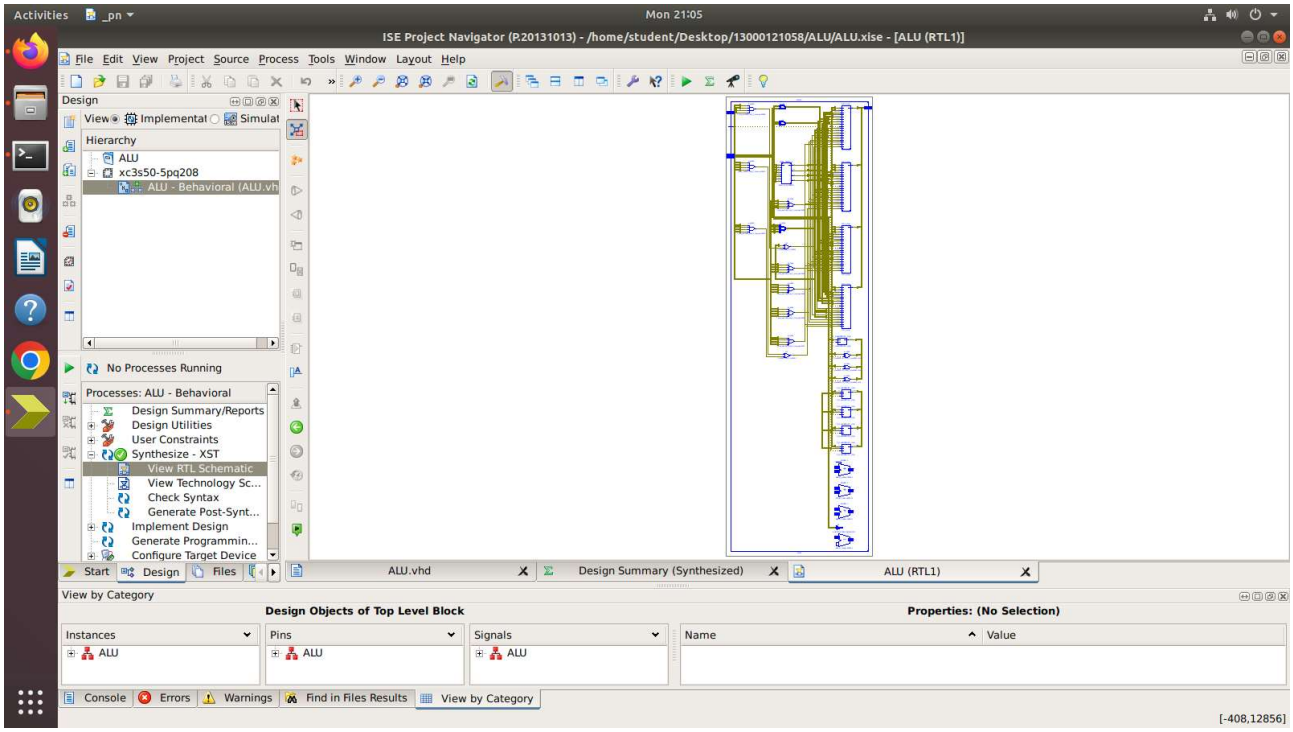
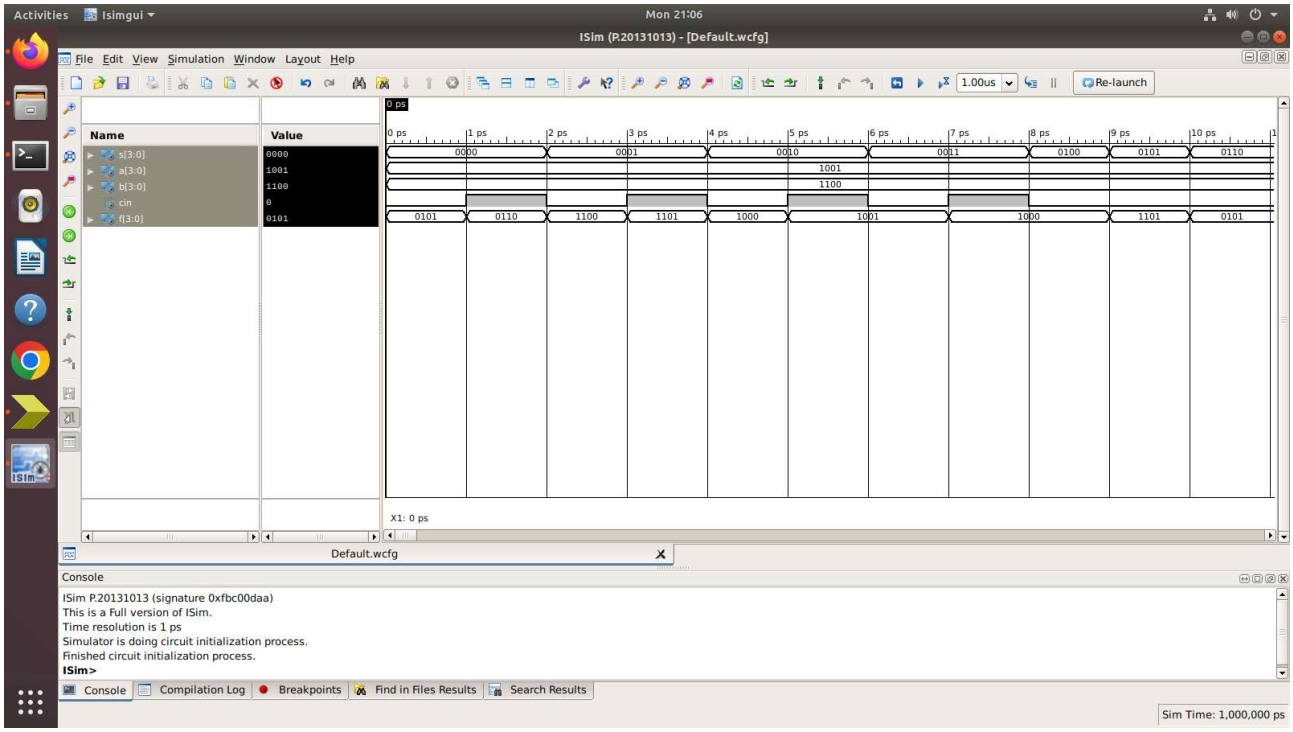
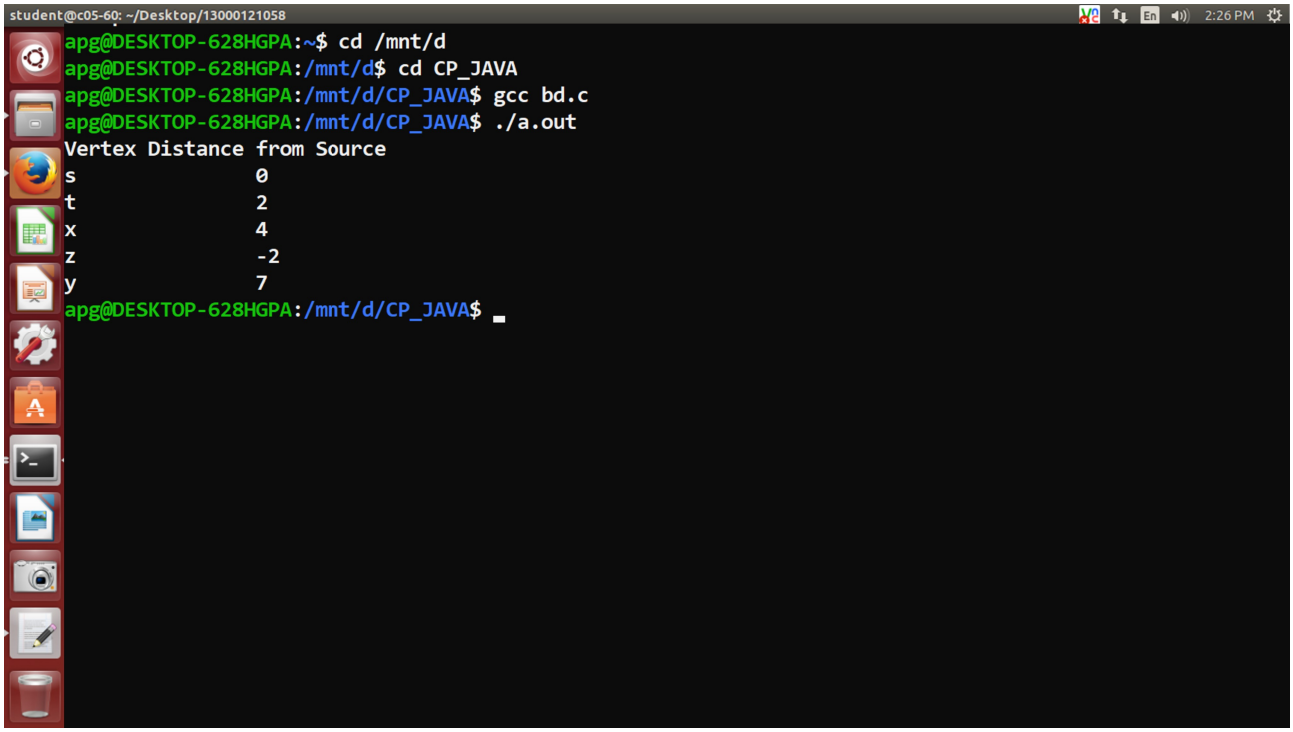
SCHEMATIC OUTPUT



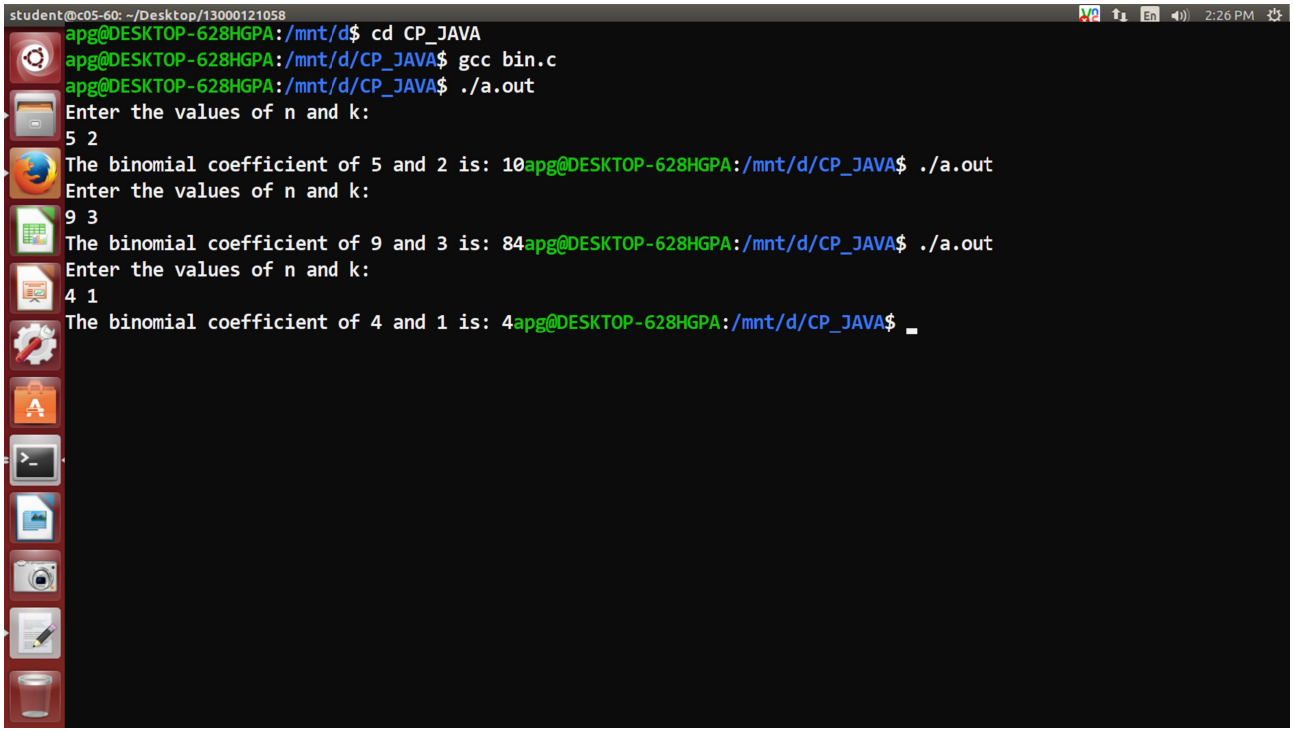
TEST OUTPUT


Activities
Isimgui
Mon 21:06
ISim (P.20131013) - [Default.wcfg]
File Edit View Simulation Window Layout Help
1.00us
Re-launch
Name
Value
s[3:0]
a[3:0]
b[3:0]
cin
f[3:0]
0000
1001
1100
0
0101
0 ps
1 ps
2 ps
3 ps
4 ps
5 ps
6 ps
7 ps
8 ps
9 ps
10 ps
0000
0001
0010
0011
0100
0101
0110
1001
1100
0101
0110
1100
1101
1000
1001
1000
1101
0101
X1: 0 ps
Default.wcfg
Console
ISim P.20131013 (signature 0xfbc00daa)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
Console
Compilation Log
Breakpoints
Find in Files Results
Search Results
Sim Time: 1,000,000 ps

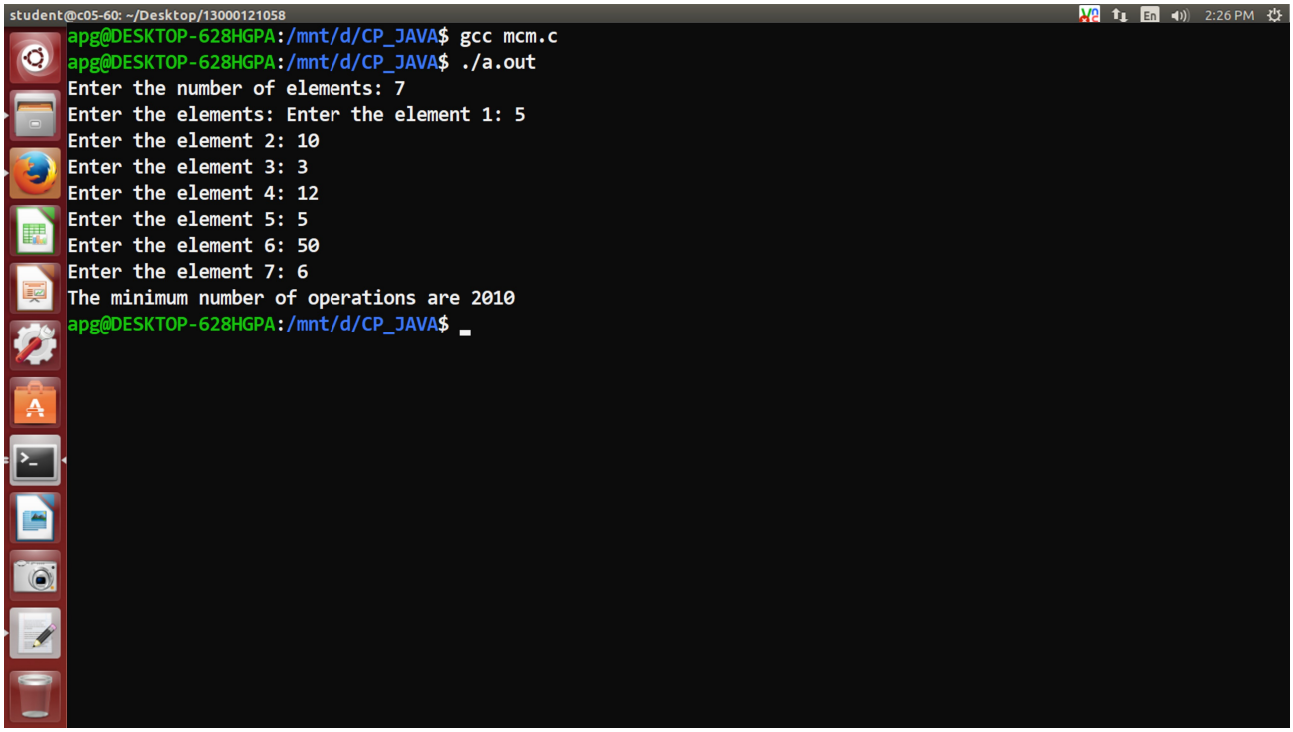
OUTPUT


student@c05-60: ~/Desktop/13000121058
2:26 PM
apg@DESKTOP-628HGPA:~$ cd /mnt/d
apg@DESKTOP-628HGPA:/mnt/d$ cd CP_JAVA
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ gcc bd.c
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ ./a.out
Vertex Distance from Source
s 0
t 2
x 4
z -2
y 7
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$

OUTPUT


student@c05-60: ~/Desktop/13000121058
2:26 PM
apg@DESKTOP-628HGPA:/mnt/d$ cd CP_JAVA
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ gcc bin.c
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ ./a.out
Enter the values of n and k:
5 2
The binomial coefficient of 5 and 2 is: 10apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ ./a.out
Enter the values of n and k:
9 3
The binomial coefficient of 9 and 3 is: 84apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ ./a.out
Enter the values of n and k:
4 1
The binomial coefficient of 4 and 1 is: 4apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$

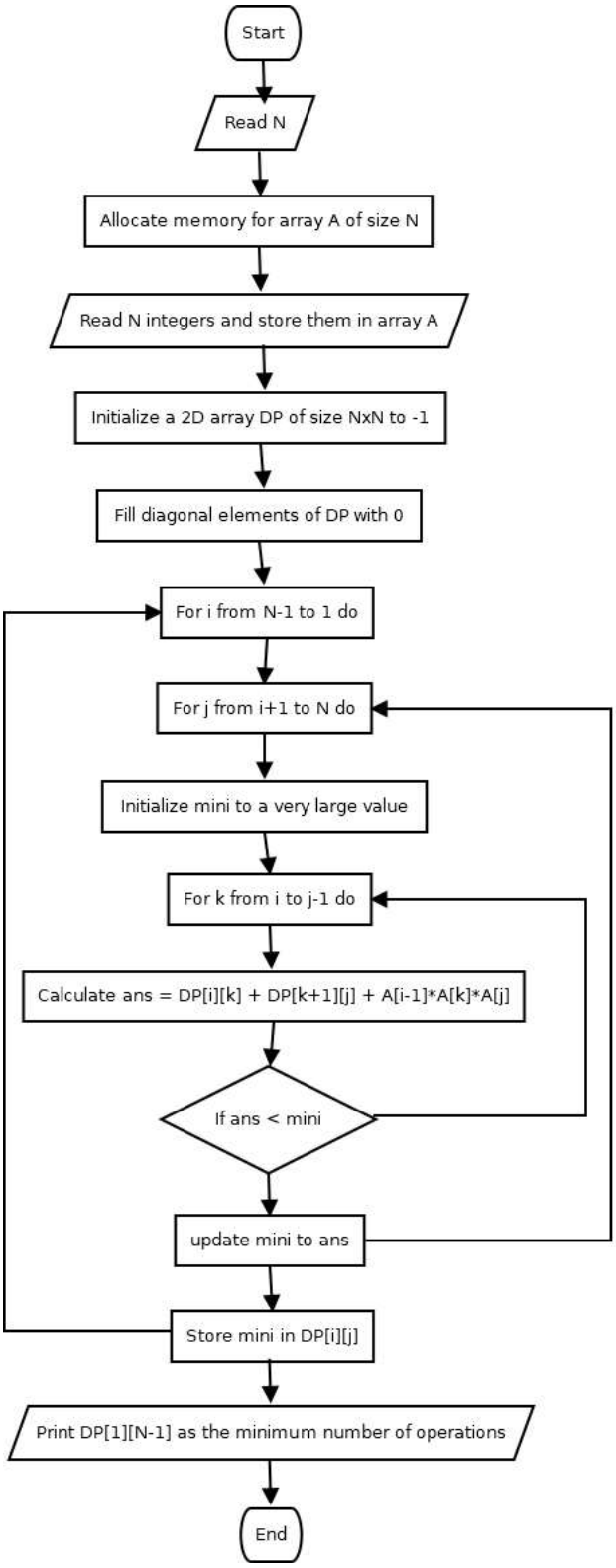
OUTPUT


student@c05-60: ~/Desktop/13000121058
2:26 PM
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ gcc mcm.c
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$ ./a.out
Enter the number of elements: 7
Enter the elements: Enter the element 1: 5
Enter the element 2: 10
Enter the element 3: 3
Enter the element 4: 12
Enter the element 5: 5
Enter the element 6: 50
Enter the element 7: 6
The minimum number of operations are 2010
apg@DESKTOP-628HGPA:/mnt/d/CP_JAVA$

ASSIGNMENT 3.2

Find the minimum number of scalar multiplication needed for chain of matrix whose sequences are < 5, 10, 3, 12, 5, 50, 6 > using the dynamic programming technique.

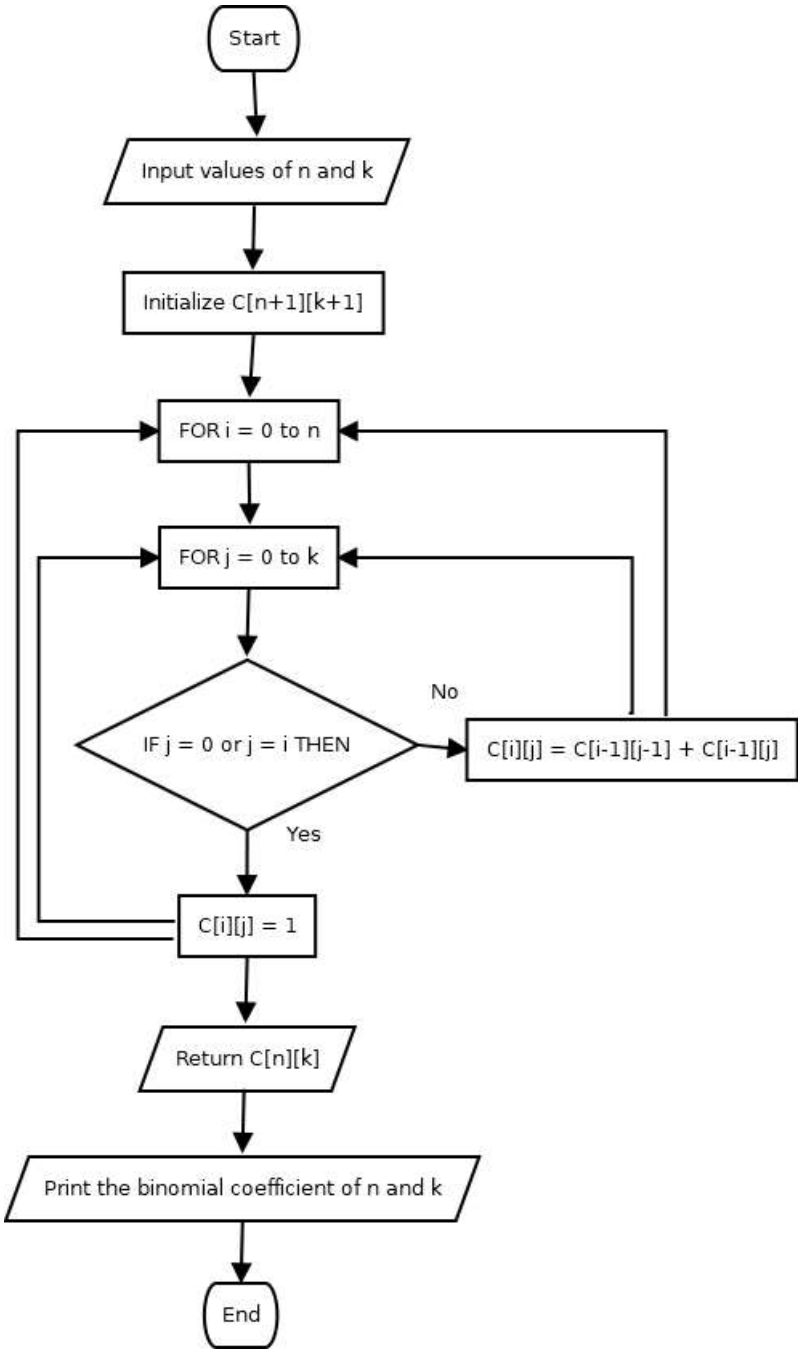
FLOWCHART



ASSIGNMENT 3.1

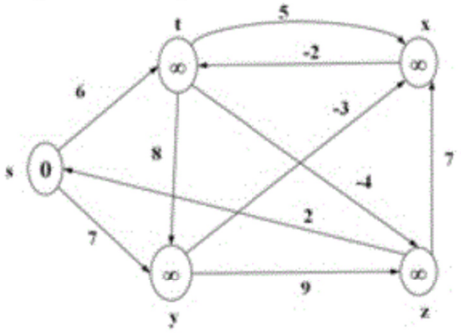
Write a program to find the binomial coefficient using Dynamic programming method.

FLOWCHART

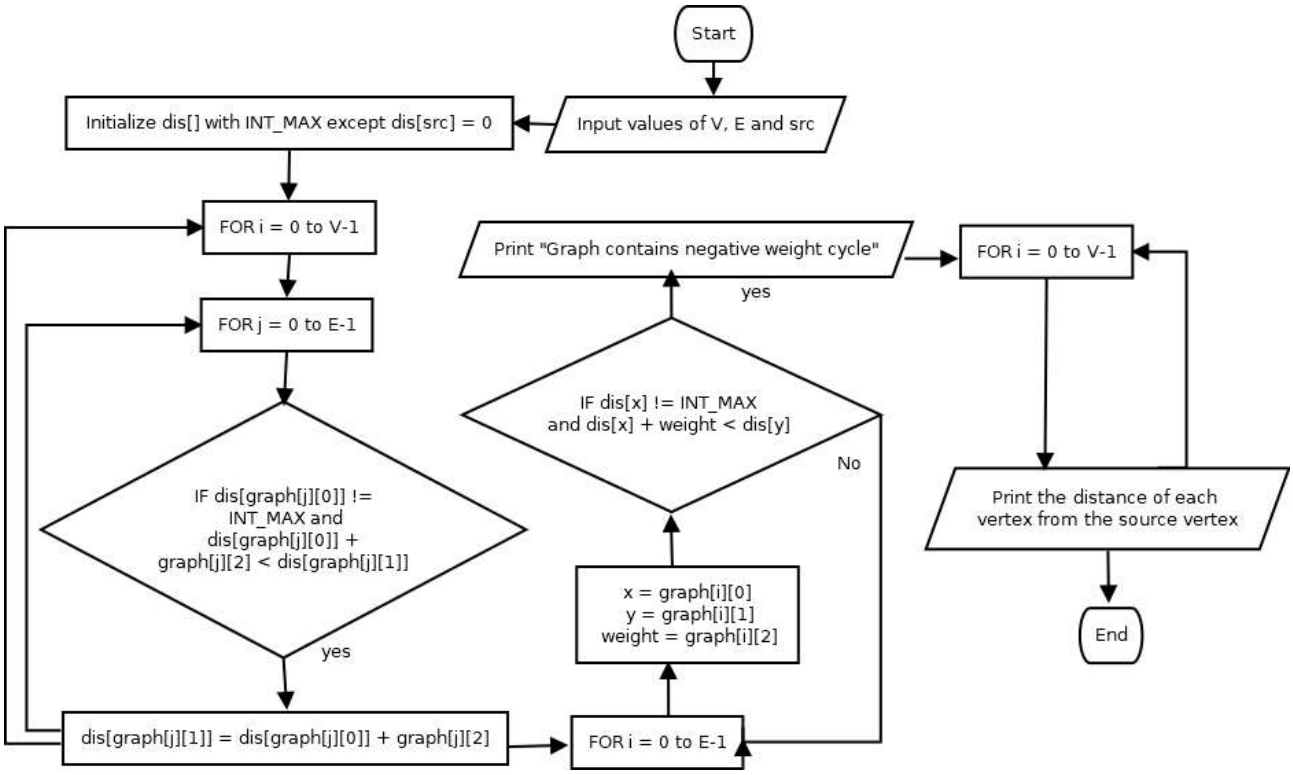


ASSIGNMENT 4.2

WAP using the single-source-shortest-path problem to find out the shortest path from the source vertex ‘s’ using the Bellman-Ford’s algorithm, using the dynamic programming technique.



FLOWCHART



mcm.c

```c
#include <stdio.h>
#include <limits.h>
#include <stdlib.h>
int matrixMultiplication(int arr[], int N)
{
    int dp[N][N];
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
            dp[i][j] = -1;
    }
    for (int i = 1; i < N; i++)
        dp[i][i] = 0;
    for (int i = N - 1; i >= 1; i--)
    {
        for (int j = i + 1; j < N; j++)
        {
            int mini = INT_MAX;
            for (int k = i; k <= j - 1; k++)
            {
                int ans = dp[i][k] + dp[k + 1][j] + arr[i - 1] * arr[k] * arr[j];
                mini = (mini < ans) ? mini : ans;
            }
            dp[i][j] = mini;
        }
    }
    return dp[1][N - 1];
}
int main()
{
    //int arr[] = {5, 10, 3, 12, 5, 50, 6};
    int *arr,n;
    printf("Enter the number of elements: ");
    scanf("%d",&n);
    fflush(stdin);
    arr = (int *)malloc(n*sizeof(int));
    printf("Enter the elements: ");
    for(int i=0;i<n;i++)
    {
        printf("Enter the element %d: ",i+1);
        scanf("%d",&arr[i]);
        fflush(stdin);
    }
    printf("The minimum number of operations are %d\n", matrixMultiplication(arr, n));
    return 0;
}
```

bin.c

```c
#include <stdio.h>
#include <stdlib.h>
int binomialCoeff(int n, int k) {
    int C[n + 1][k + 1];
    int i, j;
    for (i = 0; i <= n; i++) {
        for (j = 0; j <= k; j++) {
            if (j == 0 || j == i)
                C[i][j] = 1;

            else
                C[i][j] = C[i - 1][j - 1] + C[i - 1][j];
        }
    }
    return C[n][k];
}
int main() {
    int n, k;
    printf("Enter the values of n and k:\n");
    scanf("%d %d", &n, &k);
    printf("The binomial coefficient of %d and %d is: %d", n, k, binomialCoeff(n, k));
    return 0;
}
```

bd.c

```c
#include <stdio.h>
#include <limits.h>
#define MAX_VERTICES 100
#define MAX_EDGES 100
void BellmanFord(int graph[MAX_EDGES][3], int V, int E, int src)
{
    int dis[MAX_VERTICES];
    for (int i = 0; i < V; i++)
        dis[i] = INT_MAX;
    dis[src] = 0;
    for (int i = 0; i < V - 1; i++)
    {
        for (int j = 0; j < E; j++)
        {
            if (dis[graph[j][0]] != INT_MAX && dis[graph[j][0]] + graph[j][2] < dis[graph[j]
[1]])
                dis[graph[j][1]] = dis[graph[j][0]] + graph[j][2];
        }
    }
    for (int i = 0; i < E; i++)
    {
        int x = graph[i][0];
        int y = graph[i][1];
        int weight = graph[i][2];
        if (dis[x] != INT_MAX && dis[x] + weight < dis[y])
            printf("Graph contains negative weight cycle");
    }
    printf("Vertex Distance from Source\n");
    for (int i = 0; i < V; i++)
    {
        if(i==0)printf("s\t\t%d\n", dis[i]);
        else if(i==1)printf("t\t\t%d\n", dis[i]);
        else if(i==2)printf("x\t\t%d\n", dis[i]);
        else if(i==3)printf("z\t\t%d\n", dis[i]);
        else if(i==4)printf("y\t\t%d\n", dis[i]);
    }
    //printf("%d\t\t%d\n", i, dis[i]);
}
int main()
{
    int V = 5;
    int E = 9;
    int graph[MAX_EDGES][3] = { {0,1,6},{1,4,8},{0,4,7},{4,3,9},{3,2,7},{2,1,-2},{4,2,-3},
{1,3,-4},{3,0,2} };
    BellmanFord(graph, V, E, 0);
    return 0;
}
```