

Name: **Muhammad Anas Baig**

Enrollment No.: **01-134152-037**

Section: **BS(CS)-4A**



## LAB-JOURNAL-4

### Exercise 1:

Ackermann's function is defined recursively on non-negative integers as follows.

$A(m,n) = n+1$  if  $m == 0$

$A(m,n) = A(m-1, 1)$  if  $m \neq 0, n == 0$

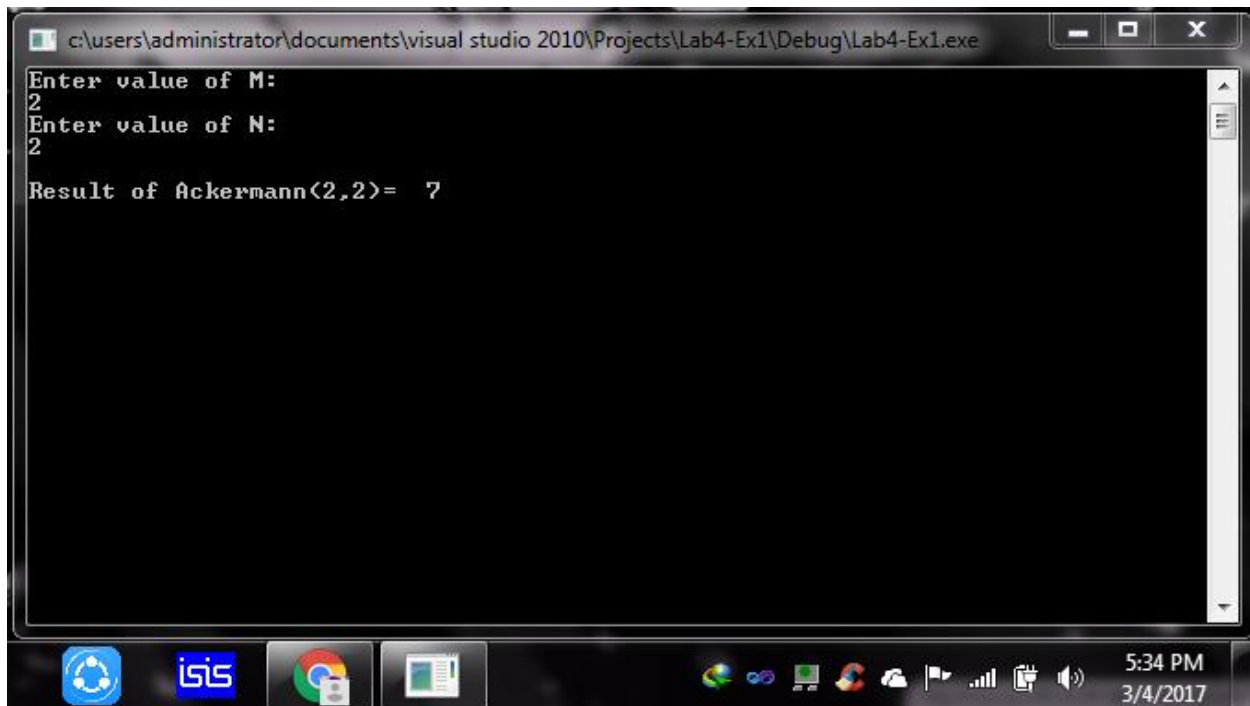
$A(m,n) = A(m-1, A(m, n-1))$  if  $m \neq 0, n \neq 0$

Implement it as a recursive function Ackermann(M,N) which takes two positive integers as input and returns a positive integer as result. Once implemented test your program by evaluating Ackermann(2,2).

### Solution:

```
1. #include <iostream>
2. #include "conio.h"
3. using namespace std;
4.
5. int ackermann(int m, int n)
6. {
7.     if(m==0)
8.     {
9.         return (n+1);
10.    }
11.    else if(m!=0 && n==0)
12.    {
13.        return (ackermann(m-1, 1));
14.    }
15.    else if(m!=0 && n!=0)
16.    {
17.        return ( ackermann( m-1, ackermann(m, n-1) ) );
18.    }
19. }
20.
21. void main()
22. {
23.     int m, n;
24.     cout<<"Enter value of M:"<<endl;
25.     cin>>m;
26.     cout<<"Enter value of N:"<<endl;
27.     cin>>n;
28.     cout<<endl;
29.     cout<<"Result of Ackermann("<<m<<","<<n<<")= "<<ackermann(m, n);
30.     getch();
31. }
```

## Output:



```
c:\users\administrator\documents\visual studio 2010\Projects\Lab4-Ex1\Debug\Lab4-Ex1.exe
Enter value of M:
2
Enter value of N:
2
Result of Ackermann(2,2)= 7
```

## Exercise 2:

Write a function `sum(int a[], int size)` to (recursively) compute the sum of the elements in an array.

### Example Run:

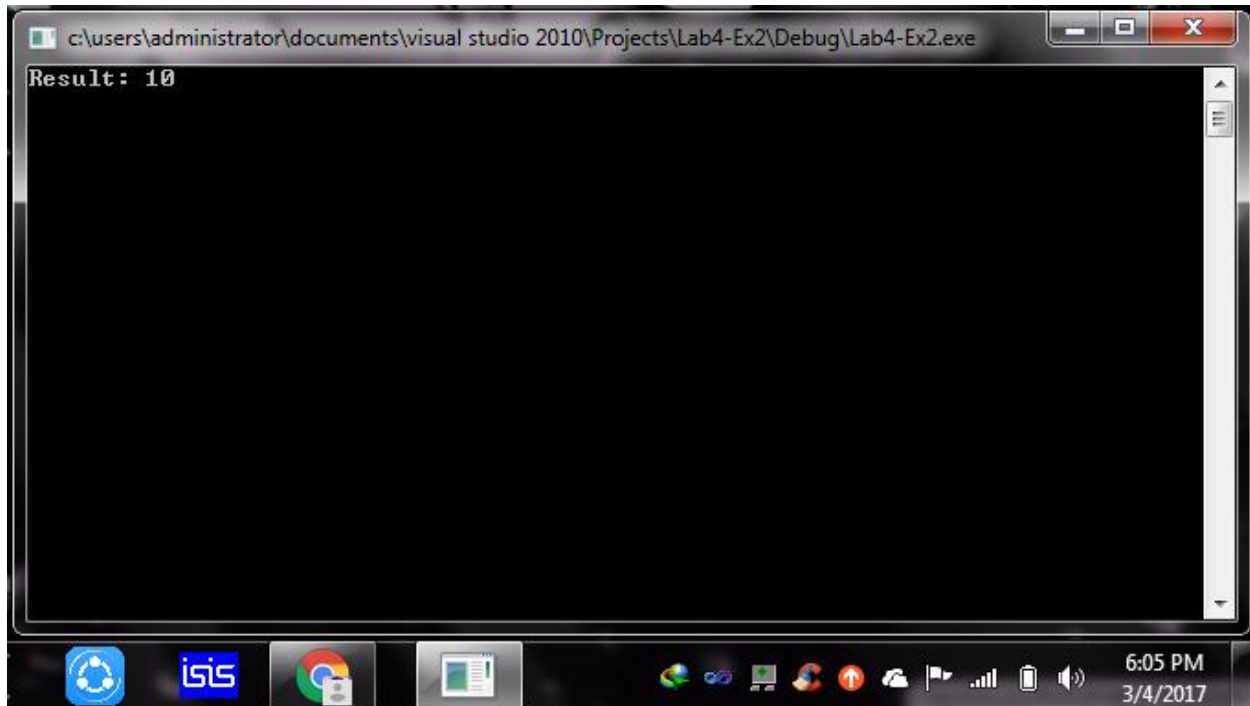
```
int arr[] = {1,2,3,4};
int result = sum(arr,4) ;
cout<<result<<endl ; //Should print 10
```

## Solution:

```
1. #include <iostream>
2. #include "conio.h"
3. using namespace std;
4.
5. int sum(int *a, int size)
6. {
7.     if(size == 0)
8.     {
9.         return (0);
10.    }
11.    else
12.    {
13.        return ( a[size-1] + sum(a, size-1) );
14.    }
15. }
16.
17. void main()
18. {
```

```
19.     int a[5]={1, 2, 3, 4};
20.     cout<<"Result: "<<sum(a, 4);
21.     getch();
22. }
```

## Output:



## Exercise 3:

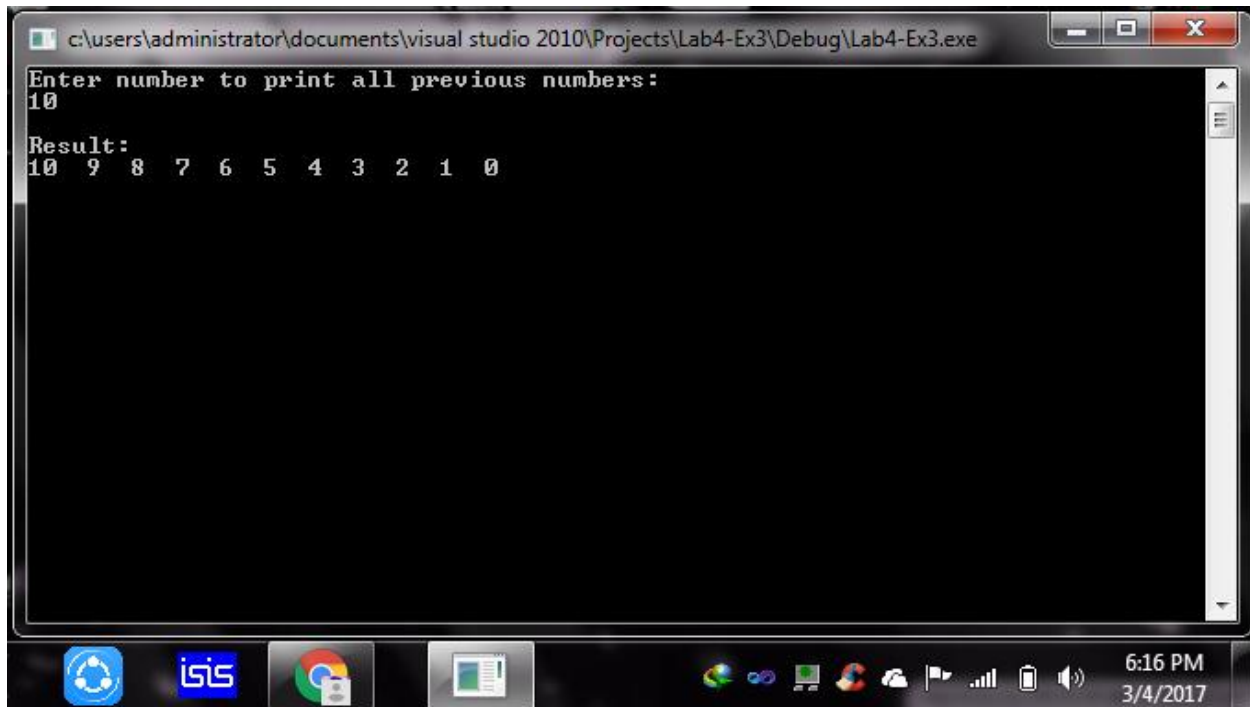
Write a recursive function to print integers from a given number N to 0. When called as print (10), the function should print: **10 9 8 7 6 5 4 3 2 1 0**

## Solution:

```
1. #include <iostream>
2. #include "conio.h"
3. using namespace std;
4.
5. int print(int number)
6. {
7.     if( number == 0 )
8.     {
9.         return (0);
10.    }
11.    else
12.    {
13.        cout<<number<<" ";
14.        print( number-1 );
15.    }
16. }
17.
18. void main()
```

```
19. {  
20.     int number;  
21.     cout<<"Enter number to print all previous numbers:"<<endl;  
22.     cin>>number;  
23.     cout<<endl;  
24.     cout<<"Result:"<<endl;  
25.     cout<<print(number);  
26.     getch();  
27. }
```

## Output:



## Exercise 4:

Binomial coefficients are normally computed using the following formula.

$$\binom{n}{m} = \frac{n!}{(n-m)!m!}$$

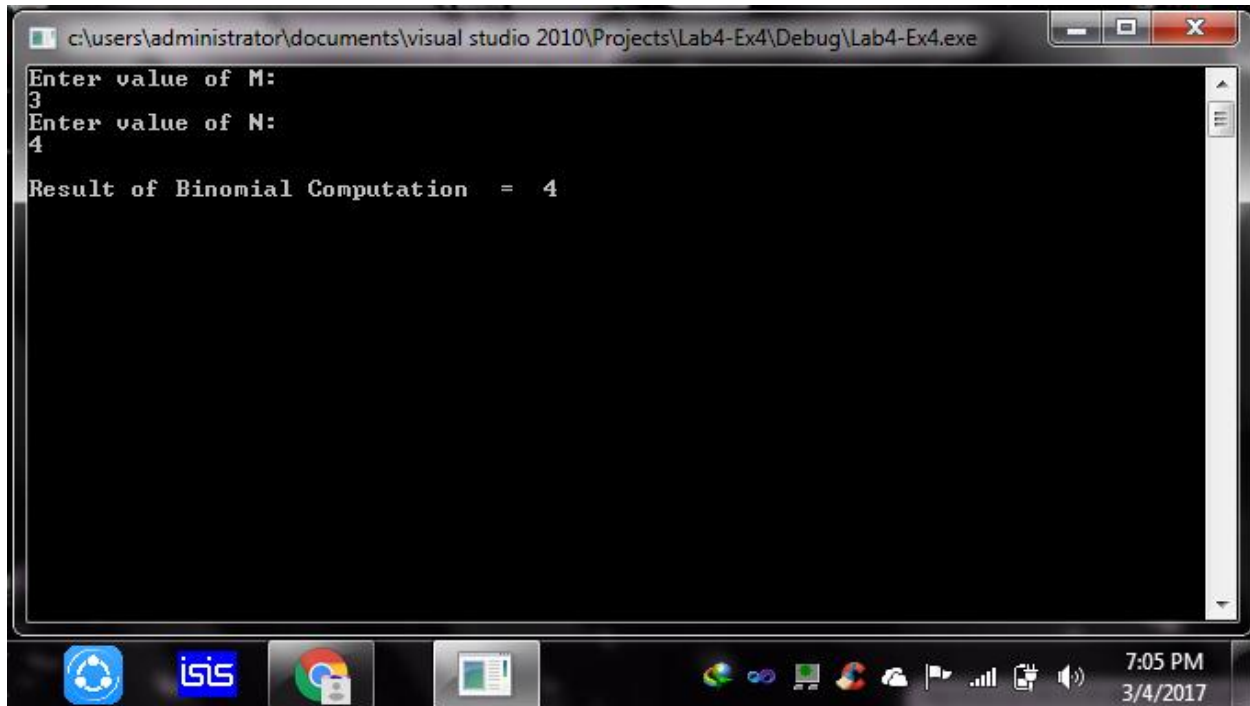
Binomial coefficients can also be computed using the following recursive definition.

$$\binom{n}{m} = \begin{cases} 1 & m = 0, \\ 1 & n = m, \\ \binom{n-1}{m} + \binom{n-1}{m-1} & \text{otherwise.} \end{cases}$$

Write a C++ program to compute binomial coefficients using the mentioned recursive definition.

**Solution:**

```
1. #include <iostream>
2. #include "conio.h"
3. using namespace std;
4.
5. int binomial_computation(int m, int n)
6. {
7.     if( m == 0 )
8.     {
9.         return (1);
10.    }
11.    else if( n == m )
12.    {
13.        return (1);
14.    }
15.    else
16.    {
17.        return ( binomial_computation(m, n-1) + binomial_computation(m-1, n-
18.    1) );
19.    }
20. }
21. void main()
22. {
23.     int m, n;
24.     cout<<"Enter value of M:"<<endl;
25.     cin>>m;
26.     cout<<"Enter value of N:"<<endl;
27.     cin>>n;
28.     cout<<endl;
29.     if(n>=m) //M must be less than N bcz (N-M)! = Infinity
30.     {
31.         cout<<"Result of Binomial Computation = "<<binomial_computation(m, n);
32.     }
33.     else
34.     {
35.         cout<<"INVALID VALUES!!! M couldn't be greater than N because
36.         (N-M)! = Infinity"<<endl;
37.     }
38.     getch();
39. }
```

**Output:**

```
c:\users\administrator\documents\visual studio 2010\Projects\Lab4-Ex4\Debug\Lab4-Ex4.exe
Enter value of M:
3
Enter value of N:
4
Result of Binomial Computation = 4
```