

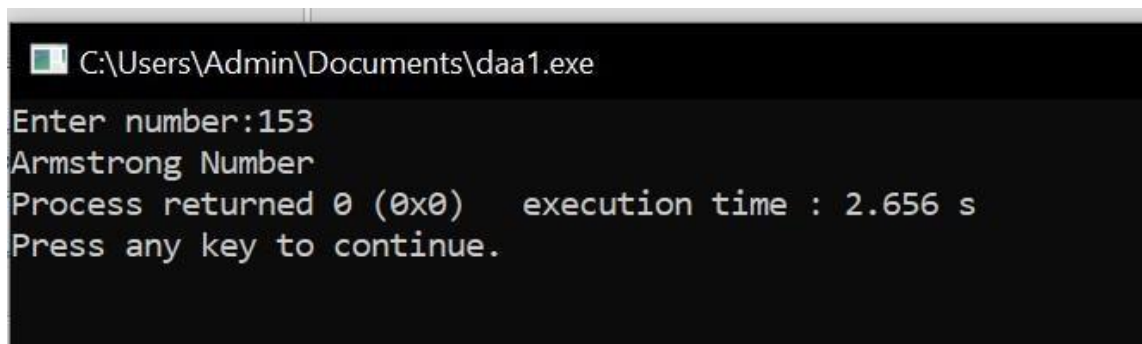
UBA0672 – DAA [DAY – 1]

Shaik .Ahamad Alisha
192125106

1.Armstrong number

Program:

```
#include<stdio.h> int main()
{   int n,n1,sum=0,r;
printf("Enter number:");
scanf("%d",&n);   n1=n;
    while (n1>0)
    {
        r=n1%10;
sum=sum+(r*r*r);
n1=n1/10;
    }
    if(sum==n)
    {
        printf("Armstrong Number");
    }   else
    {
        printf("Not Armstrong Number");
    }
}
```



```
C:\Users\Admin\Documents\daa1.exe
Enter number:153
Armstrong Number
Process returned 0 (0x0)   execution time : 2.656 s
Press any key to continue.
```

Time complexity 5 programs

2. Program :

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

Program:

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

```

{   int n;
    scanf("%d",&n);
function(n);   return
0;
}

void function(int n)
{   int c=0;   int
i=1,s=1;   c++;
c++;
    while(s<=n)
    {       c++;
i++;
c++;
s+=i;
c++;   }   c++;
    printf("Time Complexity : %d",c);
}

```

```

C:\Users\Admin\Documents\daa2.exe
12
Time Complexity : 15
Process returned 0 (0x0)   execution time : 4.510 s
Press any key to continue.

```

3 .Program :

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:

A positive Integer n

Output:

Print the value of the counter variable

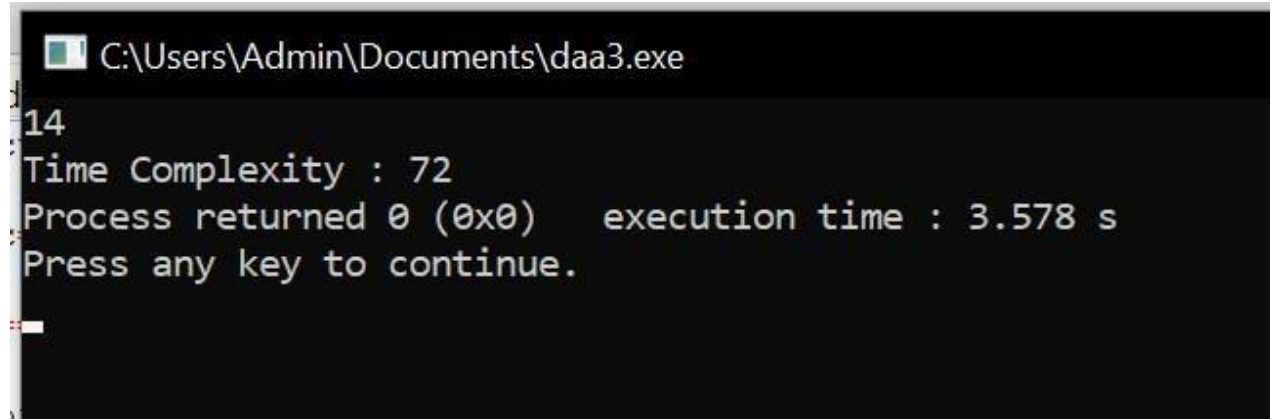
Program:

```
#include<stdio.h> int main()
{   int n;
    scanf("%d",&n);   function(n);
return 0;
}
void function(int n)
```

```

{   int c=0;
c++;
    if(n==1)
    {
printf("");
c++;   }   else
    {
        for(int i=1;i<=n;i++)
        {           c++;
            for(int j=1;j<=1;j++)
            {           c++;
printf("");c++;
printf("");c++;
                break;
            }           c++;
        }           c++;
    }
printf("Time Complexity : %d",c); }

```



```

C:\Users\Admin\Documents\daa3.exe
14
Time Complexity : 72
Process returned 0 (0x0)   execution time : 3.578 s
Press any key to continue.

```

4 .Program :

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

Program:

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

```
int main() {    int n;
```

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

```
Factor(n);    return
```

```
0;
```

```
}
```

```
int Factor(int n)
```

```
{ int c=0;
```

```
    int i=0;
```

```
    c++;
```

```
    for(int
```

```
    i=1;i<=
    n;i++)
    {      c++;      c++;
if(n%i==0)
    {
        //printf("%d\n",i);

    } }  c++;  printf("Time
Complexity : %d",c);  return 0;
}
```

```
C:\Users\Admin\Documents\daa4.exe
10
Time Complexity : 22
Process returned 0 (0x0)   execution time : 1.964 s
Press any key to continue.
```

5. Program:

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

Program:

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

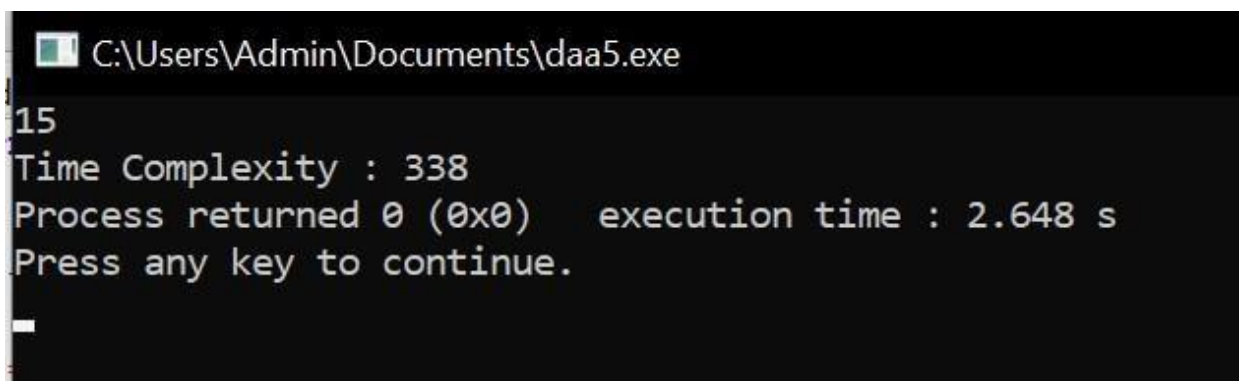
```
int main() {    int n;
```

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



```
function(n); return
0;
}
```

```
void function(int n) {
int c=0,cn=0;
cn++; for(int
i=n/2;i<n;i++) {
cn++; for(int
j=1;j<n;j=2*j)
{ cn++; for(int
k=1;k<n;k=k*2)
{
cn++; c++;
cn++;
} cn++;
} cn++;
} cn++; printf("Time Complexity
: %d",cn);
}
```



```
C:\Users\Admin\Documents\daa5.exe
15
Time Complexity : 338
Process returned 0 (0x0) execution time : 2.648 s
Press any key to continue.
```

6 . Program :

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

Program:

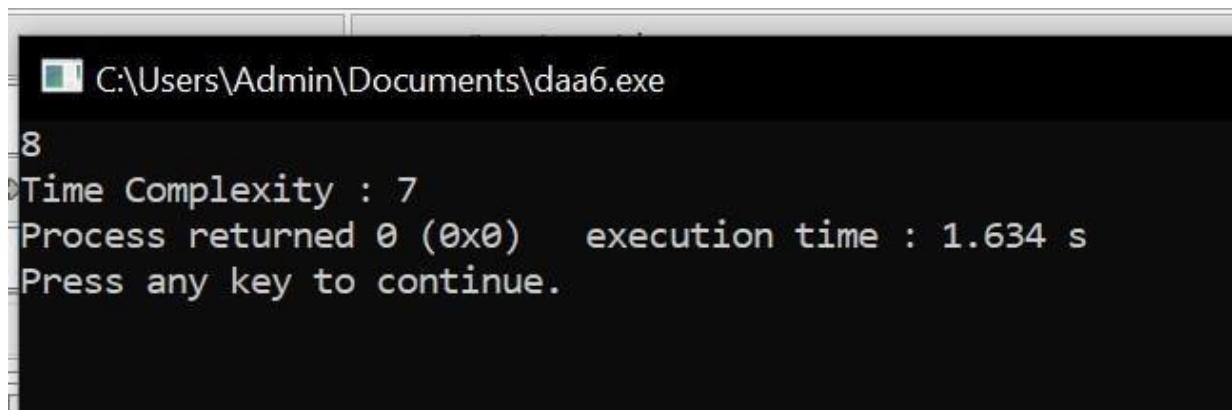
```
#include<stdio.h> int main()
```

```
{   int n;
    scanf("%d",&n);
reverse(n);   return
0;
}
```

```
void reverse(int n)
```

```
{ int c=0;
  int
```

```
    rev=0,r
    emaind
    er;
c++; while(n!=0)
    {
        c++;
remainder=n%10;    c++;
rev=rev*10+remainder;    c++;
n/=10;    c++;
    }    c++;
    //printf("%d",rev);    c++;
printf("Time Complexity : %d",c);
}
```



```
C:\Users\Admin\Documents\daa6.exe
8
Time Complexity : 7
Process returned 0 (0x0) execution time : 1.634 s
Press any key to continue.
```

7. Write a program to search a number in a list using binary search and estimate time complexity

Program:

```
#include<stdio.h> int main() { int c=0;
int n,k,i,j,f=0,a[50]; c++;
printf("Enter number of elements:");
scanf("%d",&n); printf("Enter
elements:\n"); for(i=0;i<n;i++)
{ c++;
scanf("%d",&a[i]);
} c++; printf("Enter Element to
search:"); scanf("%d",&k);
for(i=0;i<n;i++)
{ c++;
c++; if(k==a[i])
{
printf("Element is found at index %d\n",i); f=1;
c++;
} } }
```

```

c++; c++;
if(f==0)
{
    printf("Element is not found");
}
printf("\nTime Complexity : %d",c); }

```

```

C:\Users\Admin\Documents\daa7-Ls.exe
Enter number of elements:5
Enter elements:
2
7
5
9
8
Enter Element to search:9
Element is found at index 3
Time Complexity : 20
Process returned 0 (0x0)   execution time : 15.739 s
Press any key to continue.

```

8. Write a program to search a number in a list using linear search and estimate time complexity

Program:

```

#include<stdio.h>
int
main() {    int
c=0;
    int n,k,i,low,high,mid,a[50],temp;

```

```

printf("Enter number of elements:");
scanf("%d",&n);  printf("Enter
elements:\n");  for(i=0;i<n;i++)
    {      c++;
scanf("%d",&a[i]);  }
c++;  printf("Enter
Element to search:");
scanf("%d",&k);
low=0;  c++;
high=n-1;  c++;
mid=low+high/2;  c++;
    c++;
    while(low<=high)
    {      c++;
c++;
        if(a[mid]<k)
        {
            low=mid+1;  c++;
        }
        else if(a[mid]==k)
        {
            printf("\nElement is found at index %d\n",mid);          break;
        }
    else
    {
        high=mid-1;  c++;
    }
    mid=(low+high)/2;  c++;
    }  c++;  c++;
if(low>high)
{
    printf("Element is not found\n");
}
printf("\nTime Complexity : %d\n",c);
}

```

```
C:\Users\Admin\Documents\daa8-Bs.exe
Enter number of elements:6
Enter elements:
2
3
5
7
8
9
Enter Element to search:7

Element is found at index 3

Time Complexity : 23

Process returned 0 (0x0)   execution time : 6.250 s
Press any key to continue.
```

9. Write a program to find the reverse of a given number.

Program:

```
#include<stdio.h> int main()
{   int c=0;   int n,r,rev=0;
c++;   printf("Enter
number:");
scanf("%d",&n);   c++;
while (n!=0)
{
```

```

        r=n%10;    c++;
rev=(rev*10)+r;    c++;    n=n/10;
c++;
    }    c++;    printf("Reverse Number :
%d",rev);    printf("\nTime Complexity
: %d\n",c); }

```

```

C:\Users\Admin\Documents\daa9-reverse.exe
Enter number:12345
Reverse Number : 54321
Time Complexity : 18

Process returned 0 (0x0)   execution time : 1.764 s
Press any key to continue.

```

10. Write a C program to perform Strassen's Matrix Multiplication for the 2*2 matrix elements and Estimate time complexity.

Program:

```

#include<stdio.h> int main() {    int
a[2][2],b[2][2],c[2][2],i,j;    printf("Enter
elements of matrix A:\n");
for(i=0;i<=1;i++)
    {
for(j=0;j<=1;j++)
    {
        scanf("%d",&a[i][j]);
    }
}
    printf("Enter elements of matrix B:\n");
for(i=0;i<=1;i++)

```



```

    {
for(j=0;j<=1;j++)
    {
        scanf("%d",&b[i][j]);
    }
}

c[0][0]=(a[0][0]*b[0][0])+(a[0][1]*b[1][0]);
c[0][1]=(a[0][0]*b[0][1])+(a[0][1]*b[1][1]);
c[1][0]=(a[1][0]*b[0][0])+(a[1][1]*b[1][0]);
c[1][1]=(a[1][0]*b[0][1])+(a[1][1]*b[1][1]);    printf("Matrix
C : \n");    for(i=0;i<=1;i++)
    {
for(j=0;j<=1;j++)
    {
printf("%d\t",c[i][j]);
    }
printf("\n");
    }
}

```

```
"C:\Users\Admin\Documents\daa10-Mat Mul.exe"
Enter elements of matrix A:
1      2
3      4
Enter elements of matrix B:
1      2
3      4
Matrix C :
7      10
15     22

Process returned 0 (0x0)    execution time : 8.717 s
Press any key to continue.
_
```

CSA0672 – DAA – DAY 2

Shaik.Ahamad Alisha

192125106

11. Write a program to generate all the reverse of a prime should be prime

(for example 907 is prime and reverse 709 is also prime)

Generate all the no's upto N and estimate time complexity.

Program:

```
#include<stdio.h> int main()
{   int c=0;   int
n,n1,f,i,j,k,r,p[100],f1;   int
sum=0,b=0,rev=0;   c++;
c++; c++;   printf("Enter
number:");   scanf("%d",&n);
for(j=3;j<=n;j++)
    {       c++;       f=0;
c++;
for(i=2;i<j;i++)
    {       c++;
c++;
if(j%i==0)
    {
        f=f+1; c++;
    }
c++;       c++;
if(f==0)   {
n1=j;       c++;
```

```

rev=0;
    c++;
while (n1!=0)
    {
        c++;
r=n1%10; c++;      rev=(rev*10)+r;
c++;      n1=n1/10; c++;
    }      c++;      f1=0;

c++;
for(k=2;k<rev;k++)
    {
        c++;

c++;
if(rev%k==0)
    {
        f1++;

c++;
    }
}

c++;
c++;
if(f1==0)
    {
        printf("%d\n",j);
    }

    }    c++;    printf("Time
Complexity : %d",c); }

```

```
"C:\Users\Admin\Documents\daa11-Reverse prime.exe"
Enter number:100
3
5
7
11
13
17
31
37
71
73
79
97
Time Complexity : 12920
Process returned 0 (0x0)   execution time : 4.608 s
Press any key to continue.
```

12 . Compute the program to find the GCD of two numbers. And also find the finf of time Recursion used to estimate time complexity.

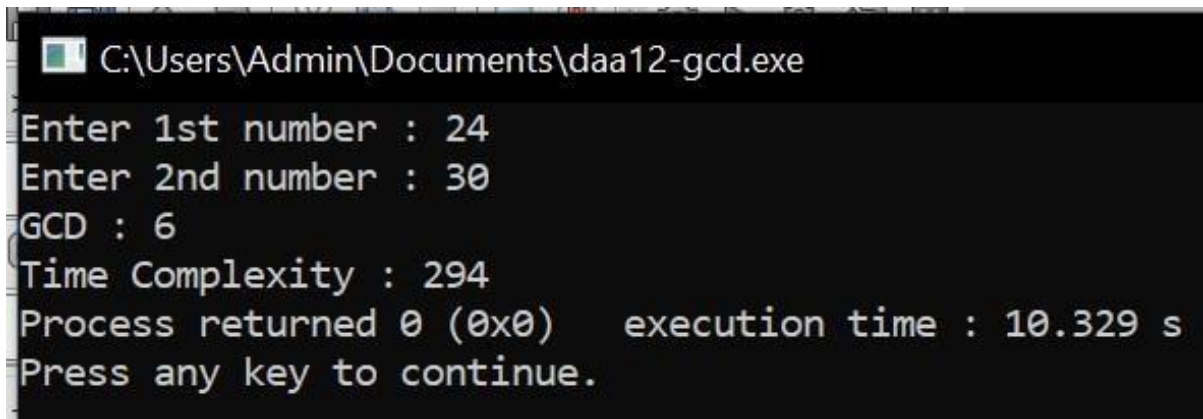
Program:

```
#include<stdio.h> int main() {   int c=0;   int
a,b,af[100],bf[100],cf[100],a1,b1,c1,i,j,g;
printf("Enter      1st   number      :   ");
scanf("%d",&a);   printf("Enter 2nd number : ");
scanf("%d",&b);   a1=-1; c++;
for(i=1;i<=a;i++)
    {       c++;       c++;
if(a%i==0)
```

```

        {          a1=a1+1;
c++;          af[a1]=i;
c++;
        }
    }  c++;  b1=1;
c++;
for(i=1;i<=b;i++)
{
c++;
c++;
if(b%i==0)
{
        b1=b1+1;
c++;          bf[b1]=i;
c++;
    }  }  c++;  c1=-1;
c++;  for(i=0;i<a1+1;i++)
{
    c++;
for(j=0;j<b1+1;j++)
{
        c++;
c++;          if(af[i]==bf[j])
{
            g=af[i];  c++;
        }  }  c++;  }
c++;  printf("GCD : %d\n",g);
printf("Time Complexity : %d",c); }

```



A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\Admin\Documents\daa12-gcd.exe". The command prompt displays the following text: "Enter 1st number : 24", "Enter 2nd number : 30", "GCD : 6", "Time Complexity : 294", "Process returned 0 (0x0) execution time : 10.329 s", and "Press any key to continue.".

```
C:\Users\Admin\Documents\daa12-gcd.exe
Enter 1st number : 24
Enter 2nd number : 30
GCD : 6
Time Complexity : 294
Process returned 0 (0x0) execution time : 10.329 s
Press any key to continue.
```

**13. Generate a program for Pascal triangle.
Estimate the time complexity for the row=5**

		1		1		
	1		2		1	
1		3		3	1	1

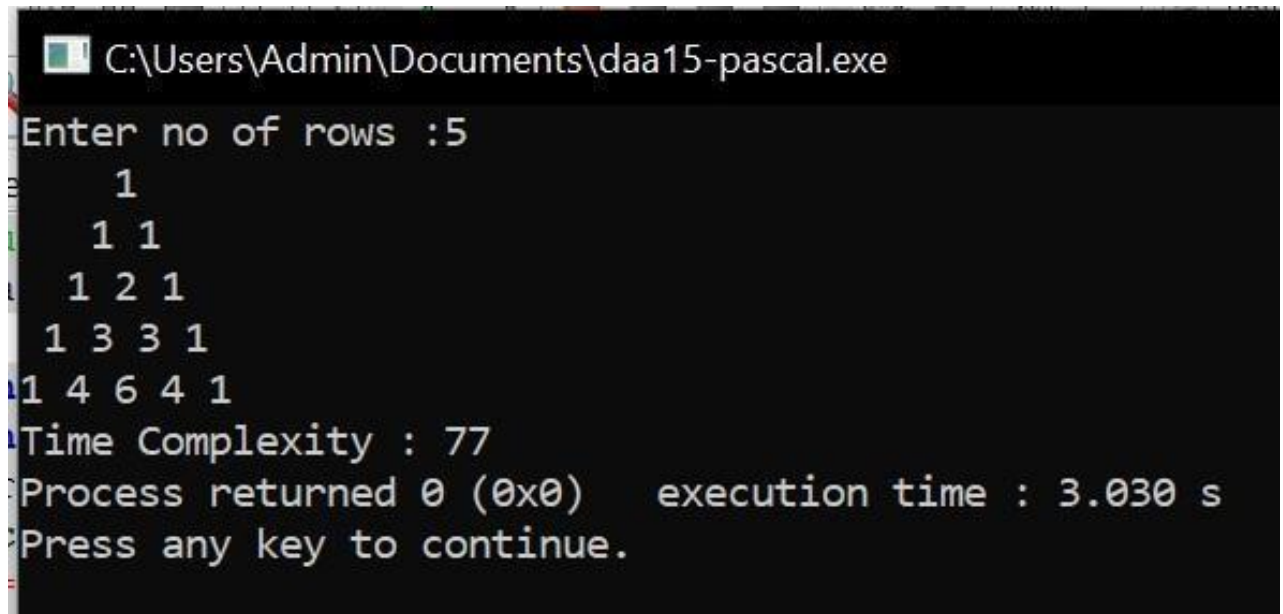
4 6 4 1

Program:

```
#include<stdio.h> int main() {   int
c=0;   int n,i,j,k,s,c1;
printf("Enter no of rows :");
scanf("%d",&n);   k=n;   c++;
for(i=0;i<n;i++)
    {       c++;       k=k-1;
c++;
for(s=0;s<k;s++)
    {       c++;
printf(" ");
    }       c++;
for(j=0;j<=i;j++)
    {
c++;
c++;
if(j==0)
    {       c1=1;
c++;
    }
else
    {       c1=c1*(ij+1)/j;
c++;
```

}

```
printf("%d ",c1);
    }    c++;    printf("\n");
}    c++;    printf("Time
Complexity : %d",c); }
```



```
C:\Users\Admin\Documents\daa15-pascal.exe
Enter no of rows :5
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
Time Complexity : 77
Process returned 0 (0x0) execution time : 3.030 s
Press any key to continue.
```

14. Write a program to find the largest element value in an array. Estimate the time complexity and no of comparison for the given set of values.

Program:

```
#include<stdio.h> int main() { int c=0;
int com=0,i,j,k,a[100],n; c++;
printf("Enter no of elements:");
scanf("%d",&n); printf("Enter elements
:\n"); for(i=0;i<n;i++)
{ c++;
scanf("%d",&a[i]);
```

```
} c++;
```

```

for(i=0;i<n;i++)
{
    c++;
    for(j=0;j<n;j++)
    {
        c++;
com++; c++;
    }
    if(a[i]>a[j])
    {
        k=a[i]; c++;
a[i]=a[j]; c++;
        a[j]=k;
c++;
    }
}
c++;

printf("Largest value : %d\n",a[0]);
printf("Comparisions : %d\n",com);
printf("Time complexity : %d\n",c); }

```

```

C:\Users\Admin\Documents\daa16-largestnum.exe
Enter no of elements:6
Enter elements :
3
8
5
9
2
11
Largest value : 11
Comparisions : 36
Time complexity : 165
Process returned 0 (0x0)   execution time : 13.408 s
Press any key to continue.

```

15. 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 $\text{fact}(n-1) * n$ Program:

```
#include<stdio.h> int fact(int n); int c=0; int
main() {   int n;   printf("Enter Number : ");
scanf("%d",&n);                               fact(n);
printf("Factorial      :      %d\n",fact(n));
printf("Time Complexity : %d\n",c);   return
0; } int fact(int n) {   int f;   c++;   if(n<=1)
{       f= 1;   c++;
}   else   {
        f= n*fact(n-1); c++;
}

return f;
}
```

C:\Users\Admin\Documents\daa17-fact.exe

Enter Number : 6

Factorial : 720

Time Complexity : 24

Process returned 0 (0x0) execution time : 1.755 s

Press any key to continue.

16. 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)

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496 Test Cases:

1. $N = 0$

2. **N = 5**

3. **N = -2**

4. **N = -5**

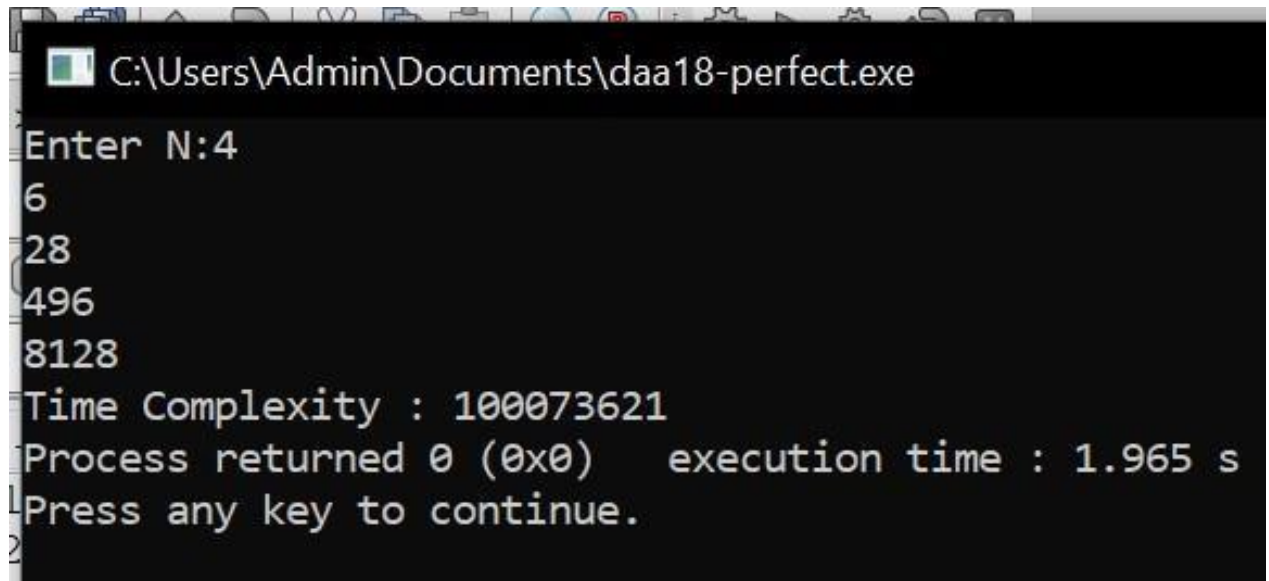
N = 0.2

Program:

```
#include<stdio.h> int main() {  
int c=0;    int i,j,sum,n,a[20],k=0;  
c++;    printf("Enter N:");  
scanf("%d",&n);    c++;  
if(n<1)    {    printf("Invalid  
Input");  
    }    else  
    {  
    for(i=6;i<10000;i++)  
    {    c++;    sum=0; c++;  
for(j=1;j<i-1;j++)  
    {    c++;  
c++;  
if(i%j==0)  
    {
```

```
sum=sum+j; c++;      }      }      c++;      c++;      if(i==sum)
```

```
        {          a[k++]=i; c++;  
        }      }c++;    for  
(i=0;i<n;i++)  
    {          c++;      printf("%d\n",a[i]);  
    }c++;  
}  
printf("Time Complexity : %d",c); }
```



```
C:\Users\Admin\Documents\daa18-perfect.exe  
Enter N:4  
6  
28  
496  
8128  
Time Complexity : 100073621  
Process returned 0 (0x0) execution time : 1.965 s  
Press any key to continue.
```

17. Write a C program to check whether is a given input is a palindrome
Program:

```
#include<stdio.h>  int main() {  int
c=0;  int n,r,rev=0,a;  c++;
printf("Enter  number:");
scanf("%d",&n);  a=n;  c++;  while
(n!=0)
{  c++;  r=n%10; c++;
rev=(rev*10)+r; c++;  n=n/10;
c++;
}  c++;  c++;
if(rev==a)
{
printf("Palindrome Number");
```

```
} else {
```



```
printf("

```

```
Not Palindr ome

```

```
Number

```

```
");

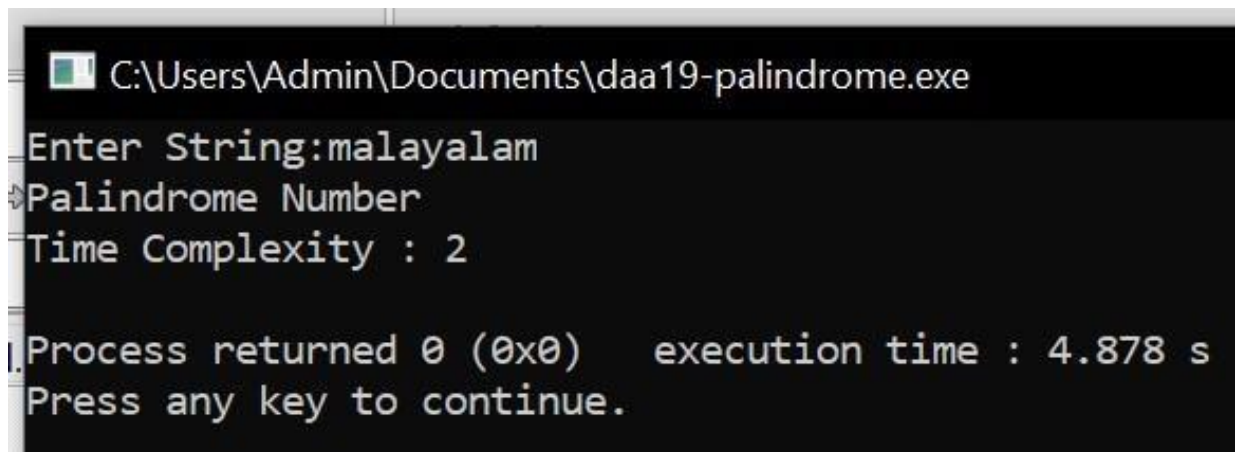
```

```

}

```

```
printf("\nTime Complexity : %d\n",c); }
```



```
C:\Users\Admin\Documents\daa19-palindrome.exe
Enter String:malayalam
Palindrome Number
Time Complexity : 2

Process returned 0 (0x0)   execution time : 4.878 s
Press any key to continue.
```

18. Write a program to perform Bubble sort and estimate time Complexity Program:

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

int com=0,i,j,k,a[100],n;    c++;

printf("Enter no of elements:");

scanf("%d",&n);    printf("Enter elements

:\n");    for(i=0;i<n;i++)

    {        c++;

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

        }    c++;

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

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

        {            c++;

com++; c++;            c++;

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

            {                k=a[i]; c++;

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

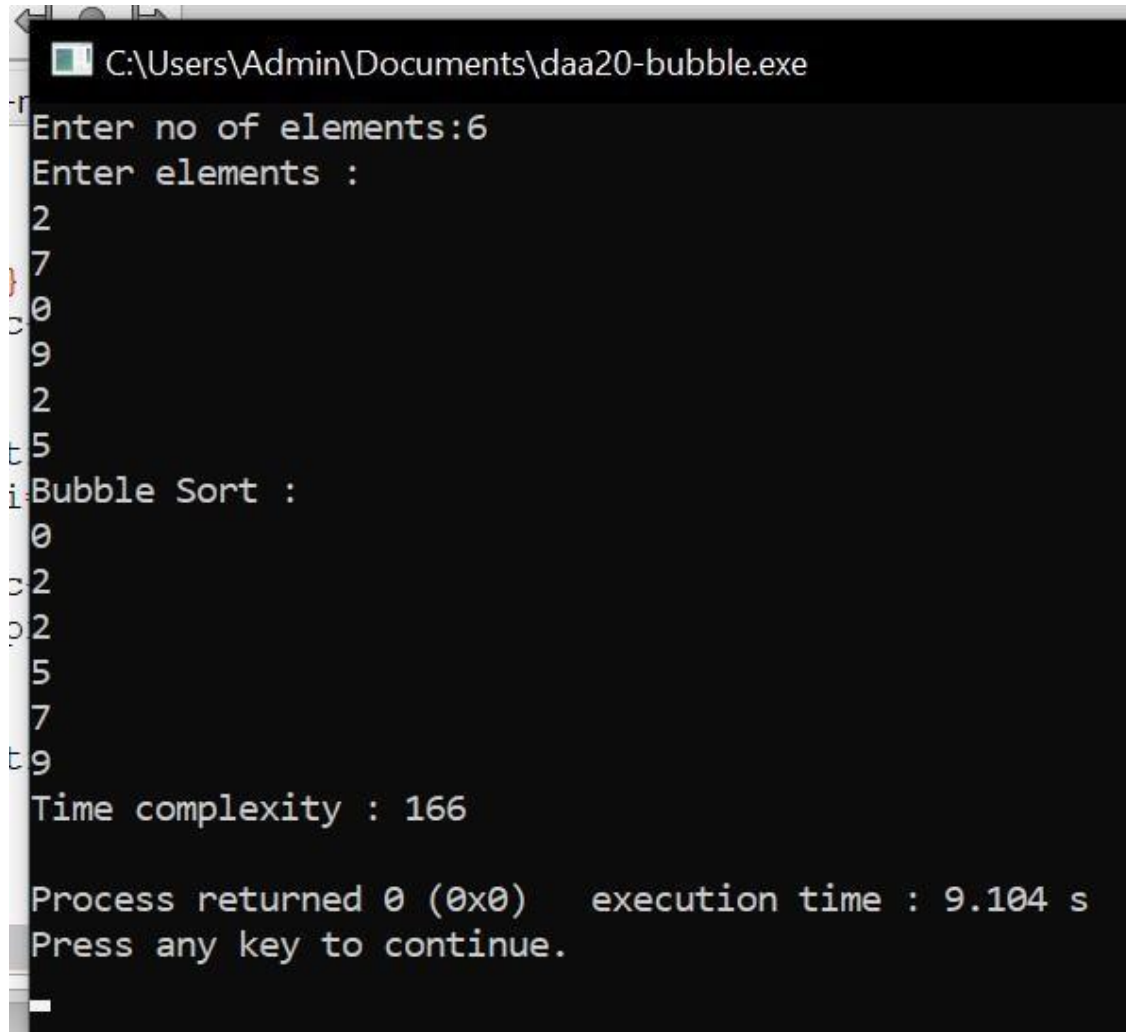
c++;

            }        }        c++;

    }    c++;    printf("Bubble
```

```
Sort :\n");    for(i=0;i<n;i++)
```

```
{    c++;  
printf("%d\n",a[i]);    }    c++;  
printf("Time complexity :  
%d\n",c);  
}
```



```
C:\Users\Admin\Documents\daa20-bubble.exe  
Enter no of elements:6  
Enter elements :  
2  
7  
0  
9  
2  
5  
Bubble Sort :  
0  
2  
2  
5  
7  
9  
Time complexity : 166  
  
Process returned 0 (0x0)    execution time : 9.104 s  
Press any key to continue.
```

19. Write a program to print the reverse of a string. And estimate the time complexity Program:

```
#include<stdio.h> int main()
{   int c=0,l,i;   char s[20];
printf("Enter String:");
scanf("%s",&s);   l=strlen(s);
c++;   printf("Reverse String : ");
for(i=l-1;i>-1;i--)   {       c++;
printf("%c",s[i]);
    }   c++;
printf("\nTime Complexity : %d\n",c);
}
```

C:\Users\Admin\Documents\daa21-rev.str.exe

Enter String:Luffy

Reverse String : yffuL

Time Complexity : 1

Process returned 0 (0x0) execution time : 3.408 s

Press any key to continue.

20. Write a program to check sub string is there in a string or not.

Program:

```
#include<stdio.h> int main() {    int c=0,l1,l2,i,cnt=0;

        char

s[100],sub[20],os[20],at='@',sub1[20];    c++;    printf("Enter
String:");    scanf("%s",&s);    printf("Enter  Sub  String:");
scanf("%s",&sub);    l1=strlen(s);    c++;    l2=strlen(sub); c++;
strncat(sub,&at,1); c++;    for(i=0;i<=l1+1-l2;i++)
    {        c++;        strncpy(os,s+i,l2);
c++;        c++;
if(strcmp(sub,os)==0)
    {        cnt++; c++;
    }    }    c++;
printf("Count : %d",cnt);    printf("\nTime
Complexity
: %d\n",c);
```

}

```
C:\Users\Admin\Documents\daa22-substr.exe
Enter String:asdfasdfjklasdfasdf
Enter Sub String:sd
Count : 4
Time Complexity : 66
Process returned 0 (0x0)   execution time : 9.362 s
Press any key to continue.
```

21. Write a C program to merge sort using divide and Conquer

Program:

```
#include<stdio.h> void mergesort(int a[],int i,int
```

```

j); void merge(int a[],int i1,int j1,int i2,int j2); int
main() { int a[30],n,i; printf("Enter no of
elements:"); scanf("%d",&n); printf("Enter array
elements:\n"); for(i=0;i<n;i++)
{ scanf("%d",&a[i]); }
mergesort(a,0,n-1);
printf("Merge Sort : \n");
for(i=0;i<n;i++) {
printf("%d\n",a[i]);
} return
0;
}

```

```

void mergesort(int a[],int i,int j)
{ int mid;
if(i<j) {
mid=(i
+j)/2;
mergesort(a,i, mid);
mergesort(a,m
id+1,j);

```

```
merge( a,i,mid,
```

```
mid+1,j
);
    }
} void merge(int a[],int i1,int j1,int i2,int j2)
{   int temp[50];   int i,j,k;   i=i1;
j=i2;
k=0;   while(i<=j1 && j<=j2)
    {
if(a[i]<a[j])
    {
        temp[k++]=a[i++];
    }   else
    {
        temp[k++]=a[j++];
    }
    }
while(i<=j1)
{
    temp[k++]=a[i++];
}
while(j<=j2)
{
    temp[k++]=a[j++];
}
```

}

```

    for(i=i1,j=0;i<=j2;i++,j++)
    {
a[i]=temp[j];
    }
}

```

```

C:\Users\Admin\Documents\daa13-merge.exe
Enter no of elements:6
Enter array elements:
2
4
7
5
9
8
Merge Sort :
2
4
5
7
8
9
Process returned 0 (0x0)   execution time : 6.943 s
Press any key to continue.

```

22. Write a C program to find max-min using divide and Conquer

Program:

```

#include<stdio.h> void mergesort(int a[],int i,int j); void
merge(int a[],int i1,int j1,int i2,int j2);
int main()

```

```
{    int a[30],n,i;
```

```

printf("Enter no of
elements:");
scanf("%d",&n);
printf("Enter array
elements:\n");
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
} mergesort(a,0,n-1); printf("\nMin
: %d",a[0]); printf("\nMax : %d",a[n-1]);
return 0;
}

```

```

void mergesort(int a[],int i,int j) { int mid;
if(i<j) { mid=(i+j)/2;
mergesort(a,i,mid); mergesort(a,mid+1,j);
merge(a,i,mid,mid+1,j);
} }

```

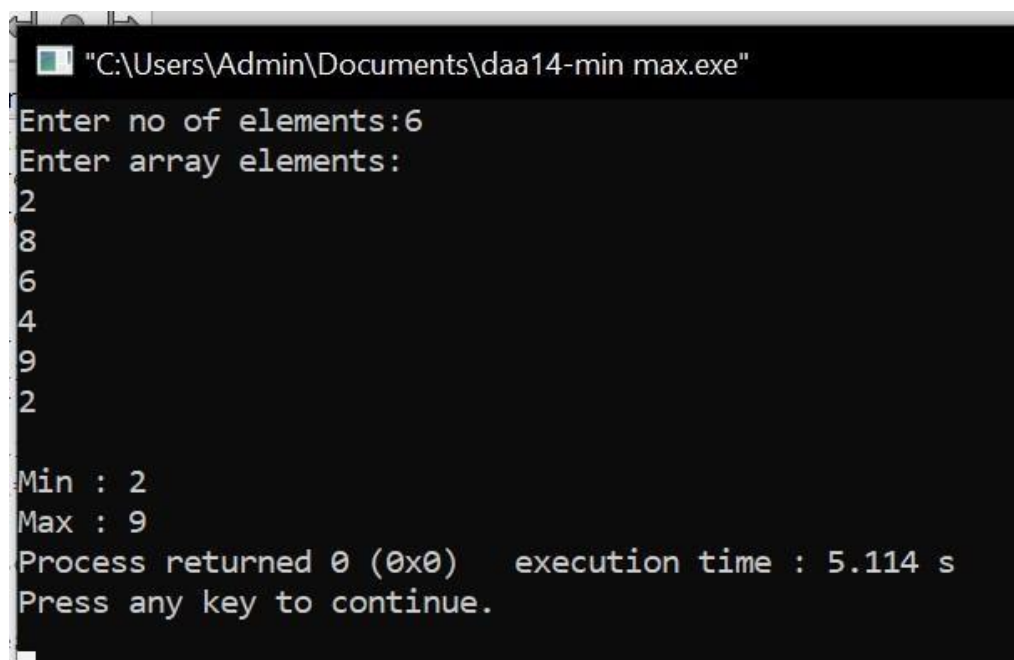
```

void merge(int a[],int i1,int j1,int i2,int j2)
{ int temp[50];
int i,j,k; i=i1; j=i2;
k=0;
while(i<=j1 && j<=j2)
{ if(a[i]<a[j])
{
temp[k++]=a[i++];
} else
{
temp[k++]=a[j++];
}
}
while(i<=j1)
{

```

```
temp[k++]=a[i++];
```

```
}  
while(j<=j2)  
{  
    temp[k++]=a[j++];  
}  
for(i=i1,j=0;i<=j2;i++,j++)  
{  
    a[i]=temp[j];  
}  
}
```



```
"C:\Users\Admin\Documents\daa14-min max.exe"  
Enter no of elements:6  
Enter array elements:  
2  
8  
6  
4  
9  
2  
  
Min : 2  
Max : 9  
Process returned 0 (0x0)   execution time : 5.114 s  
Press any key to continue.  
_
```