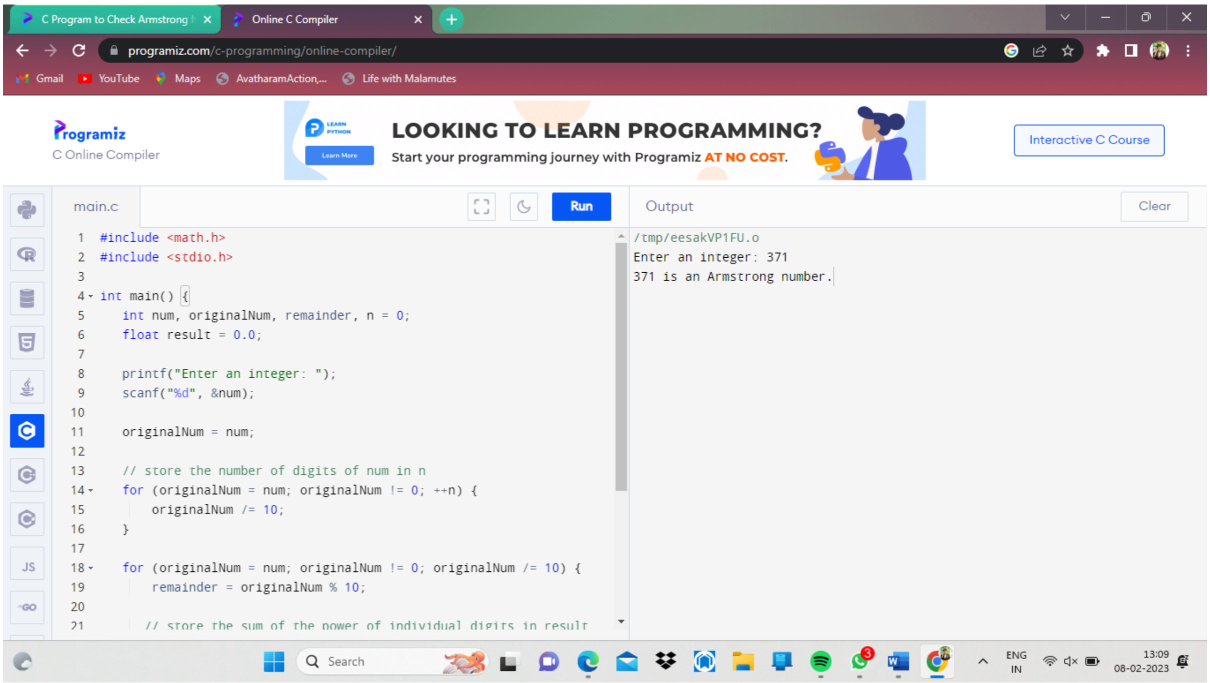
CSA0672 - DESIGN AND ANALYSIS OF ALGORITHM

NAME:V.V.L.Aishwarya

REG.NO:192011268

1) Armstrong code:



```
#include <math.h>
#include <stdio.h>

int main() {
    int num, originalNum, remainder, n = 0;
    float result = 0.0;

    printf("Enter an integer: ");
    scanf("%d", &num);

    originalNum = num;

    // store the number of digits of num in n
    for (originalNum = num; originalNum != 0; ++n) {
        originalNum /= 10;
    }

    for (originalNum = num; originalNum != 0; originalNum /= 10) {
        remainder = originalNum % 10;

        // store the sum of the power of individual digits in result
```

```
/tmp/eesakVP1FU.o
Enter an integer: 371
371 is an Armstrong number.
```

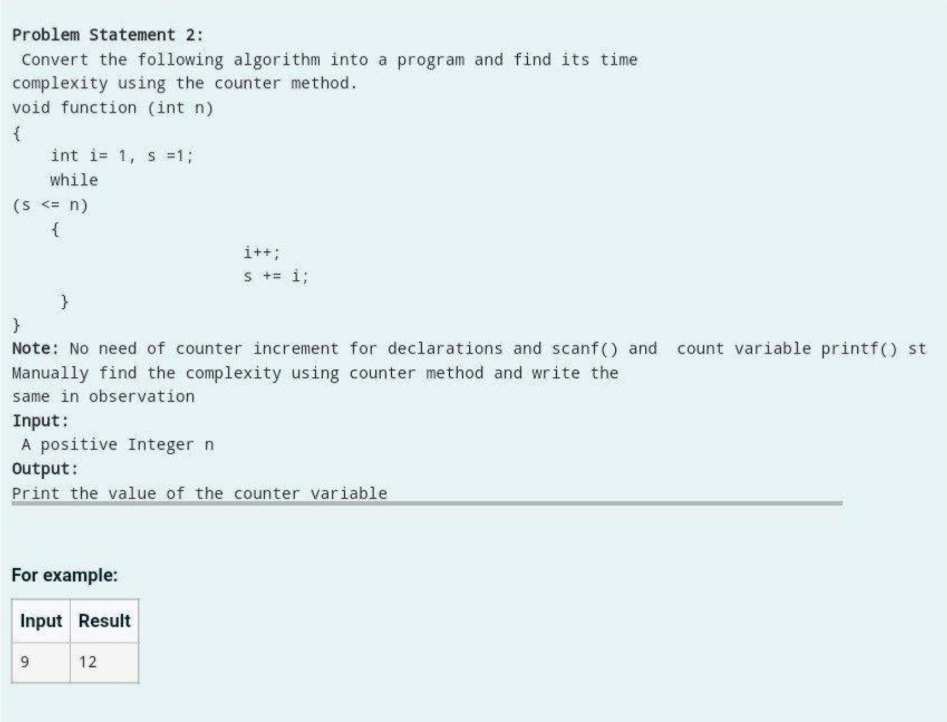
PROGRAM:

```
#include <math.h>

#include <stdio.h>


int main()

 {

   int num, originalNum, remainder, n = 0;

   float result = 0.0;


   printf("Enter an integer: ");
```

```
  scanf("%d", &num);

  originalNum = num;
  for (originalNum = num; originalNum != 0; ++n)
 {
     originalNum /= 10;
  }

  for (originalNum = num; originalNum != 0; originalNum /= 10)
{
     remainder = originalNum % 10;
    result += pow(remainder, n);
  }
  if ((int)result == num)
   printf("%d is an Armstrong number.", num);
  else
   printf("%d is not an Armstrong number.", num);
  return 0;
}
```

2)TIME COMPLEXITY

i)



Problem Statement 2:

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void function (int n)
{
    int i= 1, s =1;
    while
(s <= n)
    {
                    i++;
                    s += i;
     }
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() st

Manually find the complexity using counter method and write the same in observation

Input:

A positive Integer n

Output:

Print the value of the counter variable

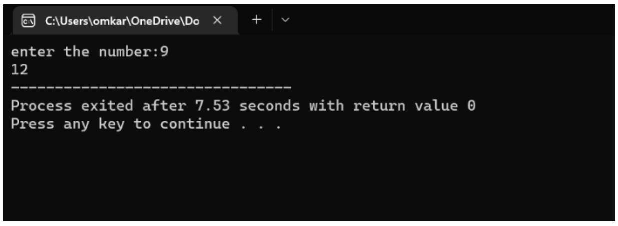
For example:

Input	Result
9	12

```
#include<stdio.h>
void function(int min);
int main()
{
        int n;
        printf("enter the number:");
        scanf("%d",&n);
        function(n);
        return 0;
}
void function(int n)
{
        int count=0;
        int i=1,s=1;
        count++;
        count++;
```

```
    while(s<=n)
    {
        count++;
        i++;
        count++;
        s+=i;
        count++;

    }
    count++;
    printf("%d",count);
}
```

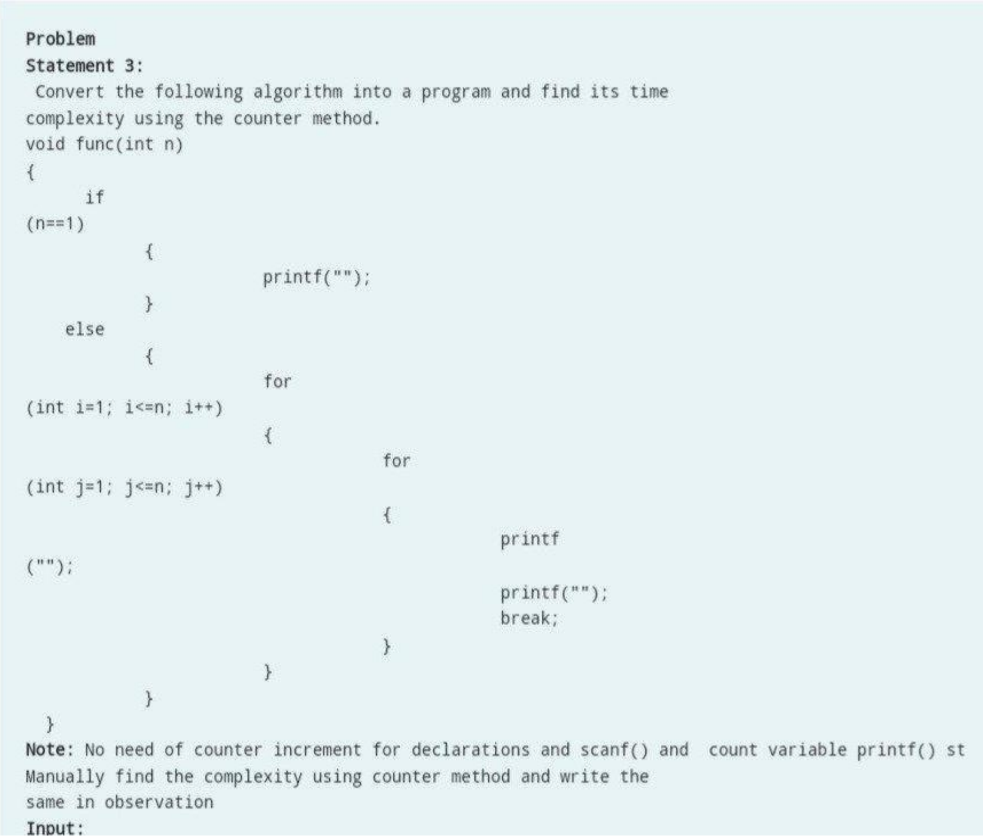
OUTPUT:



```
enter the number:9
12
--------------------------------
Process exited after 7.53 seconds with return value 0
Press any key to continue . . .
```

:

II)



Problem
Statement 3:
Convert the following algorithm into a program and find its time complexity using the counter method.

```
void func(int n)
{
      if
(n==1)
            {
                        printf("");
            }
    else
            {
                        for
(int i=1; i<=n; i++)
                        {
                                    for
(int j=1; j<=n; j++)
                                    {
                                                printf
("");
                                                printf("");
                                                break;
                                    }
                        }
            }
  }
```

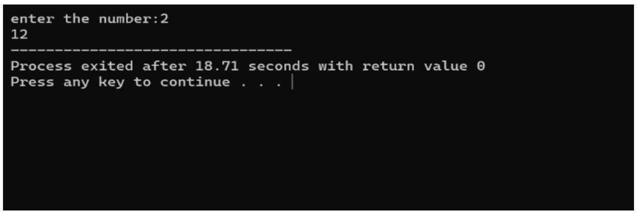
Note: No need of counter increment for declarations and scanf() and count variable printf() st
Manually find the complexity using counter method and write the same in observation
Input:

Program:

```
#include <stdio.h>
 void function(int n);
 int main()
{

int n;
printf("enter the number:");
scanf("%d",&n);
function(n);
 return 0;
}

void function(int n)
{
```

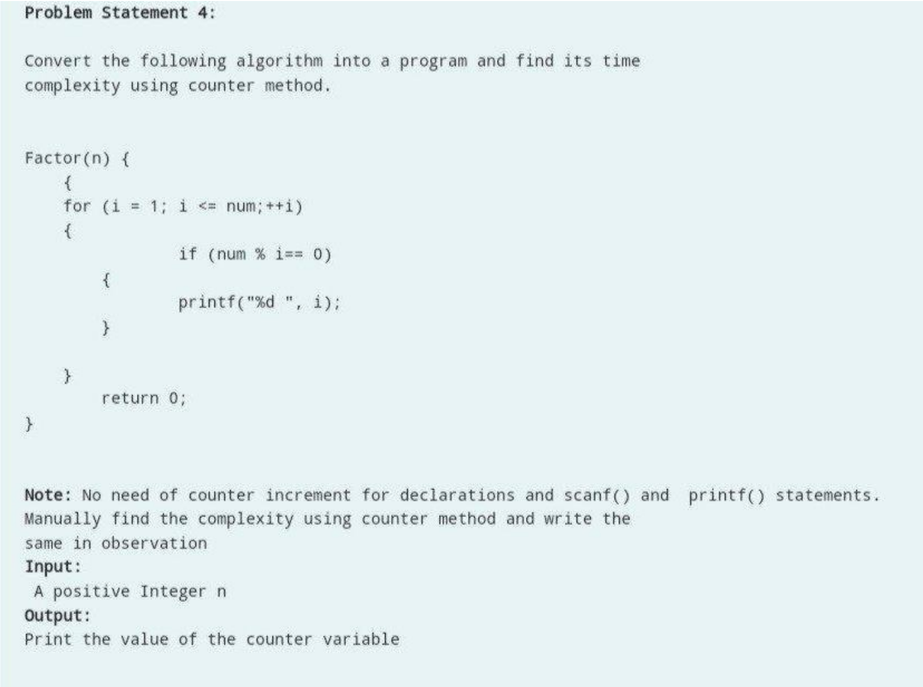
```
int count=0;
 if(n==1)
{

count++;
 count++;
}

else
{

count++;
for(int i=1;i<=n;i++)
{

count++;
for(int j=1;j<=n;j++)
{

count++;
 count++;
 count++;
  count++;
   break;
}
}count++;
}


printf("%d",count);
}
```

OUTPUT:



```
enter the number:2
12
--------------------------------
Process exited after 18.71 seconds with return value 0
Press any key to continue . . .
```

iii)



Problem Statement 4:

Convert the following algorithm into a program and find its time complexity using counter method.

```
Factor(n) {
    {
    for (i = 1; i <= num;++i)
    {
                if (num % i== 0)
        {
                printf("%d ", i);
        }

    }
        return 0;
}
```

Note: No need of counter increment for declarations and scanf() and printf() statements. Manually find the complexity using counter method and write the same in observation

Input:

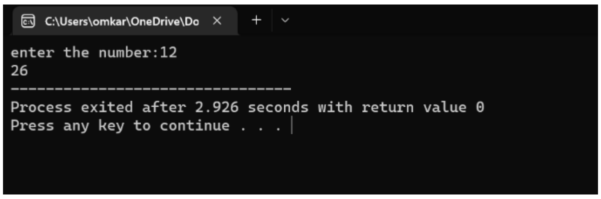
 A positive Integer n

Output:

Print the value of the counter variable

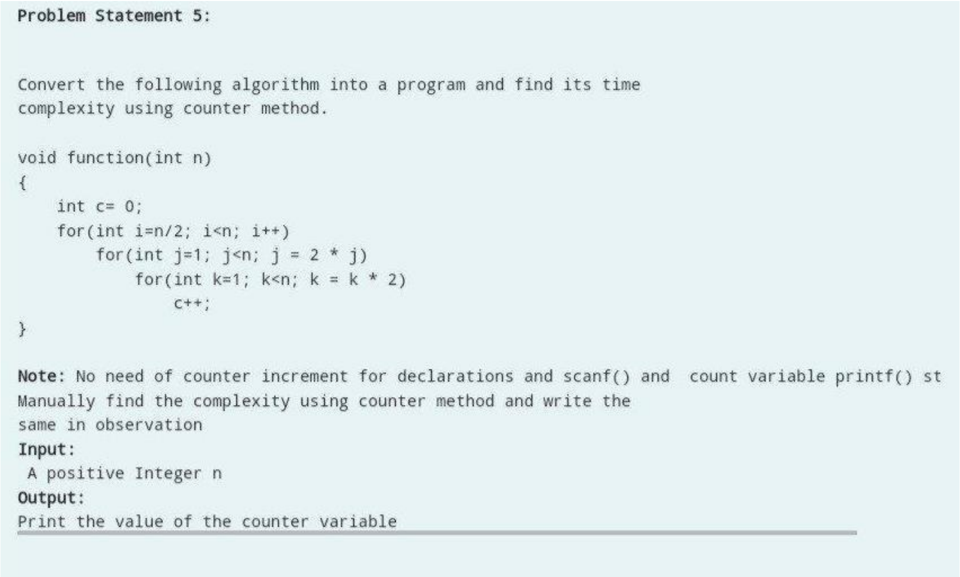
```
#include <stdio.h>
 int factor(int n);
  int count=0;
int main()
{

int n;
printf("enter the number:");
 scanf("%d",&n);
  factor(n);
   printf("%d",count);
    return 0;
}

int factor(int n)
{

int i;
 count++;
for(i=1;i<=n;++i)
{

count++;
if(n%i==0)
{

}count++;
}
count++;
 return 0;
}
```

OUTPUT:



```
enter the number:12
26
--------------------------------
Process exited after 2.926 seconds with return value 0
Press any key to continue . . .
```

iv)



Problem Statement 5:

Convert the following algorithm into a program and find its time complexity using counter method.

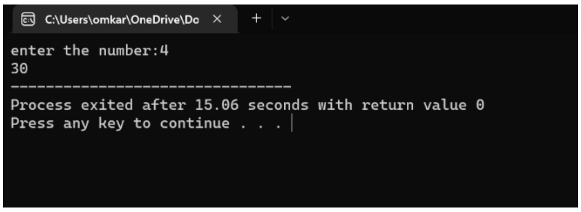
```
void function(int n)
{
    int c= 0;
    for(int i=n/2; i<n; i++)
        for(int j=1; j<n; j = 2 * j)
            for(int k=1; k<n; k = k * 2)
                c++;
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() st
Manually find the complexity using counter method and write the same in observation
Input:
 A positive Integer n
Output:
Print the value of the counter variable

```
#include <stdio.h>
 void function(int n);
  int main()
{


int n;
 printf("enter the number:");
scanf("%d",&n);
 function(n);
```

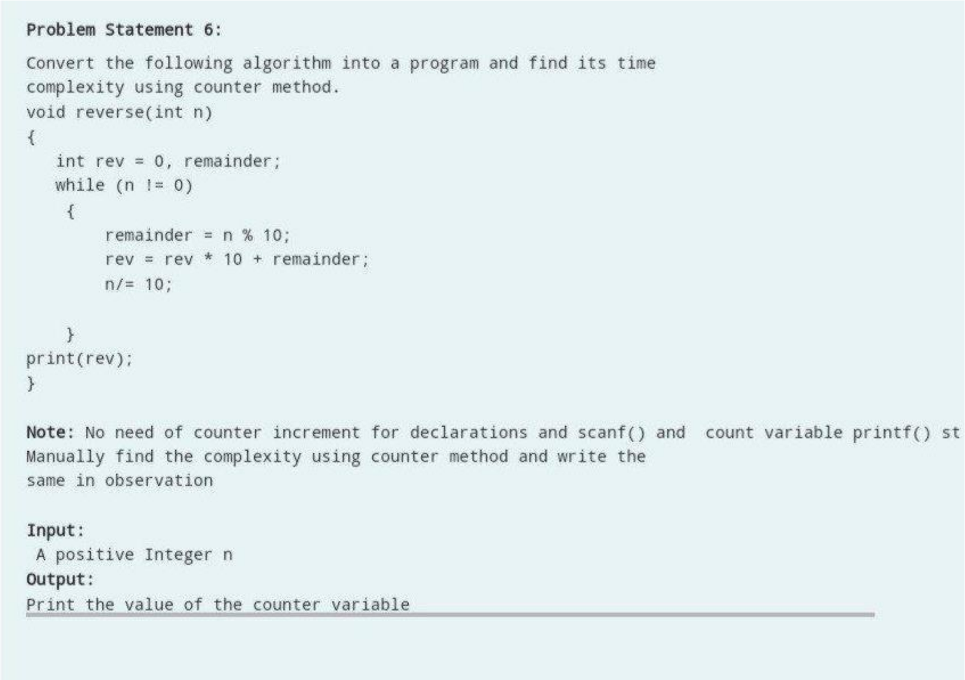
```
  return 0;
}

void function(int n)
{

int count=0;
 int c=0;
  count++;
for(int i=n/2;i<n;i++)
{

count++;
for(int j=1;j<n;j=2*j)
{

count++;
for(int k=1;k<n;k=k*2)
{

count++;
 c++;
count++;
}
count++;
}
count++;
}
count++;
 printf("%d",count);
}
```

OUTPUT:



```
enter the number:4
30
--------------------------------
Process exited after 15.06 seconds with return value 0
Press any key to continue . . .
```

v)



Problem Statement 6:

Convert the following algorithm into a program and find its time complexity using counter method.

```
void reverse(int n)
{
   int rev = 0, remainder;
   while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n/= 10;

    }
print(rev);
}
```

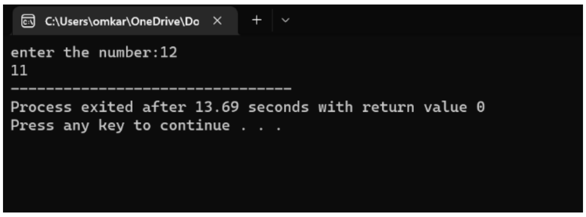
Note: No need of counter increment for declarations and scanf() and count variable printf() st
Manually find the complexity using counter method and write the same in observation

Input:
 A positive Integer n
Output:
Print the value of the counter variable

```
#include <stdio.h>
 void reverse(int n);


int main()
{
```

```
int n;
printf("enter the number:");
 scanf("%d",&n);
  reverse(n);
   return 0;
}

void reverse(int n)
{

int count=0;
int rev=0,remainder; count++;
 while(n!=0)
{

count++;
 remainder=n%10;
  count++;
   rev=rev*10+remainder;
   count++;
n=n/10;
 count++;
}
 count++;
 count++;
printf("%d",count);
}
```

OUTPUT:



```
enter the number:12
11
--------------------------------
Process exited after 13.69 seconds with return value 0
Press any key to continue . . .
```

3)BINARY SEARCH AND TIME COMPLEXITY

```
#include <stdio.h>
int main()
{
int i, low, high, mid, n, key, array[100];
int count=0;
printf("Enter number of elements");
scanf("%d",&n);
printf("Enter %d integers", n);
for(i = 0; i < n; i++)
scanf("%d",&array[i]);
printf("Enter value to find");
scanf("%d", &key);
low = 0;
count++;
high = n - 1;
count++;

while (low <= high) {
        count++;
if(array[mid] < key)
low = mid + 1;
else if (array[mid] == key) {
count++;
printf("%d found at location %d ", key, mid+1);
break;
}
else
high = mid - 1;
mid = (low + high)/2;
count++;
}
if(low > high)
printf("Not found! %d isn't present in the list.n", key);
printf("time complexity : %d",count);
return 0;
}
#include<stdio.h>
int main()
```

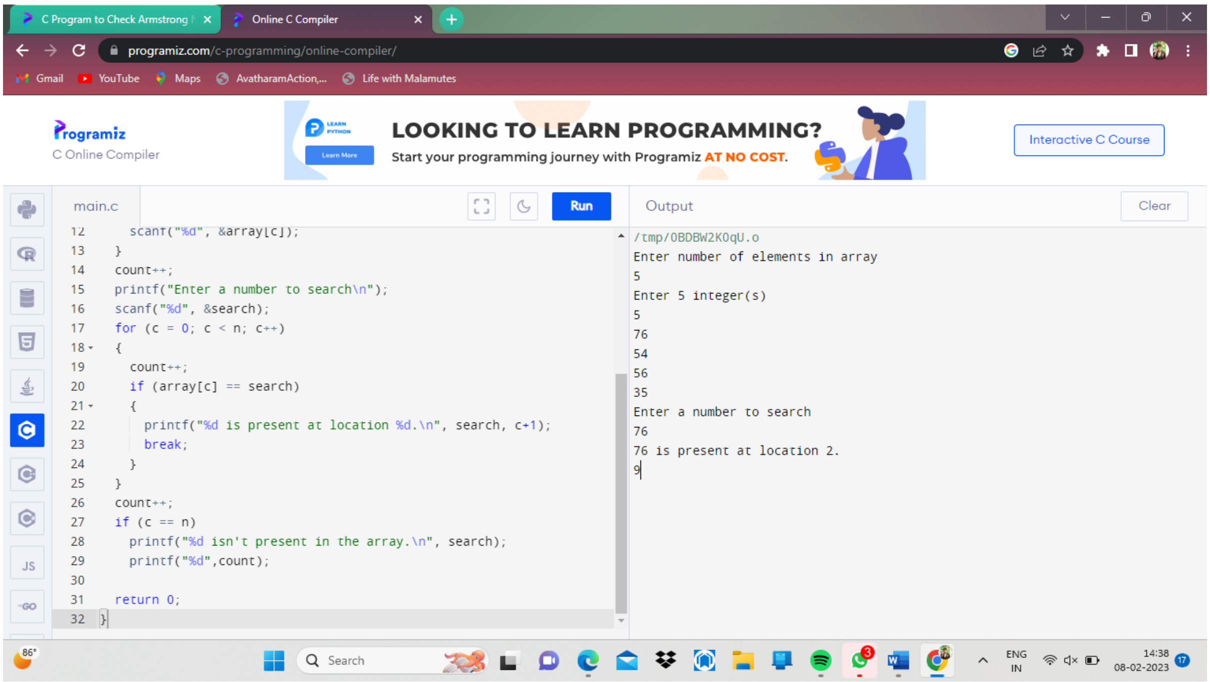
```
{
 int array[100], search, c, n;
 int count=0;
 printf("Enter number of elements in array\n");
 scanf("%d", &n);
 printf("Enter %d integer(s)\n", n);
 for (c = 0; c < n; c++)
 {
     count++;
  scanf("%d", &array[c]);
 }
 count++;
 printf("Enter a number to search\n");
 scanf("%d", &search);
 for (c = 0; c < n; c++)
 {
     count++;
  if (array[c] == search)
  {
   printf("%d is present at location %d.\n", search, c+1);
   break;
  }
 }
```

```
  count++;
  if (c == n)
    printf("%d isn't present in the array.\n", search);
    printf("%d",count);

  return 0;
}
```

4)LINEAR SEARCH

PROGRAM:

```
#include<stdio.h>
int main()
{
  int array[100], search, c, n;
  int count=0;
  printf("Enter number of elements in array\n");
  scanf("%d", &n);
  printf("Enter %d integer(s)\n", n);
  for (c = 0; c < n; c++)
  {
        count++;
    scanf("%d", &array[c]);
  }
  count++;
  printf("Enter a number to search\n");
  scanf("%d", &search);
  for (c = 0; c < n; c++)
  {
        count++;
    if (array[c] == search)
    {
      printf("%d is present at location %d.\n", search, c+1);
      break;
    }
  }
  count++;
  if (c == n)
    printf("%d isn't present in the array.\n", search);
    printf("%d",count);

  return 0;
}
```



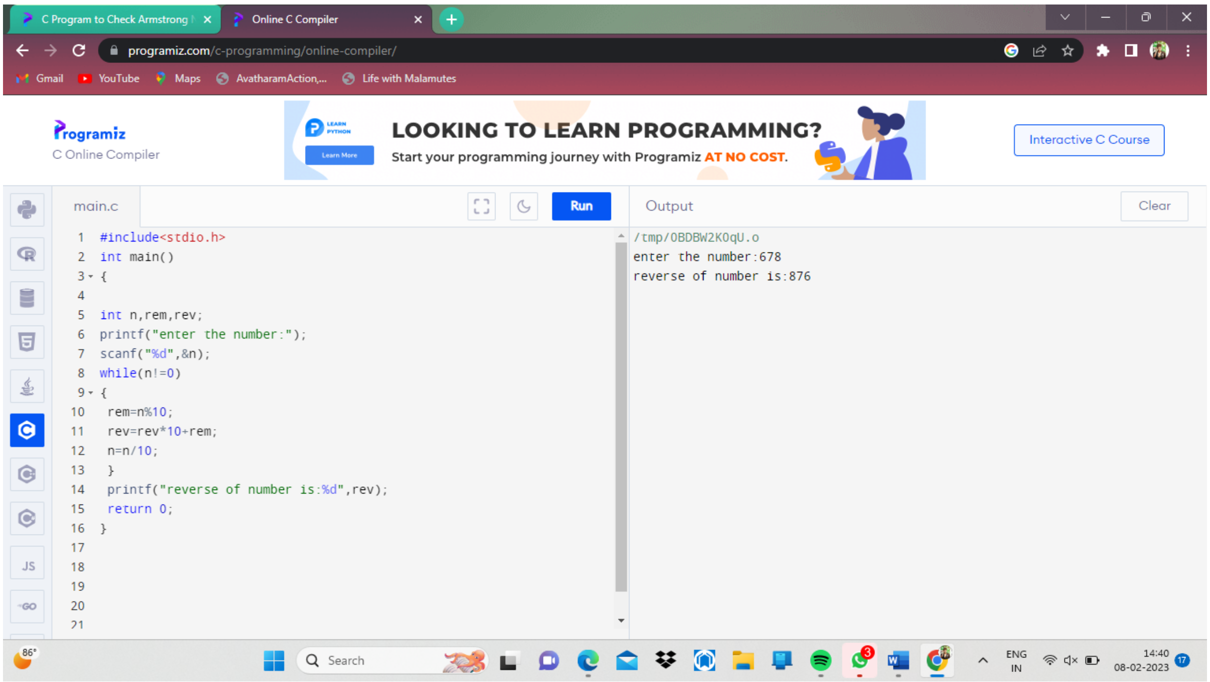
```
    scanf("%d", &array[c]);
  }
  count++;
  printf("Enter a number to search\n");
  scanf("%d", &search);
  for (c = 0; c < n; c++)
  {
    count++;
    if (array[c] == search)
    {
      printf("%d is present at location %d.\n", search, c+1);
      break;
    }
  }
  count++;
  if (c == n)
    printf("%d isn't present in the array.\n", search);
    printf("%d",count);

  return 0;
}
```

```
/tmp/OBDBW2K0qU.o
Enter number of elements in array
5
Enter 5 integer(s)
5
76
54
56
35
Enter a number to search
76
76 is present at location 2.
9
```

5)REVERSE OF NUMBER

5)REVERSE OF A STRING

```
#include<stdio.h>
int main()
{
int n,rem,rev;
printf("enter the number:");
scanf("%d",&n);
while(n!=0)
{
 rem=n%10;
 rev=rev*10+rem;
 n=n/10;
 }
 printf("reverse of number is:%d",rev);
 return 0;
}
```



```
#include<stdio.h>
int main()
{

int n,rem,rev;
printf("enter the number:");
scanf("%d",&n);
while(n!=0)
{
 rem=n%10;
 rev=rev*10+rem;
 n=n/10;
 }
 printf("reverse of number is:%d",rev);
 return 0;
 }
```

```
/tmp/0BDBW2K0qU.o
enter the number:678
reverse of number is:876
```

6) C PROGRAM TO PERFORM STRASSEN’S MATRIX MULTIPLICATION

PROGRAM:

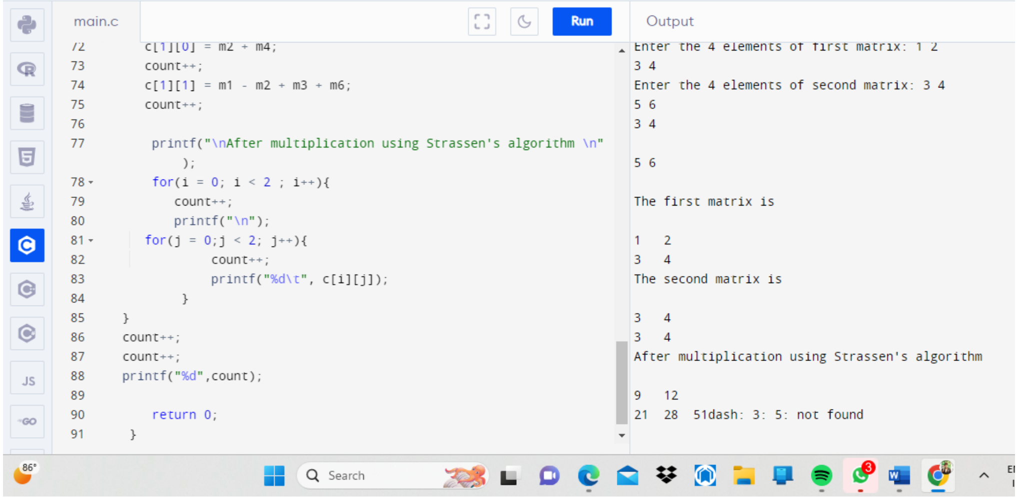
```
#include<stdio.h>
int main()
{
        int a[2][2], b[2][2], c[2][2], i, j;
        int m1, m2, m3, m4 , m5, m6, m7;
        int count=0;
```

```
  printf("Enter the 4 elements of first matrix: ");
    count++;
    for(i = 0;i < 2; i++)
    {
          count++;
          for(j = 0;j < 2; j++){
                     count++;
             scanf("%d", &a[i][j]);
         }
}
count++;
count++;

  printf("Enter the 4 elements of second matrix: ");
    for(i = 0; i < 2; i++){
            count++;
        for(j = 0;j < 2; j++){
             count++;
             scanf("%d", &b[i][j]);
         }
     }
     count++;
```

```
        printf("\nThe first matrix is\n");
for(i = 0; i < 2; i++){
            count++;
            printf("\n");
            for(j = 0; j < 2; j++){
                 count++;
                 printf("%d\t", a[i][j]);
          }
        }
        count++;
        count++;

  printf("\nThe second matrix is\n");
        for(i = 0;i < 2; i++){
                count++;
            printf("\n");
            for(j = 0;j < 2; j++){
             count++;
         printf("%d\t", b[i][j]);
     }
}
count++;
count++;
```

```
m1= (a[0][0] + a[1][1]) * (b[0][0] + b[1][1]);
count++;
m2= (a[1][0] + a[1][1]) * b[0][0];
count++;
m3= a[0][0] * (b[0][1] - b[1][1]);
count++;
m4= a[1][1] * (b[1][0] - b[0][0]);
count++;
m5= (a[0][0] + a[0][1]) * b[1][1];
count++;
m6= (a[1][0] - a[0][0]) * (b[0][0]+b[0][1]);
count++;
m7= (a[0][1] - a[1][1]) * (b[1][0]+b[1][1]);
count++;

c[0][0] = m1 + m4- m5 + m7;
count++;
        c[0][1] = m3 + m5;
        count++;
        c[1][0] = m2 + m4;
        count++;
        c[1][1] = m1 - m2 + m3 + m6;
        count++;
```

```
          printf("\nAfter multiplication using Strassen's
algorithm \n");
          for(i = 0; i < 2 ; i++){
               count++;
            printf("\n");
      for(j = 0;j < 2; j++){
               count++;
                  printf("%d\t", c[i][j]);
               }
   }
   count++;
   count++;
   printf("%d",count);


          return 0;
        }
```

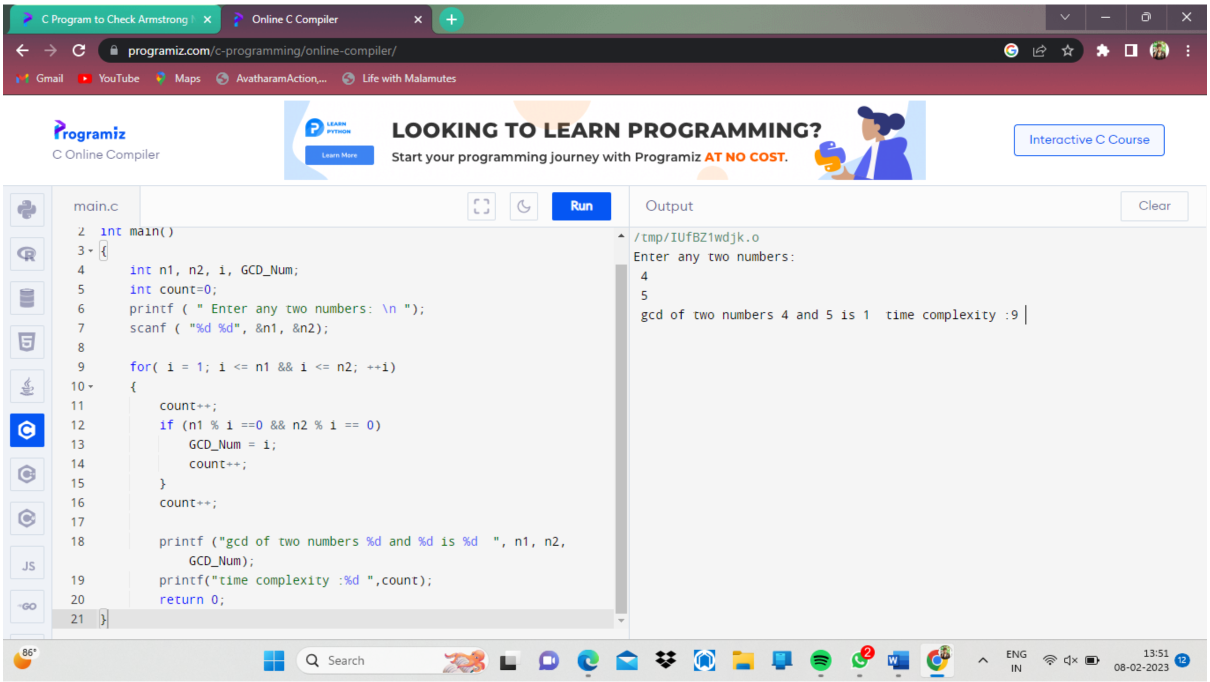


```
        c[1][0] = m2 + m4;
        count++;
        c[1][1] = m1 - m2 + m3 + m6;
        count++;

         printf("\nAfter multiplication using Strassen's algorithm \n"
            );
         for(i = 0; i < 2 ; i++){
            count++;
            printf("\n");
        for(j = 0;j < 2; j++){
                 count++;
                 printf("%d\t", c[i][j]);
             }
     }
     count++;
     count++;
     printf("%d",count);

         return 0;
      }
```

```
Enter the 4 elements of first matrix: 1 2
3 4
Enter the 4 elements of second matrix: 3 4
5 6
3 4

5 6

The first matrix is

1   2
3   4
The second matrix is

3   4
3   4
After multiplication using Strassen's algorithm

9   12
21  28  51dash: 3: 5: not found
```

8)find the gcd of two numbers with time complexity

```
#include <stdio.h>
int main()
{
    int n1, n2, i, GCD_Num;
    int count=0;
    printf ( " Enter any two numbers: \n ");
    scanf ( "%d %d", &n1, &n2);

    for( i = 1; i <= n1 && i <= n2; ++i)
    {
        count++;
        if (n1 % i ==0 && n2 % i == 0)
            GCD_Num = i;
            count++;
        }
            count++;

        printf ("gcd of two numbers %d and %d is %d  ", n1, n2,
GCD_Num);
        printf("time complexity :%d ",count);
        return 0;
}
```

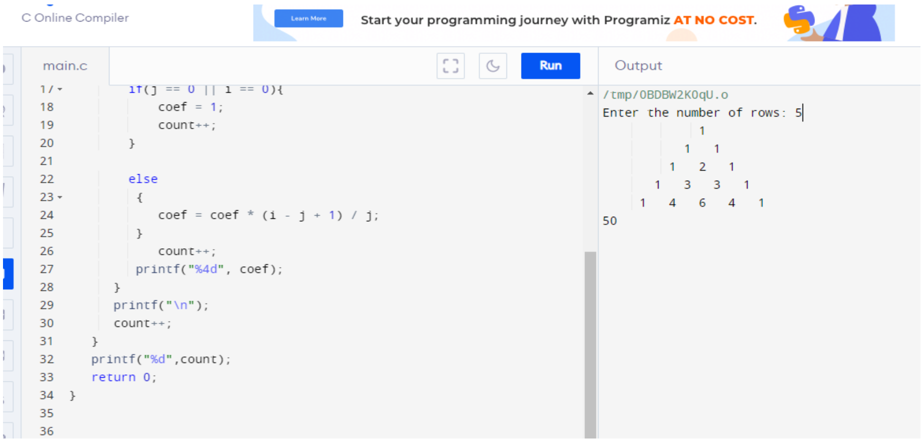


```
int main()
{
    int n1, n2, i, GCD_Num;
    int count=0;
    printf ( " Enter any two numbers: \n ");
    scanf ( "%d %d", &n1, &n2);

    for( i = 1; i <= n1 && i <= n2; ++i)
    {
        count++;
        if (n1 % i ==0 && n2 % i == 0)
            GCD_Num = i;
            count++;
        }
        count++;

        printf ("gcd of two numbers %d and %d is %d  ", n1, n2,
            GCD_Num);
        printf("time complexity :%d ",count);
        return 0;
}
```

```
/tmp/IUfBZ1wdjk.o
Enter any two numbers:
 4
 5
 gcd of two numbers 4 and 5 is 1  time complexity :9
```

9)Generate a program for pascal triangle

```
#include<stdio.h>
int main()
{
   int rows, coef = 1, space, i, j;
   int count=0;
   printf("Enter the number of rows: ");
   scanf("%d", &rows);
   for (i = 0; i<rows; i++)
   {
          count++;
      for (space = 1; space <= rows - i; space++)
         printf("  ");
         count++;
      for (j = 0; j <= i; j++)
          {
                 count++;
        if(j == 0 || i == 0){
            coef = 1;
            count++;
        }

        else
         {
            coef = coef * (i - j + 1) / j;
         }
            count++;
         printf("%4d", coef);
      }
      printf("\n");
      count++;
   }
   printf("%d",count);
   return 0;
}
```



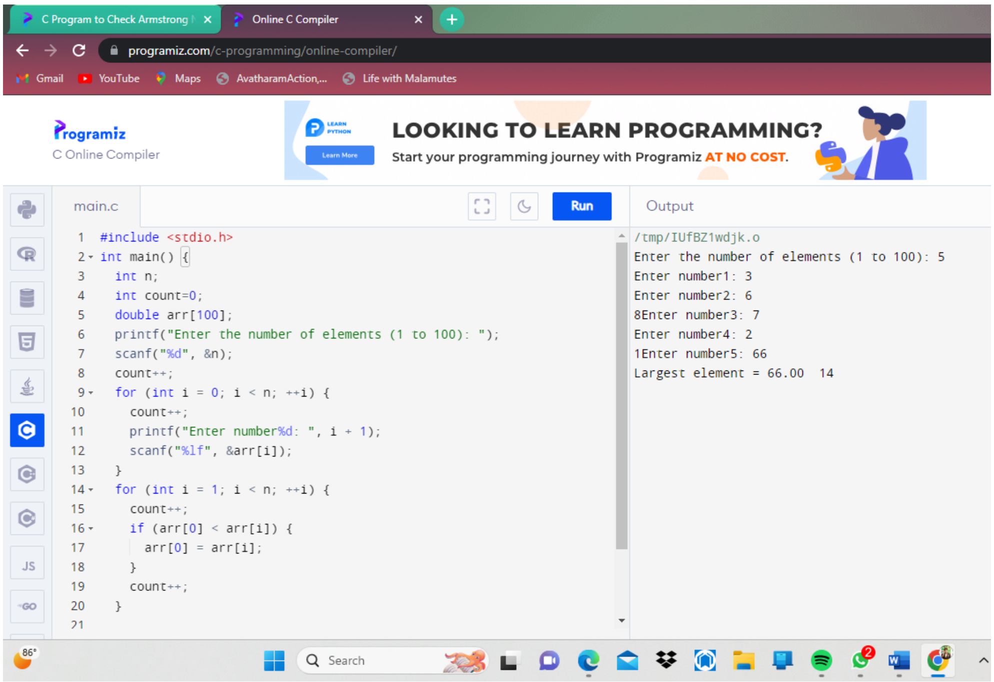
```
        if(j == 0 || i == 0){
            coef = 1;
            count++;
        }

        else
         {
            coef = coef * (i - j + 1) / j;
         }
            count++;
         printf("%4d", coef);
      }
      printf("\n");
      count++;
   }
   printf("%d",count);
   return 0;
}
```

```
/tmp/OBDBW2K0qU.o
Enter the number of rows: 5
                1
             1     1
          1     2     1
       1     3     3     1
    1     4     6     4     1
50
```

10)largest elements in the array with time complexity

Program:

```
#include <stdio.h>
int main() {
  int n;
  int count=0;
  double arr[100];
  printf("Enter the number of elements (1 to 100): ");
  scanf("%d", &n);
  count++;
  for (int i = 0; i < n; ++i) {
        count++;
    printf("Enter number%d: ", i + 1);
    scanf("%lf", &arr[i]);
  }
  for (int i = 1; i < n; ++i) {
      count++;
    if (arr[0] < arr[i]) {
      arr[0] = arr[i];
    }
    count++;
  }
  printf("Largest element = %.2lf  ", arr[0]);
 printf("%d",count);
 return 0;
}
```



C Program to Check Armstrong
Online C Compiler
programiz.com/c-programming/online-compiler/
Gmail
YouTube
Maps
AvatharamAction,...
Life with Malamutes
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Learn More
main.c
Run
Output
Search

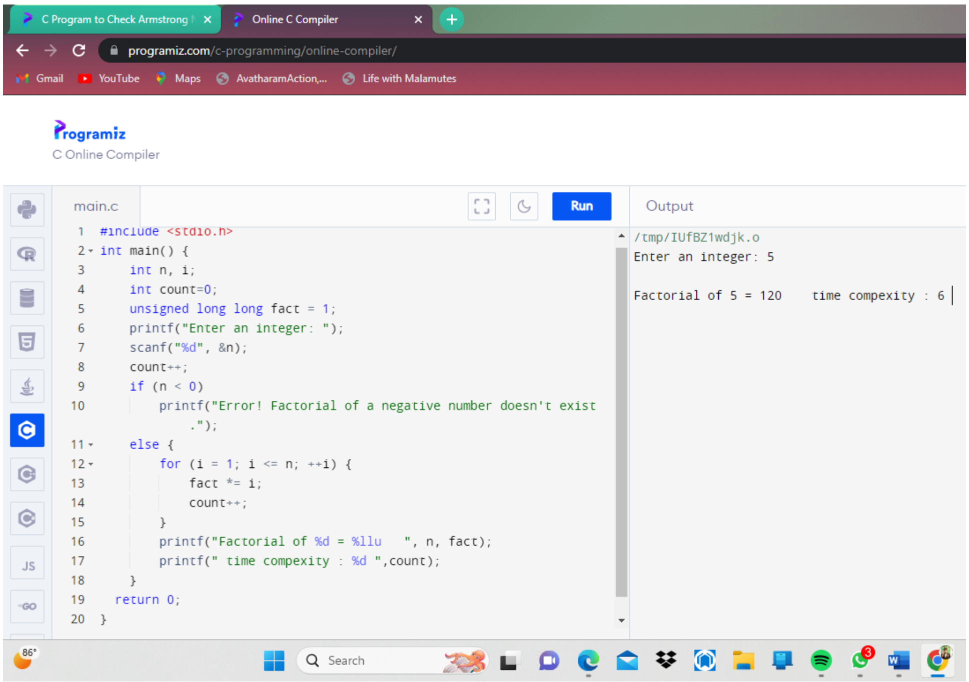
```
#include <stdio.h>
int main() {
  int n;
  int count=0;
  double arr[100];
  printf("Enter the number of elements (1 to 100): ");
  scanf("%d", &n);
  count++;
  for (int i = 0; i < n; ++i) {
    count++;
    printf("Enter number%d: ", i + 1);
    scanf("%lf", &arr[i]);
  }
  for (int i = 1; i < n; ++i) {
    count++;
    if (arr[0] < arr[i]) {
      arr[0] = arr[i];
    }
    count++;
  }
```

```
/tmp/IUfBZ1wdjk.o
Enter the number of elements (1 to 100): 5
Enter number1: 3
Enter number2: 6
8Enter number3: 7
Enter number4: 2
1Enter number5: 66
Largest element = 66.00  14
```

11)Write a program to find the factorial (fact)of a number and to estimate time complexity. [Condition such as i. n=0, return 1   otherwise fact (n-1) * n]

PROGRAM:

```
#include <stdio.h>
int main() {
    int n, i;
    int count=0;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);
    count++;
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else {
        for (i = 1; i <= n; ++i) {
            fact *= i;
            count++;
        }
        printf("Factorial of %d = %llu   ", n, fact);
        printf(" time compexity : %d ",count);
    }
  return 0;
}
```



```
#include <stdio.h>
int main() {
    int n, i;
    int count=0;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);
    count++;
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist
            .");
    else {
        for (i = 1; i <= n; ++i) {
            fact *= i;
            count++;
        }
        printf("Factorial of %d = %llu   ", n, fact);
        printf(" time compexity : %d ",count);
    }
  return 0;
}
```

```
/tmp/IUfBZ1wdjk.o
Enter an integer: 5

Factorial of 5 = 120    time compexity : 6
```

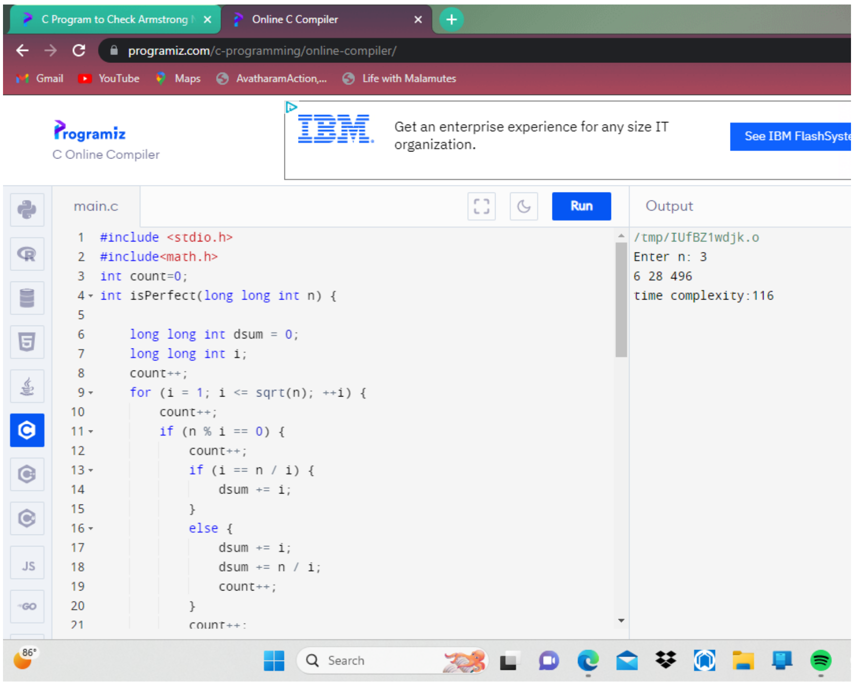
12) Write a program to print the first n perfect numbers. (Hint Perfect number means a positive integer that is equal to the sum of its proper divisors)

PROGRAM:

```
#include <stdio.h>

#include<math.h>

int count=0;

int isPerfect(long long int n) {


    long long int dsum = 0;

    long long int i;

    count++;

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

        count++;

        if (n % i == 0) {
```

```
      count++;
            if (i == n / i) {
                dsum += i;
            }
            else {
                dsum += i;
                dsum += n / i;
                count++;
            }
            count++;
        }
        count++;
    }
    count++;
  dsum = dsum - n;
  count++;
    if (dsum == n) return 1;
    else    return 0;
}
int isPrime(long long int n) {

    if (n == 1)
        return 0;
    for (int i = 2; i <= sqrt(n); ++i) {
        count++;
        if (n % i == 0)
            return 0;
    }
    return 1;
  count++;
}
int main() {
    long long int n, i, temp;
    printf("Enter n: ");
    scanf("%d", &n);
  count++;
    i = 1;
    while (n > 0) {
    count++;
        if (isPrime(i) == 1) {
            temp = pow(2, i - 1) * (pow(2, i) - 1);
            count++;
            if (isPerfect(temp) == 1) {
                printf("%d ", temp);
                n = n - 1;
```

```
                        count++;
                    }
                }
                i = i + 1;
                count++;
            }
            printf("\n");
        printf("%d",count);
}
```



```
#include <stdio.h>
#include<math.h>
int count=0;
int isPerfect(long long int n) {

    long long int dsum = 0;
    long long int i;
    count++;
    for (i = 1; i <= sqrt(n); ++i) {
        count++;
        if (n % i == 0) {
            count++;
            if (i == n / i) {
                dsum += i;
            }
            else {
                dsum += i;
                dsum += n / i;
                count++;
            }
            count++;
```

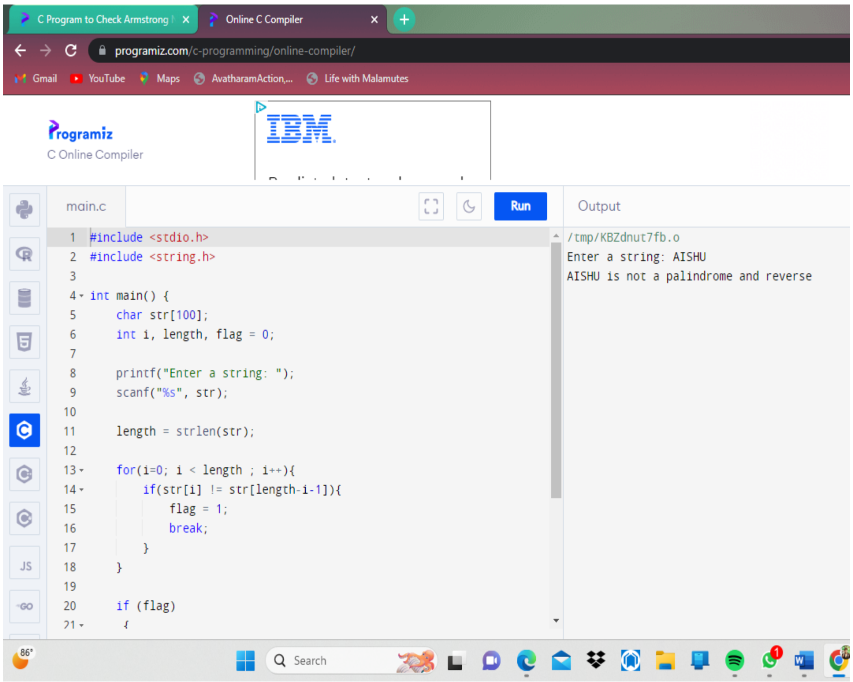
```
/tmp/IUfBZ1wdjk.o
Enter n: 3
6 28 496
time complexity:116
```

13)Write a C program to check whether is a given input is a palindrome.

PROGRAM:

```
#include <stdio.h>
```

```
#include <string.h>

int main() {
    char str[100];
    int i, length, flag = 0;

    printf("Enter a string: ");
    scanf("%s", str);

    length = strlen(str);

    for(i=0; i < length ; i++){
        if(str[i] != str[length-i-1]){
            flag = 1;
            break;
        }
    }

    if (flag)
         {
        printf("%s is not a palindrome and reverse\n", str);
    }
        else
        {
        printf("%s is a palindrome\n", str);
    }

    return 0;
}
```



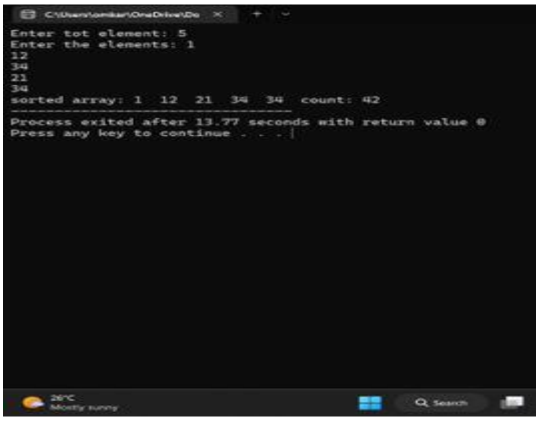
```
#include <stdio.h>
#include <string.h>

int main() {
    char str[100];
    int i, length, flag = 0;

    printf("Enter a string: ");
    scanf("%s", str);

    length = strlen(str);

    for(i=0; i < length ; i++){
        if(str[i] != str[length-i-1]){
            flag = 1;
            break;
        }
    }

    if (flag)
     {
```

```
/tmp/KBZdnut7fb.o
Enter a string: AISHU
AISHU is not a palindrome and reverse
```

14) Write a program to perform Bubble sort and estimate time Complexity

PROGRAM:

```
#include<stdio.h>
int main()
{
    int a,count=0;
    printf("Enter no.of element: ");
scanf("%d",&a);

    int arr[ele];
    printf("Enter the elements: ");

    for (int i = 0; i < a; i++){
        count++;
        scanf("%d",&arr[i]);
    }count++;



    for (int i = 0; i < a; i++)
{
count++;
        for (int j =i+1; j < a; j++)
        {
count++;
         if (arr[i]>arr[j])
         {
            count++;
            int temp=arr[i];
            count++;
            arr[i]=arr[j];
            count++;
            arr[j]=temp;
            count++;
         }
        }count++;

    }count++;

    printf("sorted array: ");
    for (int i = 0; i < a; i++)
    {count++;
        count++;
        printf("%d  ",arr[i]);
    }count++;
    printf("count: %d",count);
}
```

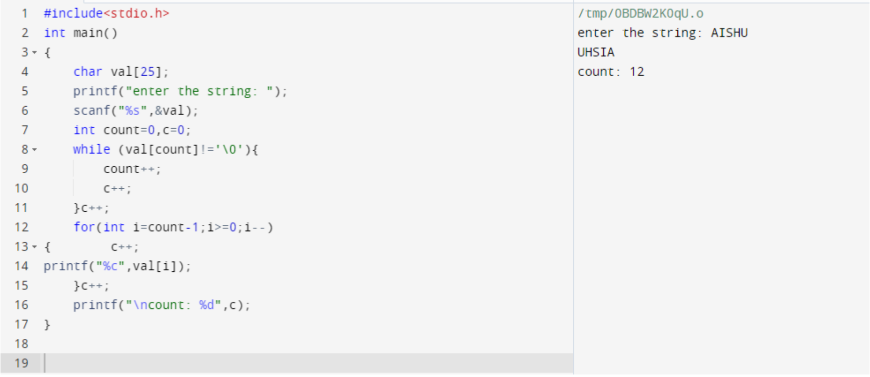


```
Enter tot element: 5
Enter the elements: 1
12
34
21
34
sorted array: 1  12  21  34  34  count: 42
--------------------------------
Process exited after 13.77 seconds with return value 0
Press any key to continue . . .
```

15) REVERSE OF A STRING

PROGRAM:

```
#include<stdio.h>
int main()
{
    char val[25];
    printf("enter the string: ");
    scanf("%s",&val);
    int count=0,c=0;
    while (val[count]!='\0'){
        count++;
        c++;
    }c++;
    for(int i=count-1;i>=0;i--)
{      c++;
printf("%c",val[i]);
    }c++;
    printf("\ncount: %d",c);
}
```



```
#include<stdio.h>
int main()
{
    char val[25];
    printf("enter the string: ");
    scanf("%s",&val);
    int count=0,c=0;
    while (val[count]!='\0'){
        count++;
        c++;
    }c++;
    for(int i=count-1;i>=0;i--)
{        c++;
printf("%c",val[i]);
    }c++;
    printf("\ncount: %d",c);
}
```

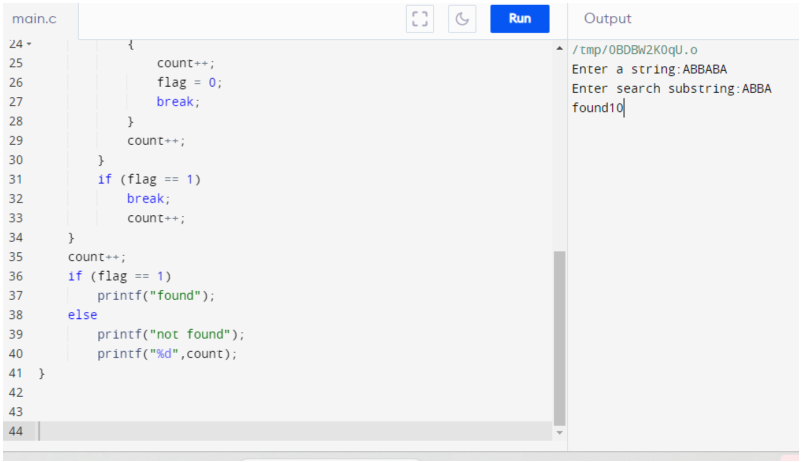
```
/tmp/0BDBW2K0qU.o
enter the string: AISHU
UHSIA
count: 12
```

16) SUB STRING IS THERE IN A STRING OR NOT

PROGRAM:

```
#include<stdio.h>


int main()
{
    char str[80], search[10];
    int count1 = 0, count2 = 0, i, j, flag;
    int count=0;
    printf("Enter a string:");
    gets(str);
    printf("Enter search substring:");
    gets(search);
    while (str[count1] != '\0')
    count1++;
while (search[count2] != '\0')
    count2++;
for (i = 0; i <= count1 - count2; i++)
{
    count++;
    for (j = i; j < i + count2; j++)
    {
            count++;
        flag = 1;
        if (str[j] != search[j - i])
        {
            count++;
            flag = 0;
            break;
        }
        count++;
    }
    if (flag == 1)
        break;
        count++;
}
count++;
```

```
    if (flag == 1)
        printf("found");
    else
        printf("not found");
        printf("%d",count);
}
```



```c
            {
                count++;
                flag = 0;
                break;
            }
            count++;
        }
        if (flag == 1)
            break;
            count++;
    }
    count++;
    if (flag == 1)
        printf("found");
    else
        printf("not found");
        printf("%d",count);
}
```

```
/tmp/0BDBW2K0qU.o
Enter a string:ABBABA
Enter search substring:ABBA
found10
```