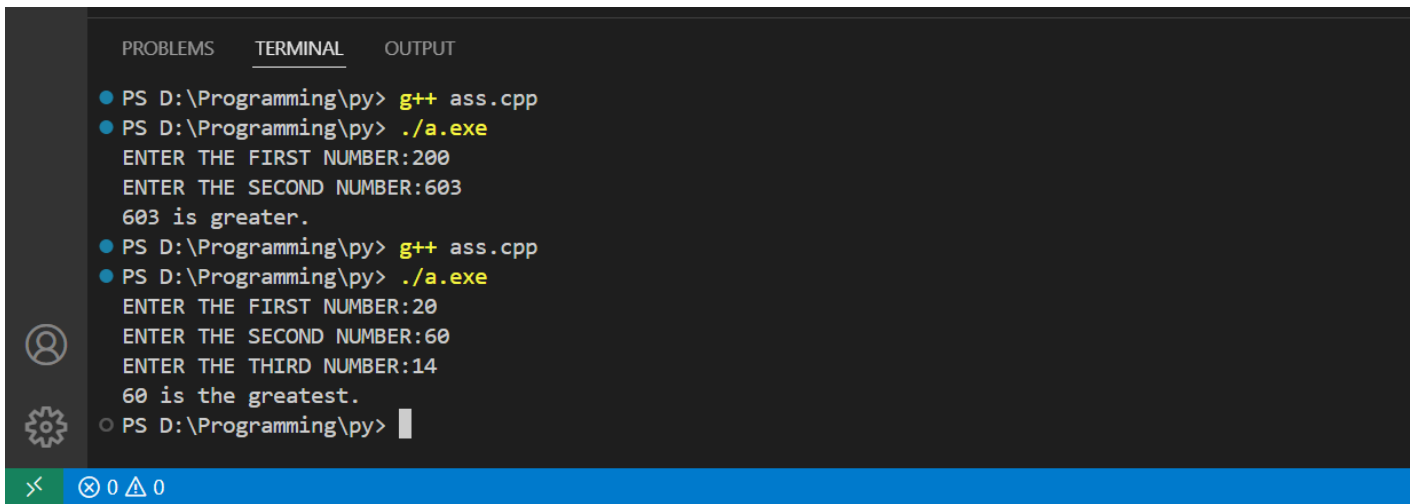
```
    (a1>a2) ? (a1>a3) ? cout << a1 << "is greater." : cout << a3 << "is greater." : (a2>a3) ?
    cout << a2 << "is greater." : cout << a3 << "is greater.";

```

```
    return 0;
```

```
}
```

## OUTPUT:



```
PROBLEMS  TERMINAL  OUTPUT

• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
ENTER THE FIRST NUMBER:200
ENTER THE SECOND NUMBER:603
603 is greater.
• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
ENTER THE FIRST NUMBER:20
ENTER THE SECOND NUMBER:60
ENTER THE THIRD NUMBER:14
60 is the greatest.
• PS D:\Programming\py> 
```

**QUE 5:** WAP to find the sum of first ‘n’ natural numbers.

**SOLUTION:**

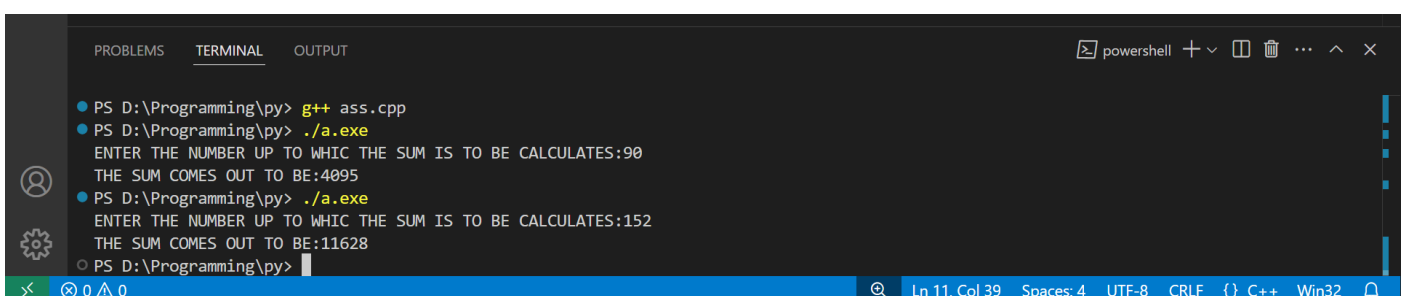
```
// to find sum of 'n' natural numbers.

#include<iostream>

using namespace std;

int main ( )
{
    int n, i, a = 0;
    cout<< "ENTER THE NUMBER UP TO WHICH THE SUM IS TO BE FOUND: ";
    cin >> n;
    for (i = 0; i <= n; i++)
    {
        a = a + i;
    }
    cout<< "THE SUM COMES OUT TO BE: "<< a;
    return 0;
}
```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT

• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
ENTER THE NUMBER UP TO WHICH THE SUM IS TO BE CALCULATES:90
THE SUM COMES OUT TO BE:4095
• PS D:\Programming\py> ./a.exe
ENTER THE NUMBER UP TO WHICH THE SUM IS TO BE CALCULATES:152
THE SUM COMES OUT TO BE:11628
• PS D:\Programming\py> 
```

**QUE 6:** WAP to find factorial of a given number.

**SOLUTION:**

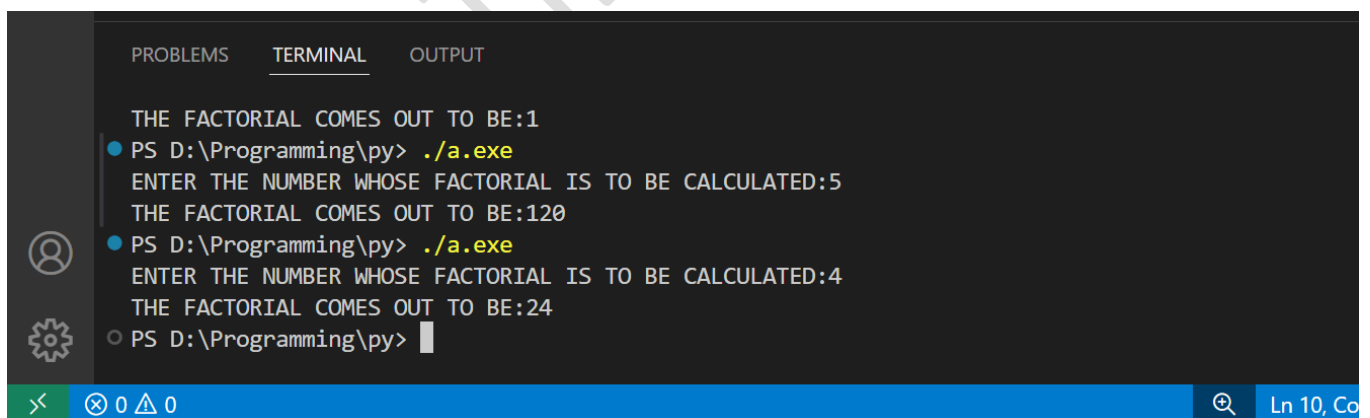
```
// to find the factorial of a given number.

#include<iostream>

using namespace std;

int main ( )
{
    int n, i, a = 1;
    cout<< "ENTER THE NUMBER WHOSE FACTORIAL IS TO BE CALCULATED: ";
    cin>> n;
    for (i = 1; i <= n; i++)
    {
        a = a * i;
    }
    cout<< "THE FACTORIAL COMES OUT TO BE: "<< a;
    return 0;
}
```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT

THE FACTORIAL COMES OUT TO BE:1
● PS D:\Programming\py> ./a.exe
ENTER THE NUMBER WHOSE FACTORIAL IS TO BE CALCULATED:5
THE FACTORIAL COMES OUT TO BE:120
● PS D:\Programming\py> ./a.exe
ENTER THE NUMBER WHOSE FACTORIAL IS TO BE CALCULATED:4
THE FACTORIAL COMES OUT TO BE:24
○ PS D:\Programming\py> 
```

**QUE 7:** WAP to find sum of digits of 'n' digit number.

**SOLUTION:**

```
// to find sum of digits of an 'n' digit number.

#include <iostream>

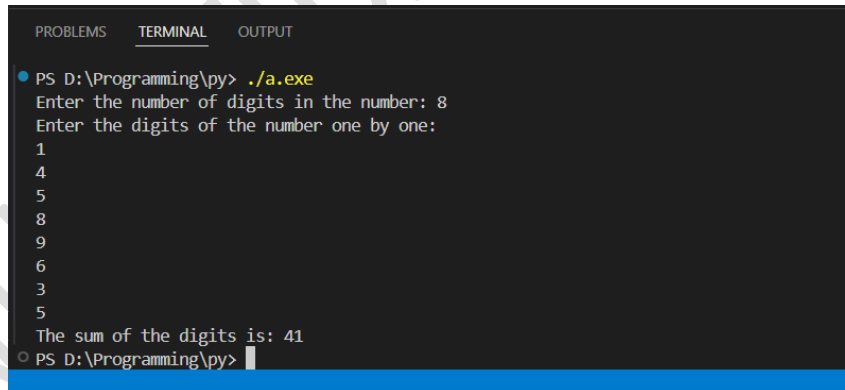
using namespace std;
```

```

int main()
{
    int n, digits[n];
    cout << "Enter the number of digits in the number: ";
    cin >> n;
    cout << "Enter the digits of the number one by one: " << endl;
    for (int i = 0; i < n; i++)
    {
        cin >> digits[i];
    }
    int sum = 0;
    for (int i = 0; i < n; i++)
    {
        sum += digits[i];
    }
    cout << "The sum of the digits is: " << sum << endl;
    return 0;
}

```

#### OUTPUT:



```

PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> ./a.exe
Enter the number of digits in the number: 8
Enter the digits of the number one by one:
1
4
5
8
9
6
3
5
The sum of the digits is: 41
PS D:\Programming\py>

```

**QUE 8:** WAP to find the reverse of a number.

#### SOLUTION:

```

#include <iostream>
using namespace std;
int main()
{

```

```

int number, reverse = 0;

cout << "Enter a number: ";

cin >> number;

while (number != 0) {

    int digit = number % 10;

    reverse = reverse * 10 + digit;

    number /= 10;

}

cout << "The reverse of the number is: " << reverse << endl;

return 0;

}

```

### OUTPUT:

```

PROBLEMS  TERMINAL  OUTPUT

9
6
3
5
The sum of the digits is: 41
● PS D:\Programming\py> g++ fact.cpp
● PS D:\Programming\py> ./a.exe
Enter a number: 5654
The reverse of the number is: 4565
● PS D:\Programming\py> ./a.exe
Enter a number: 864952
The reverse of the number is: 259468
○ PS D:\Programming\py>

```

**QUE 9 :** WAP to determine the given number is a palindrome or not.

### SOLUTION:

```

#include <iostream>

using namespace std;

int main()

{

    int number, original, reverse = 0;

    cout << "Enter a number: ";

    cin >> number;

    original = number;

    for (; number != 0; number /= 10)

```



```

{
    int digit = number % 10;
    reverse = reverse * 10 + digit;
}
if (original == reverse)
{
    cout << "The number is a palindrome." << endl;
}
else
{
    cout << "The number is not a palindrome." << endl;
}
return 0;
}

```

#### OUTPUT:

```

PROBLEMS  TERMINAL  OUTPUT
● PS D:\Programming\py> ./a.exe
  Enter a number: 12214
  The number is not a palindrome.
● PS D:\Programming\py> g++ fact.cpp
● PS D:\Programming\py> ./a.exe
  Enter a number: 51215
  The number is a palindrome.
○ PS D:\Programming\py>

```

**QUE 10:** WAP to print Fibonacci series up to 'n' terms.

#### SOLUTION:

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cout << "Enter the number of terms for the Fibonacci series: ";
    cin >> n;
    int n1 = 0, n2 = 1, next;
    cout << "Fibonacci Series: ";

```

```

for (int i = 1; i <= n; i++) {
    if (i == 1) {
        cout << n1 << " ";
        continue;
    }
    if (i == 2) {
        cout << n2 << " ";
        continue;
    }
    next = n1 + n2;
    cout << next << " ";
    n1 = n2;
    n2 = next;
}
cout << endl;
return 0;
}

```

#### OUTPUT:

```

PROBLEMS  TERMINAL  OUTPUT
• PS D:\Programming\py> g++ fact.cpp
• PS D:\Programming\py> ./a.exe
Enter the number of terms for the Fibonacci series: 5
Fibonacci Series: 0 1 1 2 3
• PS D:\Programming\py> ./a.exe
Enter the number of terms for the Fibonacci series: 10
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34
○ PS D:\Programming\py>

```

**QUE 11:** WAP to determine given n digit number is Armstrong number or not.

#### SOLUTION:

```

#include <iostream>

using namespace std;

int main()
{

```

```

int num,n,new_no,digit,power;

cout << "Enter the number: ";

cin >> num;

n = num;

if (n==0)
{
    digit=1;
}
else {
    for(digit=0; n!=0; digit++)
        n=n/10;
}

n = num;

int ans=0;

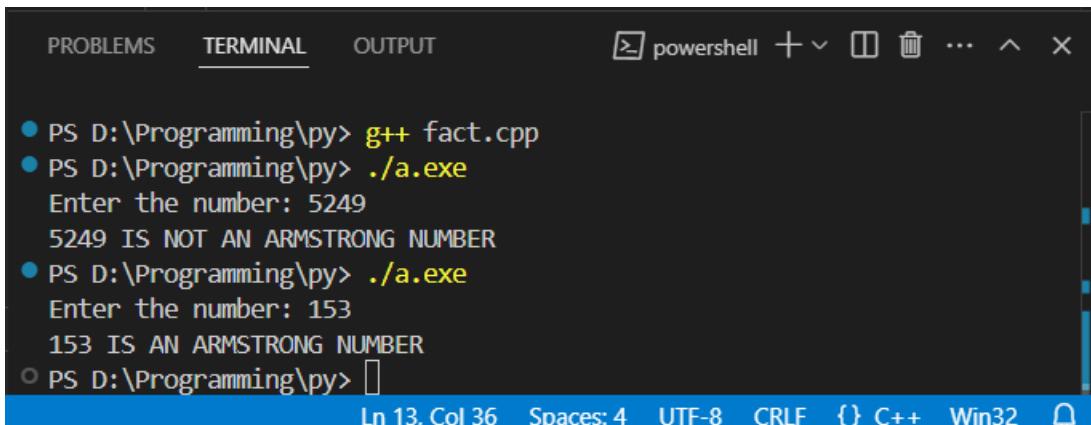
while(n!=0) {
    new_no=n%10;
    power=1;
    for( int i=0; i<digit; i++) {
        power=power*new_no;
    }
    ans=ans+power;
    n=n/10;
}

if (ans==num)
{ cout<<num<<" IS AN ARMSTRONG NUMBER ";}
else {
    { cout<<num<<" IS NOT AN ARMSTRONG NUMBER ";}
}

return 0;
}

```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT  powershell + - [ ] [ ] ... ^ x

• PS D:\Programming\py> g++ fact.cpp
• PS D:\Programming\py> ./a.exe
Enter the number: 5249
5249 IS NOT AN ARMSTRONG NUMBER
• PS D:\Programming\py> ./a.exe
Enter the number: 153
153 IS AN ARMSTRONG NUMBER
• PS D:\Programming\py> [ ]

Ln 13, Col 36  Spaces: 4  UTF-8  CRLF  {} C++  Win32  [ ]
```

**QUE 12:** WAP to find all even numbers between 100 & 200.

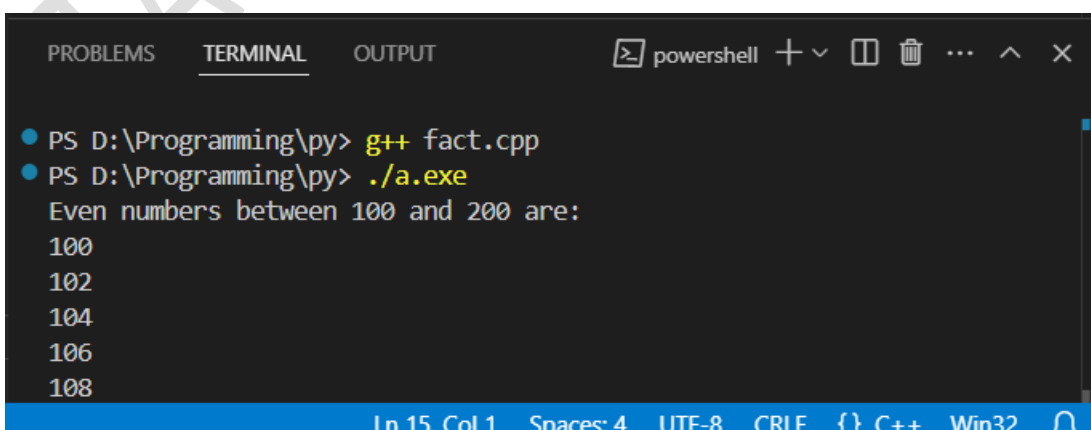
**SOLUTION:**

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Even numbers between 100 and 200 are:" << endl;
    for (int i = 100; i <= 200; i++)
    {
        if (i % 2 == 0)
        {
            cout << i << endl;
        }
    }
    return 0;
}
```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT  powershell + - [ ] [ ] ... ^ x

• PS D:\Programming\py> g++ fact.cpp
• PS D:\Programming\py> ./a.exe
Even numbers between 100 and 200 are:
100
102
104
106
108

Ln 15, Col 1  Spaces: 4  UTF-8  CRLF  {} C++  Win32  [ ]
```

**QUE 13:** WAP to print first 50 prime numbers.

**SOLUTION:**

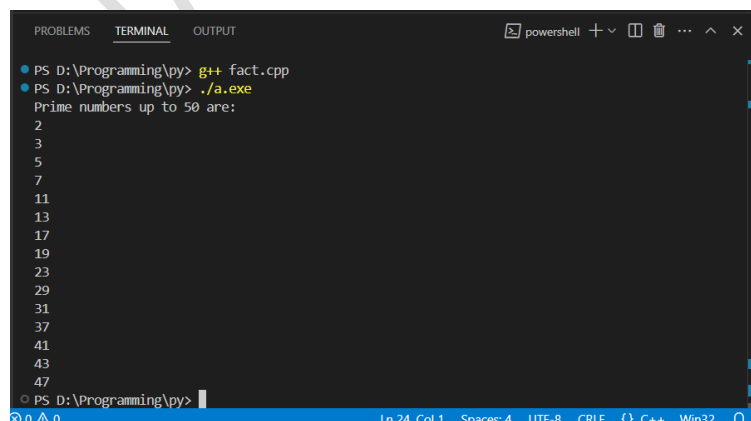
```
#include <iostream>

using namespace std;

int main()
{
    cout << "Prime numbers up to 50 are: " << endl;

    for (int n = 2; n <= 50; n++)
    {
        int counter = 1;
        for (int i = 2; i < n; i++)
        {
            if (n % i == 0)
            {
                counter = 0;
                break;
            }
        }
        if (counter == 1) {
            cout << n << endl;
        }
    }
    return 0;
}
```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
Prime numbers up to 50 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
PS D:\Programming\py>
```

**QUE 14:** WAP to print all 4-digit Armstrong numbers.

**SOLUTION:**

```
#include <iostream>

using namespace std;

int main()
{
    int counter,sum=0;
    cout << "4-digit Armstrong numbers are:" << endl;
    for (int num = 1000; num < 10000; num++) {
        counter = num;
        sum=0;
        while (counter != 0) {
            int digit = counter % 10;
            int power = 1;
            for (int i = 0; i < 4; i++) {
                power *= digit;
            }
            sum += power;
            counter /= 10;
        }
        if (sum == num) {
            cout << num << " ";
        }
    }
    cout << endl;
    return 0;
}
```

**OUTPUT:**

```
PROBLEMS  TERMINAL  OUTPUT
powershell + v [ ] [ ] ... ^ x

4-digit Armstrong numbers are:
• PS D:\Programming\py> g++ fact.cpp
• PS D:\Programming\py> ./a.exe
4-digit Armstrong numbers are:
1634 8208 9474
• PS D:\Programming\py> g++ fact.cpp
• PS D:\Programming\py> ./a.exe
4-digit Armstrong numbers are:
1634 8208 9474
• PS D:\Programming\py> [ ]

0 0 0 Ln 15, Col 14 Spaces: 4 UTF-8 CRLF {} C++ Win32 [ ]
```

**QUE 15:** WAP to print patterns

(a) \*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

**SOLUTION:**

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 0; i <= 5; i++)
    {
        for (int j = 0; j < i; j++)
        {
            cout << "* ";
        }
        cout << "\n";
    }
    return 0;
}
```

**OUTPUT:**

```
PROBLEMS  TERMINAL  OUTPUT

* * *
* * * *
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe

*
* *
* * *
* * * *
* * * * *
PS D:\Programming\py> 
```

(b) \*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**SOLUTION:**

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 0; i <= 5; i++)
    {
        for (int j = 5 - i; j > 0; j--)
        {
            cout << "* ";
        }
        cout << "\n";
    }
    return 0;
}
```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT
powershell + v [ ] [ ] ... ^ X

* * * * *
* * * * *
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
* * * * *
* * * * *
* * * * *
* * *
* *
*

PS D:\Programming\py> [ ]
0 0 0  Ln 14, Col 2  Spaces: 4  UTF-8  CRLF  {} C++  Win32  [ ]
```

(c)     \*  
          \* \* \*  
          \* \* \* \* \*

**SOLUTION:**

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 0; i < 3; i++)
    {
        for (int k = 0; k < 3 - i; k++)
        {
            cout << " ";
        }
        for (int j = 0; j < 2 * i + 1; j++)
        {
            cout << "* ";
        }
        cout << endl;
    }
    return 0;
}
```

**OUTPUT:**

```
PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
*
* * *
* * * * *
PS D:\Programming\py>
```

⊗ 0 △ 0      Ln 18, Col 2   Spaces: 4   UTF-8   CRLF   { } C++   Win32   🔔

(d) 1  
22  
333  
4444

### SOLUTION:

```
#include <iostream>
using namespace std;
int main()
{
    int count = 0;
    for (int i = 0; i < 4; i++)
    {
        count++;
        for (int j = 0; j <= i; j++)
        {
            cout << count << " ";
        }
        cout << endl;
    }
    return 0;
}
```

### OUTPUT:

```
PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> ./a.exe
1
2 2
3 3 3
4 4 4 4
PS D:\Programming\py>
```

⊗ 0 △ 0      Ln 16, Col 2   Spaces: 4   UTF-8   CRLF   { } C++   Win32   🔔

### (e) Pascal's triangle:

#### SOLUTION:

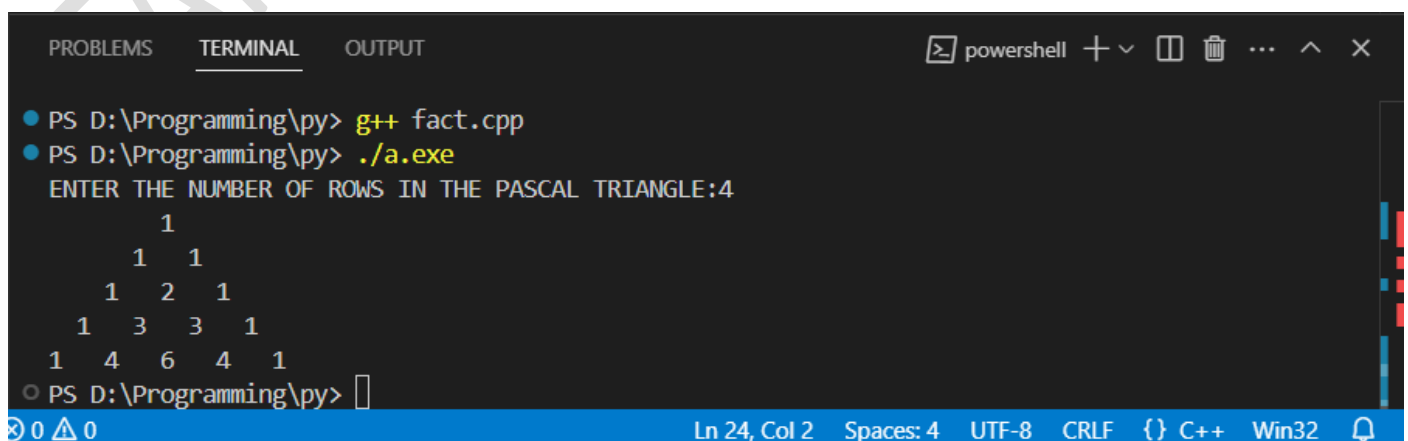
```
#include <iostream>

using namespace std;

int main()
{
    int i, j, n, coef;

    cout << "ENTER THE NUMBER OF ROWS IN THE PASCAL TRIANGLE:";
    cin >> n;
    for (int i = 0; i <= n; i++)
    {
        for (int space = i; space < n; ++space)
        {
            cout << " ";
        }
        coef = 1;
        for (int j = 0; j <= i; j++)
        {
            cout << coef << " ";
            coef = coef * (i - j) / (j + 1);
        }
        cout << endl;
    }
    return 0;
}
```

#### OUTPUT:



The screenshot shows a PowerShell terminal window with the following content:

```
PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
ENTER THE NUMBER OF ROWS IN THE PASCAL TRIANGLE:4
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
PS D:\Programming\py>
```

The terminal window has a title bar that says "powershell" and standard window controls. The status bar at the bottom shows "Ln 24, Col 2", "Spaces: 4", "UTF-8", "CRLF", "{ } C++", "Win32", and a bell icon.

**(f) FLOYD'S TRIANGLE.**

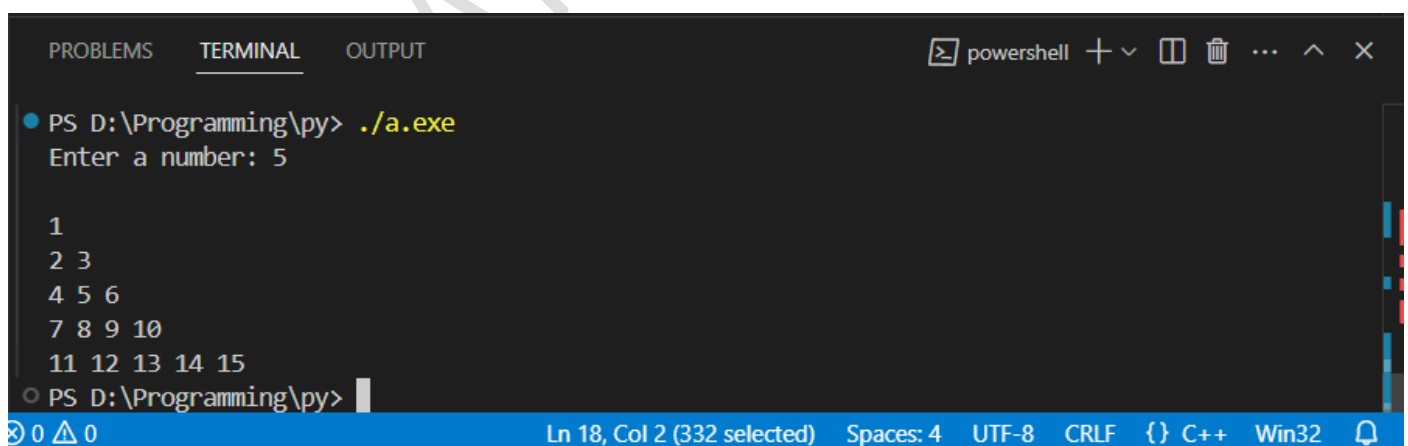
**SOLUTION:**

```
#include <iostream>

using namespace std;

int main()
{
    int n, count = 0;
    cout << "Enter a number: ";
    cin >> n;
    for (int i = 0; i <= n; i++)
    {
        for (int j = 0; j < i; j++)
        {
            count++;
            cout << count << " ";
        }
        cout << endl;
    }
    return 0;
}
```

**OUTPUT:**

A screenshot of a C++ IDE window. The 'TERMINAL' tab is active, showing the command prompt 'PS D:\Programming\py> ./a.exe' and the user input 'Enter a number: 5'. The output displays Floyd's Triangle for n=5: 1, 2 3, 4 5 6, 7 8 9 10, 11 12 13 14 15. The IDE interface includes tabs for 'PROBLEMS', 'TERMINAL', and 'OUTPUT'. The status bar at the bottom shows 'Ln 18, Col 2 (332 selected)', 'Spaces: 4', 'UTF-8', 'CRLF', '{ } C++', 'Win32', and a bell icon.

**QUE 16:** using functions, write following c++ program.

**(a)** To print all the palindrome for range 500-1000.

**SOLUTION:**

```

#include <iostream>

using namespace std;

int main()
{
    int num, orig, rev = 0;

    cout << "Palindrome numbers between 500 and 1000 are:" << endl;

    for (int num = 500; num <= 1000; num++)
    {
        orig = num;
        rev = 0;
        while (orig != 0)
        {
            int digit = orig % 10;
            rev = rev * 10 + digit;
            orig /= 10;
        }
        if (rev == num)
        {
            cout << num << " ";
        }
    }

    return 0;
}

```

## OUTPUT:

```

PS D:\Programming\py> ./a.exe
Palindrome numbers between 500 and 1000 are:
505,515,525,535,545,555,565,575,585,595,606,616,626,636,646,656,666,676,686,696,707,717,727,737,747,757,767,777,787,797,808,818,828,838,848,858,868,878,888,898,909,919,929,939,949,959,969,979,989,999,
PS D:\Programming\py> ./a.exe
Palindrome numbers between 500 and 1000 are:
505,515,525,535,545,555,565,575,585,595,606,616,626,636,646,656,666,676,686,696,707,717,727,737,747,757,767,777,787,797,808,818,828,838,848,858,868,878,888,898,909,919,929,939,949,959,969,979,989,999,
PS D:\Programming\py>

```

(b) To print first 100 odd numbers.

## SOLUTION:

```

#include <iostream>

using namespace std;

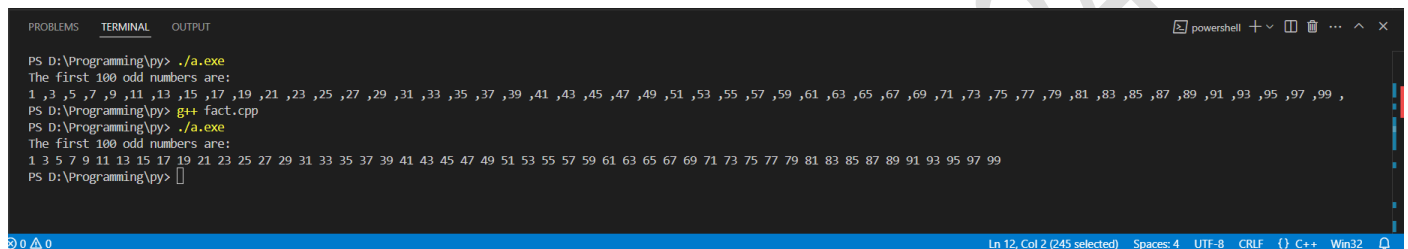
```

```

int main()
{
    cout << "The first 100 odd numbers are:" << endl;
    for (int num = 1; num <= 100; num++)
    {
        if (num % 2 != 0)
            cout << num << " ";
    }
    return 0;
}

```

### OUTPUT:



```

PS D:\Programming\py> ./a.exe
The first 100 odd numbers are:
1 ,3 ,5 ,7 ,9 ,11 ,13 ,15 ,17 ,19 ,21 ,23 ,25 ,27 ,29 ,31 ,33 ,35 ,37 ,39 ,41 ,43 ,45 ,47 ,49 ,51 ,53 ,55 ,57 ,59 ,61 ,63 ,65 ,67 ,69 ,71 ,73 ,75 ,77 ,79 ,81 ,83 ,85 ,87 ,89 ,91 ,93 ,95 ,97 ,99 ,
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
The first 100 odd numbers are:
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
PS D:\Programming\py>

```

(c) To find binary, octal & hexadecimal equivalents of a number.

### SOLUTION:

```

#include <iostream>
using namespace std;
int main()
{
    int number;
    cout << "Enter a number: ";
    cin >> number;
    cout << "Binary equivalent: ";
    for (int i = 15; i >= 0; i--)
    {
        cout << ((number >> i) & 1);
    }
    cout << endl;
    cout << "Octal equivalent: " << oct << number << endl;
    cout << "Hexadecimal equivalent: " << hex << number << endl;
}

```

```

    return 0;
}

```

## OUTPUT:

```

PROBLEMS  TERMINAL  OUTPUT
Enter a number: 15
Binary equivalent: 0000000000001111
Octal equivalent: 17
Hexadecimal equivalent: f
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
Enter a number: 16
Binary equivalent: 0000000000010000
Octal equivalent: 20
Hexadecimal equivalent: 10
PS D:\Programming\py>

```

(d) To find decimal equivalents for binary, octal & hexadecimal numbers.

## SOLUTION:

```

#include <iostream>
#include <string>
using namespace std;
int main()
{
    string input;
    int decimalValue = 0;
    cout << "Enter a number (binary, octal, or hexadecimal): ";
    cin >> input;
    if (input.find("0x") == 0 || input.find("0X") == 0)
    {
        for (size_t i = 2; i < input.length(); i++)
        {
            char c = input[i];
            if (c >= '0' && c <= '9')
            {
                decimalValue = decimalValue * 16 + (c - '0');
            }
            else if (c >= 'A' && c <= 'F')
            {

```

```

        decimalValue = decimalValue * 16 + (c - 'A' + 10);
    }
    else if (c >= 'a' && c <= 'f')
    {
        decimalValue = decimalValue * 16 + (c - 'a' + 10);
    }
}
cout << "Detected Hexadecimal number: " << input << endl;
}
else if (input[0] == '0')
{
    for (size_t i = 1; i < input.length(); i++)
    {
        char c = input[i];
        decimalValue = decimalValue * 8 + (c - '0');
    }
    cout << "Detected Octal number: " << input << endl;
}
else
{
    for (char c : input)
    {
        decimalValue = decimalValue * 2 + (c - '0');
    }
    cout << "Detected Binary number: " << input << endl;
}
cout << "Decimal equivalent: " << decimalValue << endl;

return 0;
}

```

**OUTPUT:**



- PS D:\Programming\py> g++ n.cpp
- PS D:\Programming\py> ./a.exe  
Enter a number (binary, octal, or hexadecimal): 10  
Detected Binary number: 10  
Decimal equivalent: 2
- PS D:\Programming\py> █

(e) to calculate geometric sum up to 'n' terms.

#### SOLUTION:

```
#include <iostream>
using namespace std;
int main()
{
    int n, r, result = 1;
    double sum = 0.0, a;
    cout << "Enter the number of terms (n): ";
    cin >> n;
    cout << "Enter the common ratio between consecutive terms: ";
    cin >> r;
    cout << "Enter the first term of the series: ";
    cin >> a;
    for (int i = 0; i < n; i++)
    {
        result = result * r;
    }
    for (int i = 0; i < n; i++)
    {
        sum = a * ((1 - result) / (1 - r));
    }
    cout << "The geometric sum up to " << n << " terms is: " << sum << endl;
    return 0;
}
```

#### OUTPUT:

The screenshot shows a Visual Studio Code terminal window with the following content:

```
PROBLEMS  TERMINAL  OUTPUT
Enter the number of terms (n): 10
Enter the common ratio between consecutive terms: 5
Enter the first term of the series: 4
The geometric sum up to 10 terms is: 9.76562e+06
• PS D:\Programming\py> g++ fact.cpp
• PS D:\Programming\py> ./a.exe
Enter the number of terms (n): 10
Enter the common ratio between consecutive terms: 2
Enter the first term of the series: 1
The geometric sum up to 10 terms is: 1023
• PS D:\Programming\py> 
```

The status bar at the bottom indicates: Ln 17, Col 32 Spaces: 4 UTF-8 CRLF {} C++ Win32

**QUE 17:** using recursion write c++ program to print factorials for a given range.

**SOLUTION :**

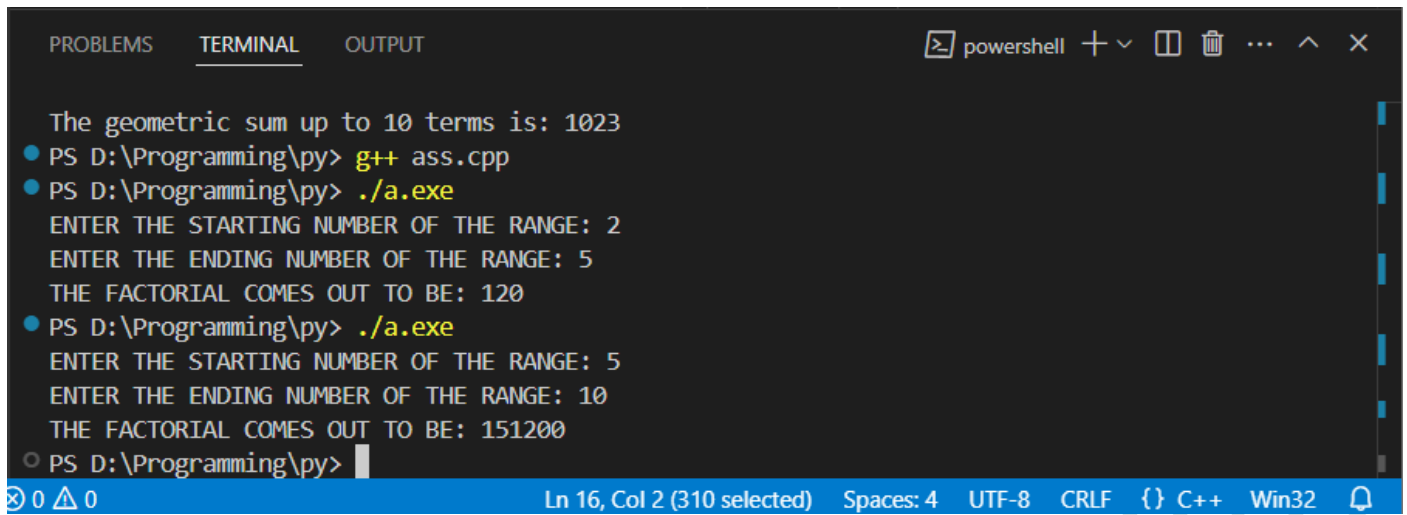
```
#include<iostream>

using namespace std;

int main ( )
{
    int n1,n2, i, a = 1;

    cout<< "ENTER THE STARTING NUMBER OF THE RANGE: ";
    cin>> n1;
    cout<< "ENTER THE ENDING NUMBER OF THE RANGE: ";
    cin>> n2;
    for (i = n1; i <= n2; i++)
    {
        a = a * i;
    }
    cout<< "THE FACTORIAL COMES OUT TO BE: "<< a;
    return 0;
}
```

**OUTPUT:**



```
PROBLEMS  TERMINAL  OUTPUT

The geometric sum up to 10 terms is: 1023
• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
  ENTER THE STARTING NUMBER OF THE RANGE: 2
  ENTER THE ENDING NUMBER OF THE RANGE: 5
  THE FACTORIAL COMES OUT TO BE: 120
• PS D:\Programming\py> ./a.exe
  ENTER THE STARTING NUMBER OF THE RANGE: 5
  ENTER THE ENDING NUMBER OF THE RANGE: 10
  THE FACTORIAL COMES OUT TO BE: 151200
• PS D:\Programming\py>

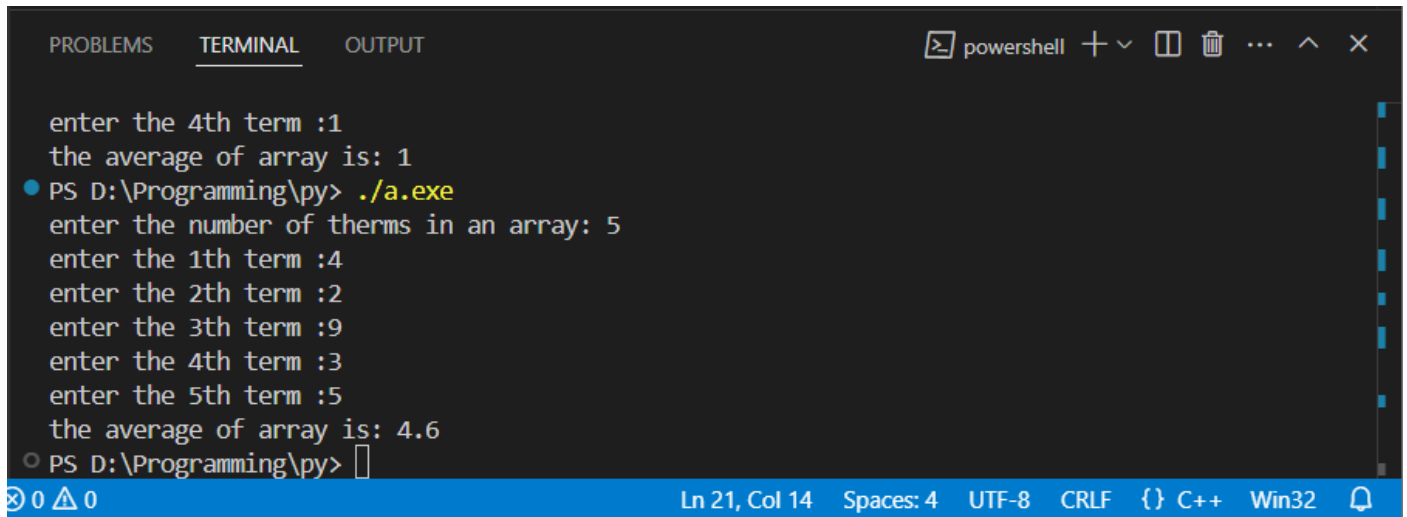
0 0 0  Ln 16, Col 2 (310 selected)  Spaces: 4  UTF-8  CRLF  {} C++  Win32
```

**QUE 18:** WAP to find the average of all the elements of a 1D array.

**SOLUTION:**

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "enter the number of terms in an array: ";
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cout << "enter the " << i + 1 << "(th) term :";
        cin >> arr[i];
    }
    float ar = 0.0;
    for (int i = 0; i < n; i++)
    {
        ar += arr[i];
    }
    double avg = ar / n;
    cout << "the average of array is: " << avg;
    return 0;
}
```

## OUTPUT:



```
PROBLEMS  TERMINAL  OUTPUT  powershell + - [ ] [ ] ... ^ X

enter the 4th term :1
the average of array is: 1
● PS D:\Programming\py> ./a.exe
enter the number of therms in an array: 5
enter the 1th term :4
enter the 2th term :2
enter the 3th term :9
enter the 4th term :3
enter the 5th term :5
the average of array is: 4.6
○ PS D:\Programming\py> [ ]
```

Ln 21, Col 14 Spaces: 4 UTF-8 CRLF {} C++ Win32

**QUE 19:** WAP to find the maximum and minimum elements of an array.

## SOLUTION:

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "enter the number of therms in an array: ";
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cout << "enter the " << i + 1 << "(th) term :";
        cin >> arr[i];
    }
    int max=0;
    for (int i = 0; i < n; i++)
    {
        if(arr[i]>max)
        {max=arr[i];}
        else
        {continue;}
    }
```

```

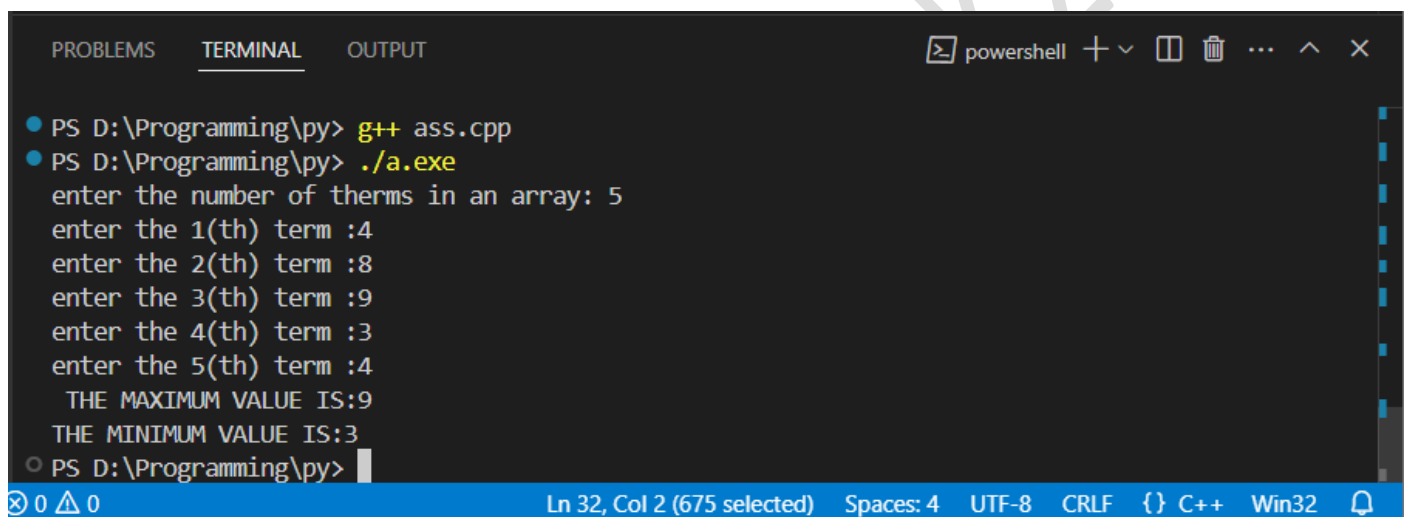
int min=max;

for (int i = 0; i < n; i++)
{
    if(arr[i]<min)
    {min=arr[i];}
    else
    {continue;}
}

cout << " THE MAXIMUM VALUE IS:"<<max<<endl<<"THE MINIMUM VALUE IS:"<<min;
return 0;
}

```

### OUTPUT:



```

PROBLEMS  TERMINAL  OUTPUT
powershell + - [] ... ^ X

• PS D:\Programming\py> g++ ass.cpp
• PS D:\Programming\py> ./a.exe
enter the number of terms in an array: 5
enter the 1(th) term :4
enter the 2(th) term :8
enter the 3(th) term :9
enter the 4(th) term :3
enter the 5(th) term :4
THE MAXIMUM VALUE IS:9
THE MINIMUM VALUE IS:3
PS D:\Programming\py>

```

**QUE 20:** WAP to find the transpose of a matrix.

### SOLUTION:

```

#include <iostream>
using namespace std;

int main()
{
    int rows, cols;

    cout << "Enter the number of rows: ";

    cin >> rows;

    cout << "Enter the number of columns: ";

    cin >> cols;

    int mat[rows][cols], trans[cols][rows];
}

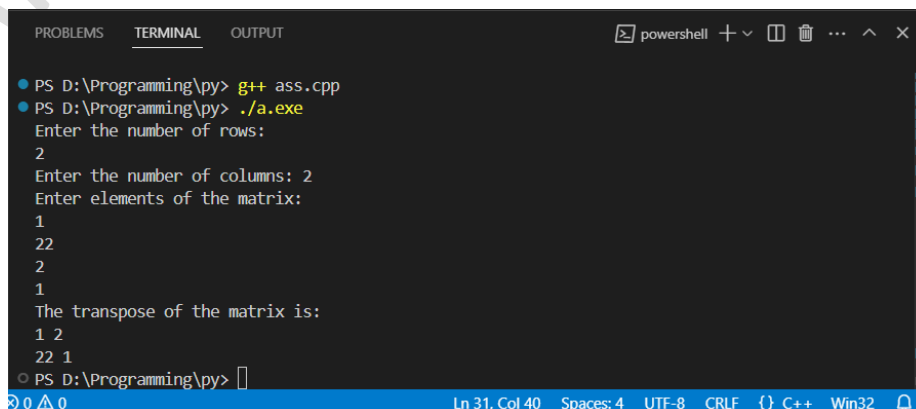
```

```

cout << "Enter elements of the matrix:" << endl;
for (int i = 0; i < rows; i++)
{
    for (int j = 0; j < cols; j++)
    {
        cin >> mat[i][j];
    }
}
for (int i = 0; i < rows; i++)
{
    for (int j = 0; j < cols; j++)
    {
        trans[j][i] = mat[i][j];
    }
}
cout << "The transpose of the matrix is:" << endl;
for (int i = 0; i < cols; i++)
{
    for (int j = 0; j < rows; j++)
    {
        cout << trans[i][j] << " ";
    }
    cout << endl;
}
return 0;
}

```

**OUTPUT:**



```

PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> g++ ass.cpp
PS D:\Programming\py> ./a.exe
Enter the number of rows:
2
Enter the number of columns: 2
Enter elements of the matrix:
1
2
2
1
The transpose of the matrix is:
1 2
2 1
PS D:\Programming\py>

```

**QUE 21:** WAP to add 2D matrix.

**SOLUTION:**

```
#include <iostream>

using namespace std;

int main() {
    int rows, cols;

    cout << "Enter the number of rows: ";
    cin >> rows;

    cout << "Enter the number of columns: ";
    cin >> cols;

    int matrix1[rows][cols], matrix2[rows][cols], result[rows][cols];
    cout << "Enter elements of the first matrix:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cin >> matrix1[i][j];
        }
    }

    cout << "Enter elements of the second matrix:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cin >> matrix2[i][j];
        }
    }

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }

    cout << "The resulting matrix after addition is:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << result[i][j] << " ";
        }
        cout << endl;
    }
}
```

```

    }
    return 0;
}

```

```

PS D:\Programming\py> ./a.exe
Enter the number of rows: 2
Enter the number of columns: 2
Enter elements of the first matrix:
1
2
2
1
Enter elements of the second matrix:
1
2
2
1
The resulting matrix after addition is:
2 4
4 2
PS D:\Programming\py>

```

**OUTPUT:**

**QUE 22:** WAP to multiply 2d matrix.

**SOLUTION:**

```

#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter the number of rows and columns in the square matrix: ";
    cin >> n;
    int matrix1[n][n], matrix2[n][n], result[n][n]={0};
    cout << "Enter elements of the first matrix:" << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> matrix1[i][j];
        }
    }
    cout << "Enter elements of the second matrix:" << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> matrix2[i][j];
        }
    }
}

```



```

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        for (int k = 0; k < n; k++) {
            result[i][j] += matrix1[i][k] * matrix2[k][j];
        }
    }
}

cout << "The resulting matrix after multiplication is:" << endl;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        cout << result[i][j] << " ";
    }
    cout << endl;
}

return 0;
}

```

#### OUTPUT:

```

PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> ./a.exe
Enter the number of rows and columns in the square matrix: 2
Enter elements of the first matrix:
1
2
2
1
Enter elements of the second matrix:
2
1
1
2
The resulting matrix after multiplication is:
4 5
5 4
PS D:\Programming\py>

```

**QUE 23:** WAP to sort an array in ascending order.

#### SOLUTION:

```

#include <iostream>

using namespace std;

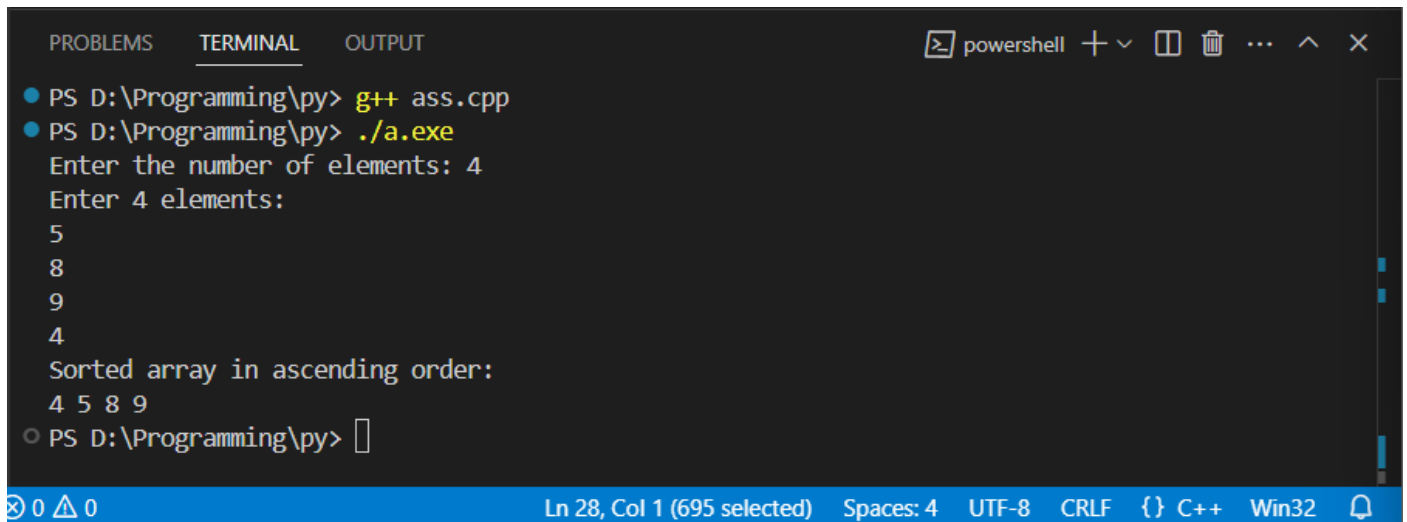
```

```

int main()
{
    int n,arr[n];
    cout << "Enter the number of elements: ";
    cin >> n;
    cout << "Enter " << n << " elements:" << endl;
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    cout << "Sorted array in ascending order:" << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}

```

**OUTPUT:**



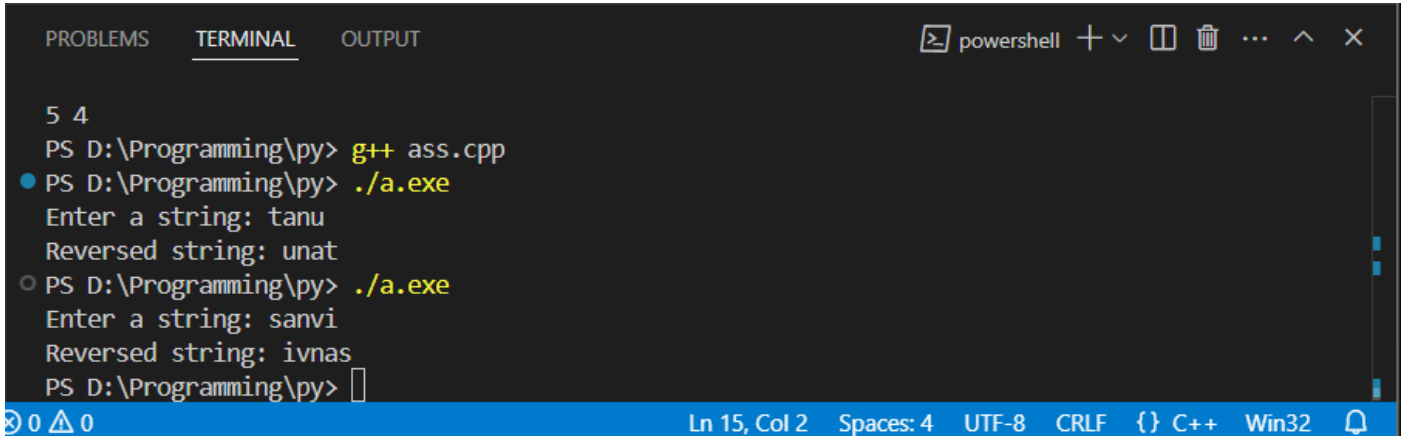
```
PROBLEMS  TERMINAL  OUTPUT
PS D:\Programming\py> g++ ass.cpp
PS D:\Programming\py> ./a.exe
Enter the number of elements: 4
Enter 4 elements:
5
8
9
4
Sorted array in ascending order:
4 5 8 9
PS D:\Programming\py> 
```

**QUE 24:** WAP to reverse a given string.

**SOLUTION:**

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str, rev;
    cout << "Enter a string: ";
    cin >> str;
    int len = str.length();
    for (int i = len - 1; i >= 0; i--) {
        rev += str[i];
    }
    cout << "Reversed string: " << rev;
    return 0;
}
```

**OUTPUT:**



```
5 4
PS D:\Programming\py> g++ ass.cpp
PS D:\Programming\py> ./a.exe
Enter a string: tanu
Reversed string: unat
PS D:\Programming\py> ./a.exe
Enter a string: sanvi
Reversed string: ivnas
PS D:\Programming\py> 
```

**QUE 25:** WAP to count all the vowels in a given string.

**SOLUTION:**

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str;
    int vowelCount = 0;
    cout << "Enter a string: ";
    getline(cin, str);
    for (char c : str)
    {
        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||
            c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
        {
            vowelCount++;
        }
    }
    cout << "Number of vowels in the string: " << vowelCount << endl;
    return 0;
}
```

**OUTPUT:**

```
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
Enter a string: tanu
Number of vowels in the string: 2
PS D:\Programming\py> 
```

**QUE 26:** WAP to check if a given string is a palindrome or not.

**SOLUTION:**

```
#include<iostream>

using namespace std;

int main(){
    string st;
    cout<<"Enter a string \n";
    cin>>st;
    int flag=0;
    int len=st.size();
    for (int i=0;i<len/2;i++)
        if(st[i] != st[len-1-i])
        {
            flag=1;
        }
    if(flag==0)
        cout<<"Palindrome Word";
    else
        cout<<" Not Palindrome Word";
    return 0;
}
```

**OUTPUT:**

```
PS D:\Programming\py> g++ fact.cpp
PS D:\Programming\py> ./a.exe
Enter a string
tanu
Not Palindrome Word
PS D:\Programming\py> ./a.exe
Enter a string
annunna
Palindrome Word
PS D:\Programming\py> 
```

**QUE 27:** WAP to check if a given string is an anagram or not.

**SOLUTION:**

```
#include<iostream>

using namespace std;

int main(){

    int arr[26]={0};

    cout << "enter a size:";

    int size;

    cin >> size;

    cout << "s1:";

    char s1[size];

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

        cin >> s1[i];

    }

    char s2[size];

    cout << "s2:";

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

        cin >> s2[i];

    }

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

        int a = s1[i]-'a';

        arr[a]=arr[a]+1;

    }

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

        int a = s2[i]-'a';

        arr[a]=arr[a]-1;;

    }

    int flag=0;

    for(int i=0;i<26;i++){

        if(arr[i]!=0){

            flag=1;

            break;

        }

    }

}
```

```

(flag==0)? cout << "true" : cout << "false";

return 0;

}

```

### OUTPUT:

```

PROBLEMS  TERMINAL  OUTPUT
false
PS D:\Programming\py> ./a.exe
enter a size:4
s1:tanu
s2:tanu
true
PS D:\Programming\py> ./a.exe
enter a size:2
s1:an
s2:no
false
PS D:\Programming\py>

```

**QUE 28:** Define a class called Car with attributes such as make, model, and year. Include member functions to set and get these attributes. Create an object of the Car class and demonstrate the use of its member functions.

### SOLUTION:

```

#include<iostream>

using namespace std;

class car {
    string make;
    string model;
    int year;
public:
    void setData(){
        cout << "enter make:";
        cin >> make;
        cout << "enter model:";
        cin >> model;
        cout << "year:";
        cin >> year;
    }
}

```

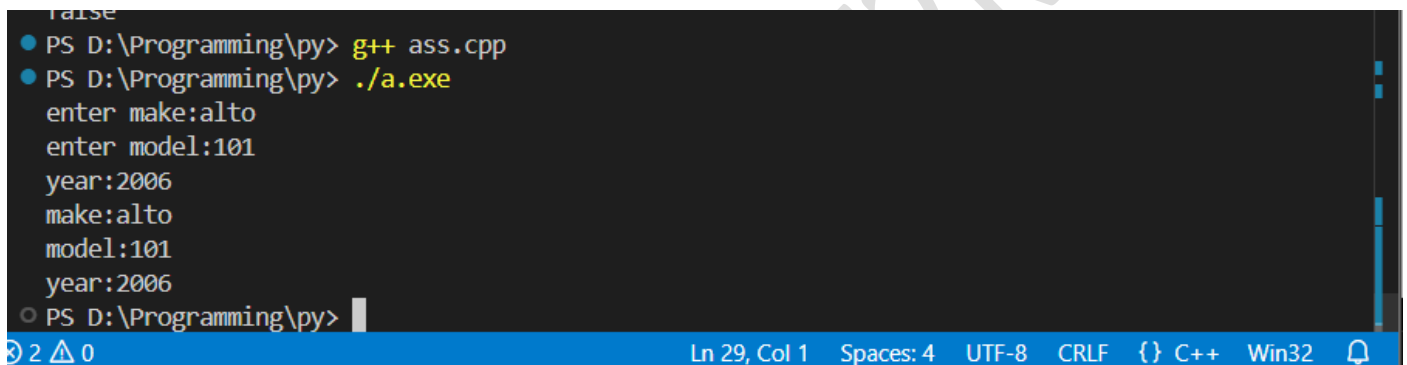
```

void getData(){
    cout << "make:" << make << endl;
    cout << "model:" << model << endl;
    cout << "year:" << year << endl;
}
};

int main () {
    car c1;
    c1.setData();
    c1.getData();
    return 0;
}

```

### OUTPUT:



```

PS D:\Programming\py> g++ ass.cpp
PS D:\Programming\py> ./a.exe
enter make:alto
enter model:101
year:2006
make:alto
model:101
year:2006
PS D:\Programming\py>

```

**Que 29.** Define a class called Address with attributes such as street, city, and zipCode. Create a class called Person that has an Address object as a member variable. Demonstrate composition by creating a Person object and accessing its Address attribute.

### SOLUTION:

```

#include <iostream>
#include <string>
using namespace std;
class Address {
public:
    string street;
    string city;
    string zipCode;
    Address(string s, string c, string z)
{

```



```

        street = s;
        city = c;
        zipCode = z;
    }
};

class Person {
public:
    string name;
    Address address;
    Person(string n, string s, string c, string z) {
        name = n;
        address = Address(s, c, z); // Initialize Address object
    }

    // Function to display Person details
    void displayDetails() {
        cout << "Name: " << name << endl;
        cout << "Address: " << address.street << ", " << address.city << " " << address.zipCode
        << endl;
    }
};

int main() {
    Person person("John Doe", "123 Main St", "Anytown", "12345");
    person.displayDetails();
    return 0;
}

```

#### OUTPUT:

```

Default case is Matched.
PS D:\Programming\py> g++ n.cpp
PS D:\Programming\py> ./a.exe
Name: John Doe
Address: 123 Main St, Anytown 12345
PS D:\Programming\py>

```

**QUE 30:** Write a program to display the minimum, maximum, sum, search and average of elements of an array.

#### SOLUTION:

```
#include <iostream>
```

```

using namespace std;

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;
    int arr[n];
    cout << "Enter the elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    // Find minimum
    int minVal = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] < minVal) {
            minVal = arr[i];
        }
    }
    // Find maximum
    int maxVal = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] > maxVal) {
            maxVal = arr[i];
        }
    }
    // Calculate sum
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    // Search for an element
    int searchVal;
    cout << "Enter the value to search: ";
    cin >> searchVal;
    bool found = false;

```

```

for (int i = 0; i < n; i++) {
    if (arr[i] == searchVal) {
        found = true;
        break;
    }
}

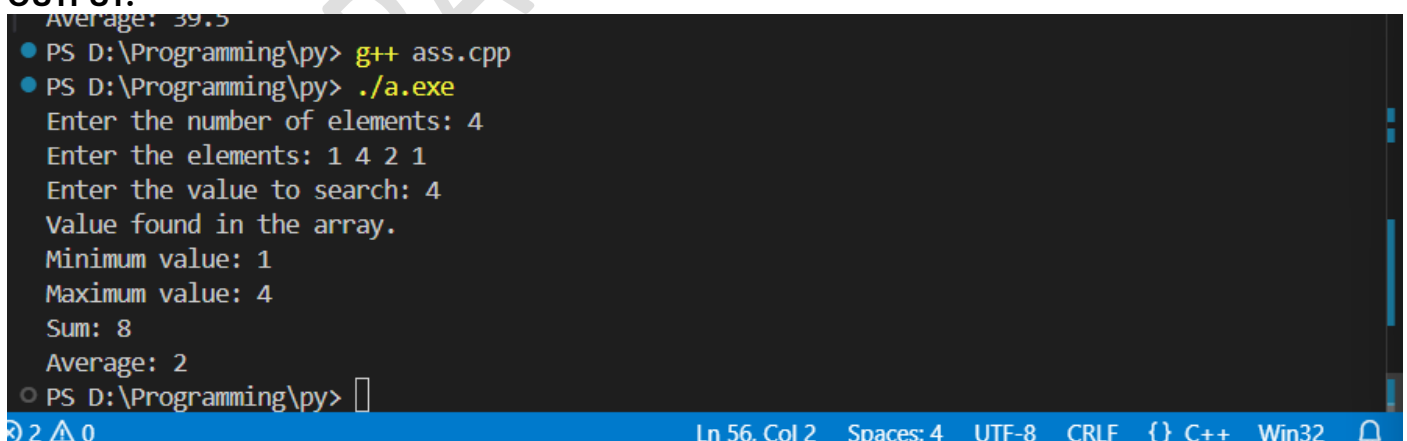
if (found) {
    cout << "Value found in the array." << endl;
} else {
    cout << "Value not found in the array." << endl;
}

// Calculate average
double average = (double)sum / n;

// Display results
cout << "Minimum value: " << minVal << endl;
cout << "Maximum value: " << maxVal << endl;
cout << "Sum: " << sum << endl;
cout << "Average: " << average << endl;
return 0;
}

```

#### OUTPUT:



```

Average: 39.5
PS D:\Programming\py> g++ ass.cpp
PS D:\Programming\py> ./a.exe
Enter the number of elements: 4
Enter the elements: 1 4 2 1
Enter the value to search: 4
Value found in the array.
Minimum value: 1
Maximum value: 4
Sum: 8
Average: 2
PS D:\Programming\py>

```

**QUE 31:** Define a class student with the following specification

Private members of class student

admno                    integer

sname                    20 character

eng. math, science     float

total                     float

Public member function of class student

ctotal()                  a function to calculate eng + math + science with float return type.

Takedata()              Function to accept values for admno, sname, eng, science Showdata()  
Function to display all the data members on the screen.

### **SOLUTION:**

```
#include <iostream>

using namespace std;

class Student {
private:
    int admno;
    char sname[20];
    float eng, math, science;
    float total;
public:
    // Function to calculate total
    float ctotal() {
        total = eng + math + science;
        return total;
    }

    void takeData() {
        cout << "Enter admission number: ";
        cin >> admno;
        cout << "Enter student name: ";
        cin >> sname;
        cout << "Enter English marks: ";
        cin >> eng;
        cout << "Enter Math marks: ";
        cin >> math;
        cout << "Enter Science marks: ";
        cin >> science;
    }
}
```

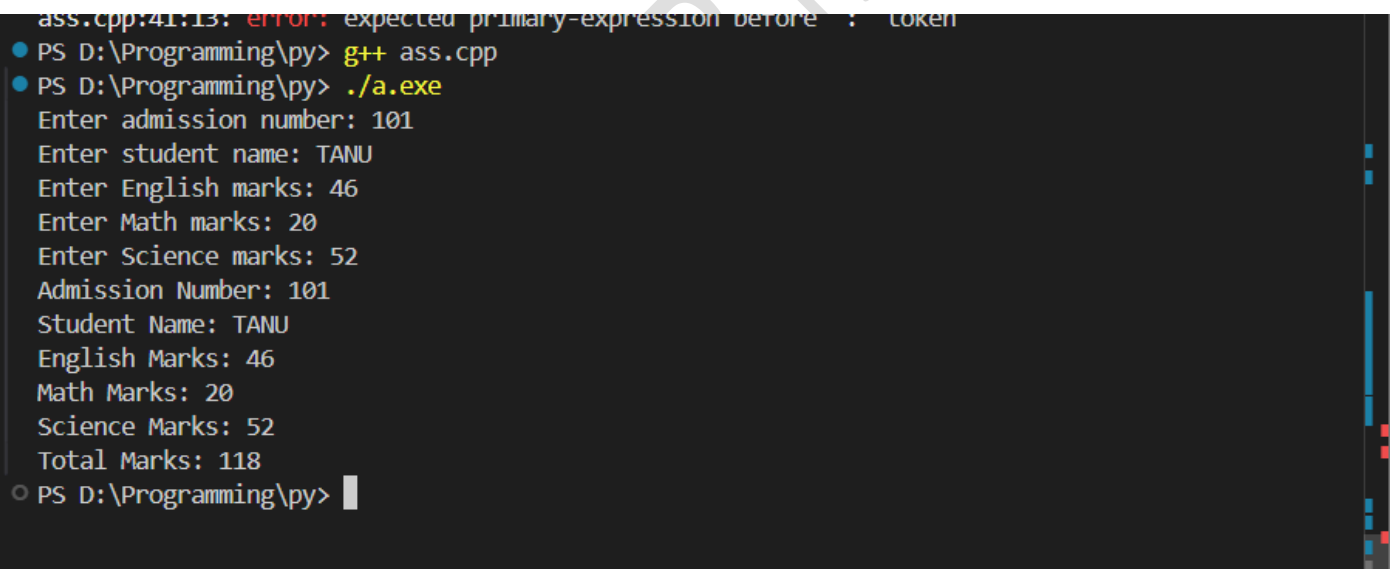
```

void showData() {
    cout << "Admission Number: " << admno << endl;
    cout << "Student Name: " << sname << endl;
    cout << "English Marks: " << eng << endl;
    cout << "Math Marks: " << math << endl;
    cout << "Science Marks: " << science << endl;
    cout << "Total Marks: " << ctototal() << endl;
}
};

int main() {
    Student student;
    student.takeData();
    student.showData();
    return 0;
}

```

#### OUTPUT:



```

ass.cpp:41:13: error: expected primary-expression before '': token
PS D:\Programming\py> g++ ass.cpp
PS D:\Programming\py> ./a.exe
Enter admission number: 101
Enter student name: TANU
Enter English marks: 46
Enter Math marks: 20
Enter Science marks: 52
Admission Number: 101
Student Name: TANU
English Marks: 46
Math Marks: 20
Science Marks: 52
Total Marks: 118
PS D:\Programming\py>

```

**QUE 32:** Define a class in C++ with following description:

Private Members

A data member Flight number of type integer

A data member Destination of type string

A data member Distance of type float

A data member Fuel of type float

A member function CALFUEL() to calculate the value of Fuel as per the following criteria

| Distance                  | Fuel |
|---------------------------|------|
| <=1000                    | 500  |
| more than 1000 and <=2000 | 1100 |
| more than 2000            | 2200 |

### Public Members

A function FEEDINFO() to allow user to enter values for Flight Number, Destination, Distance & call function CALFUEL() to calculate the quantity of Fuel.

A function SHOWINFO() to allow user to view the content of all the data members.

### SOLUTION:

```
#include<iostream>

using namespace std;

class travel{
    int flightNumber;
    string destination;
    int distance;
    float fuel;
    void calFuel(){
        if(distance<=1000) fuel=500;
        else if(distance>1000 && distance<=2000) fuel=1100;
        else fuel=2200;
    }
    public :
    void feedInfo(int fn,string des,int dist){
        flightNumber=fn;
        destination=des;
        distance=dist;
        calFuel();
    }
    void showInfo(){
        cout << "flight number:" << flightNumber << endl;
        cout << "destination:" << destination << endl;
        cout << "distance:" << distance << endl;
        cout << "fuel:" << fuel << endl;
    }
}
```

```
};

int main(){

    travel t1;

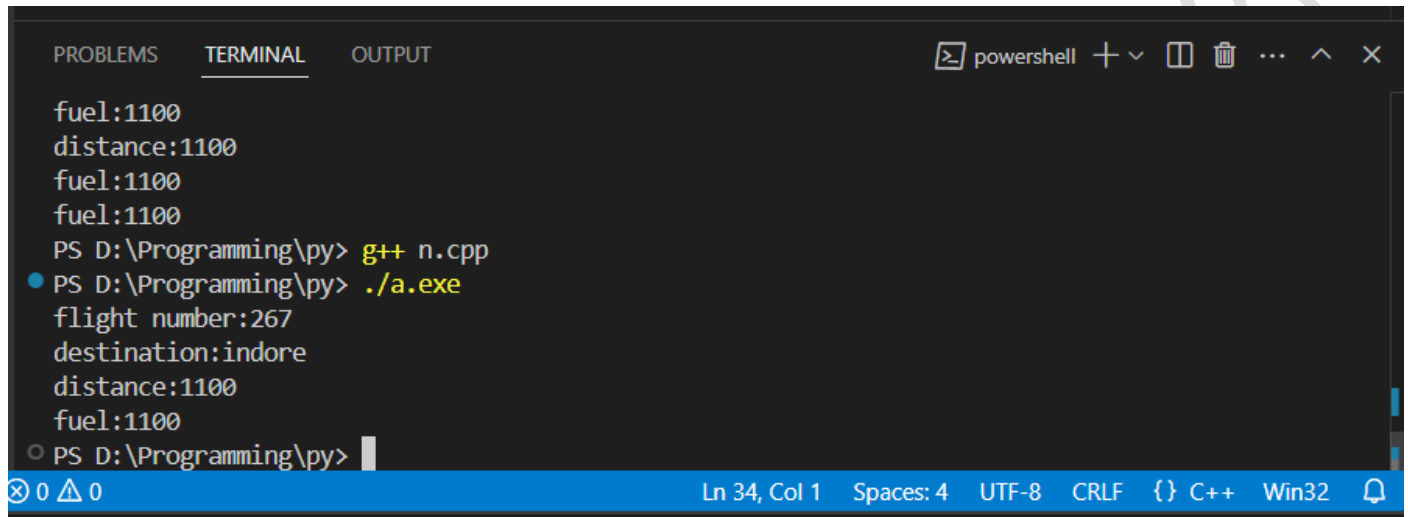
    t1.feedInfo(267,"indore",1100);

    t1.showInfo();

    return 0;

}
```

### OUTPUT:



```
PROBLEMS  TERMINAL  OUTPUT
fuel:1100
distance:1100
fuel:1100
fuel:1100
PS D:\Programming\py> g++ n.cpp
PS D:\Programming\py> ./a.exe
flight number:267
destination:indore
distance:1100
fuel:1100
PS D:\Programming\py>
```

**QUE 33:** Write a menu driven program to perform following:

- Input a matrix
- Display matrix
- Add two matrices
- Multiply two matrices
- Transpose a matrix

### SOLUTION:

```
#include <iostream>
using namespace std;
class matrix
{
    int arr1[3][3];
    int arr2[3][3];

public:
    void Switch(int button)
```

```

{
    switch (button)
    {
    case 1:
        inputdata();
        break;
    case 2:
        displaydata();
        break;
    case 3:
        add();
        break;
    case 4:
        multiply();
        break;
    case 5:
        transpose();
        break;
    default:
        printf("Default case is Matched.");
        break;
    }
}

void inputdata()
{
    cout << "enter 9 elements for matrix 1:";
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cin >> arr1[i][j];
        }
    }

    cout << " enter 9 elements for matrix 2:";
}

```



```

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        cin >> arr2[i][j];
    }
}

void displaydata()
{
    cout << "array 1:\n";
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cout << arr1[i][j] << " ";
        }
        cout << endl;
    }
    cout << "array 2:\n";
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cout << arr2[i][j] << " ";
        }
        cout << endl;
    }
}

void add()
{
    cout << "sum of two matrix";
    for (int i = 0; i < 3; i++)
    {

```

```

    for (int j = 0; j < 3; j++)
    {
        int r = arr1[i][j] + arr2[i][j];
        cout << r << " ";
    }
    cout << endl;
}
}
void multiply()
{
    printf("the resultant matrix\n");
    for (int i = 0; i < 3; i++)
    {
        int d = 0;
        for (int j = 0; j < i; j++)
        {
            d = arr2[i][j];
            arr2[i][j] = arr2[j][i];
            arr2[j][i] = d;
        }
    }
    int r = 0;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            for (int k = 0; k < 3; k++)
            {
                r = arr1[i][k] * arr2[j][k] + r;
            }
            cout << r << " ";
        }
    }
}
}

```

```

void transpose()
{
    cout << "transpose of both matrix:";
    for (int i = 0; i < 3; i++)
    {
        int d = 0;
        for (int j = 0; j < i; j++)
        {
            d = arr1[i][j];
            arr1[i][j] = arr1[j][i];
            arr1[j][i] = d;
        }
    }
    for (int i = 0; i < 3; i++)
    {
        int d = 0;
        for (int j = 0; j < i; j++)
        {
            d = arr2[i][j];
            arr2[i][j] = arr2[j][i];
            arr2[j][i] = d;
        }
    }
    displaydata();
}

};

int main()
{
    matrix m1;

    cout << "enter 1 for input matrix:" << endl;
    cout << "enter 2 for output matrix:" << endl;
    cout << "enter 3 for add two matrix:" << endl;
    cout << "enter 4 for multiply two matrix:" << endl;
    cout << "enter 5 for transpose of matrix:" << endl;

```

```

int button;

cout << "enter button:";

cin >> button;

m1.Switch(button);

cout << "enter button:";

cin >> button;

m1.Switch(button);

cout << "enter button:";

cin >> button;

m1.Switch(button);

return 0;

}

```

## OUTPUT:

```

9
● PS D:\Programming\py> g++ n.cpp
○ PS D:\Programming\py> ./a.exe
enter 1 for input matrix:
enter 2 for output matrix:
enter 3 for add two matrix:
enter 4 for multiply two matrix:
enter 5 for transpose of matrix:
enter button:1
enter 9 elements for matrix 1:1 2 3 6 5 4 7 8 9
enter 9 elements for matrix 2:1 2 3 6 9 8 7 4 5
enter button:5
transpose of both matrix:array 1:
1 6 7
2 5 8
3 4 9
array 2:
1 6 7
2 9 4
3 8 5
enter button:

```