Design and Analysis Of Algorithms

Lab Assignment

By :

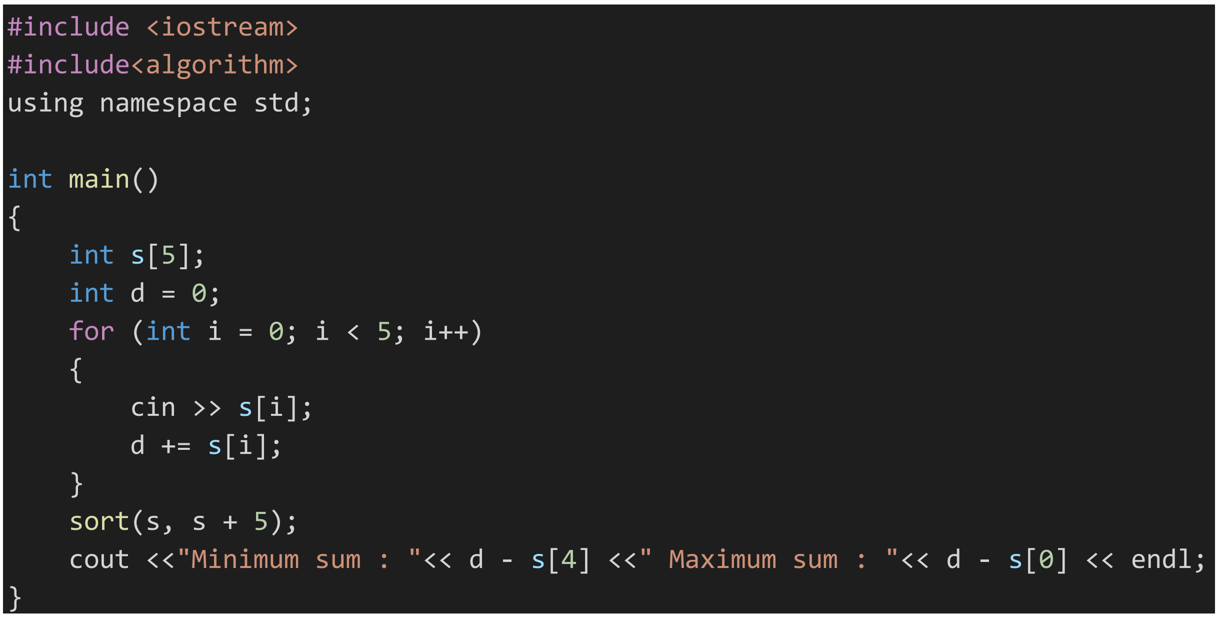
Name : Aryan Tyagi

Roll Number : 2020UCO1549

Class : COE-1

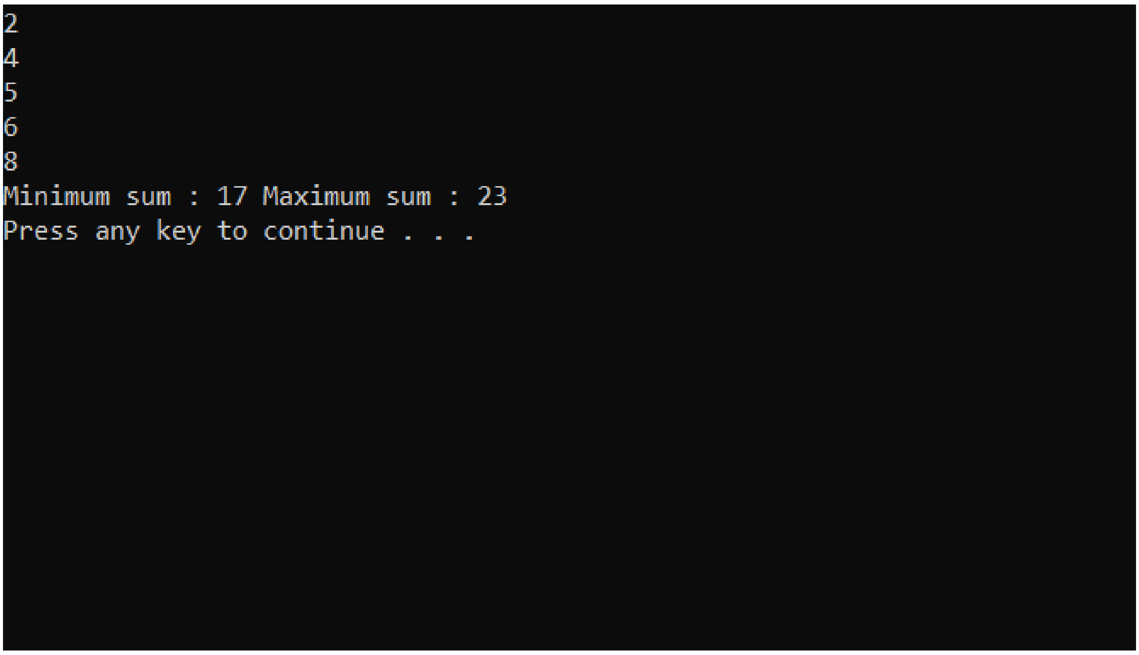
Answer 1

Code



```cpp
#include <iostream>
#include<algorithm>
using namespace std;

int main()
{
    int s[5];
    int d = 0;
    for (int i = 0; i < 5; i++)
    {
        cin >> s[i];
        d += s[i];
    }
    sort(s, s + 5);
    cout <<"Minimum sum : "<< d - s[4] <<" Maximum sum : "<< d - s[0] << endl;
}
```

Output



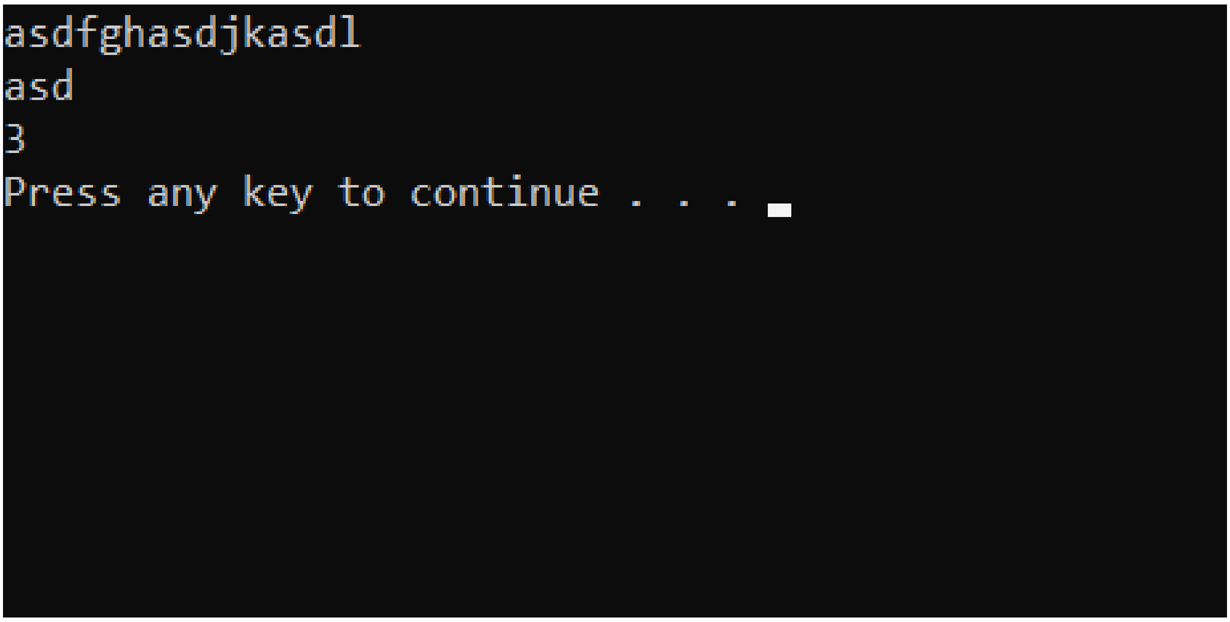
```
2
4
5
6
8
Minimum sum : 17 Maximum sum : 23
Press any key to continue . . .
```

Answer 2

Code

```
#include <iostream>

using namespace std;

void pre_calc(string pattern, int n, int prefix[])
{
    int len = 0;
    int i = 1;
    prefix[0] = 0;
    while (i < n)
    {
        if (pattern[i] == pattern[len])
        {
            len++;
            prefix[i] = len;
            i++;
        }
        else
        {

            if (len != 0)
            {
                len = prefix[len - 1];
            }
            else
            {
                prefix[i] = len;
                i++;
            }
        }
    }
}

int kmp(string str , string pattern)
{
    int m = pattern.length();
    int n = str.length();

    int prefix[m];

    pre_calc( pattern, m, prefix);
    int i = 0;
    int j = 0;
    int ans = 0;

    while (i < n)
    {
        if ( pattern[j] == str[i])
        {
            j++;
            i++;
        }
        if (j == m)
        {

            j = prefix[j - 1];
            ans++;
        }

        else if (i < n && pattern[j] != str[i])
        {

            if (j != 0)
                j = prefix[j - 1];
            else
                i++;
        }
    }

    return ans;
}

int main()
{
    string str, pattern;
    cin >> str;
    cin >> pattern;
    int ans = kmp(str,pattern);

    cout << ans <<endl;

    return 0;
}
```

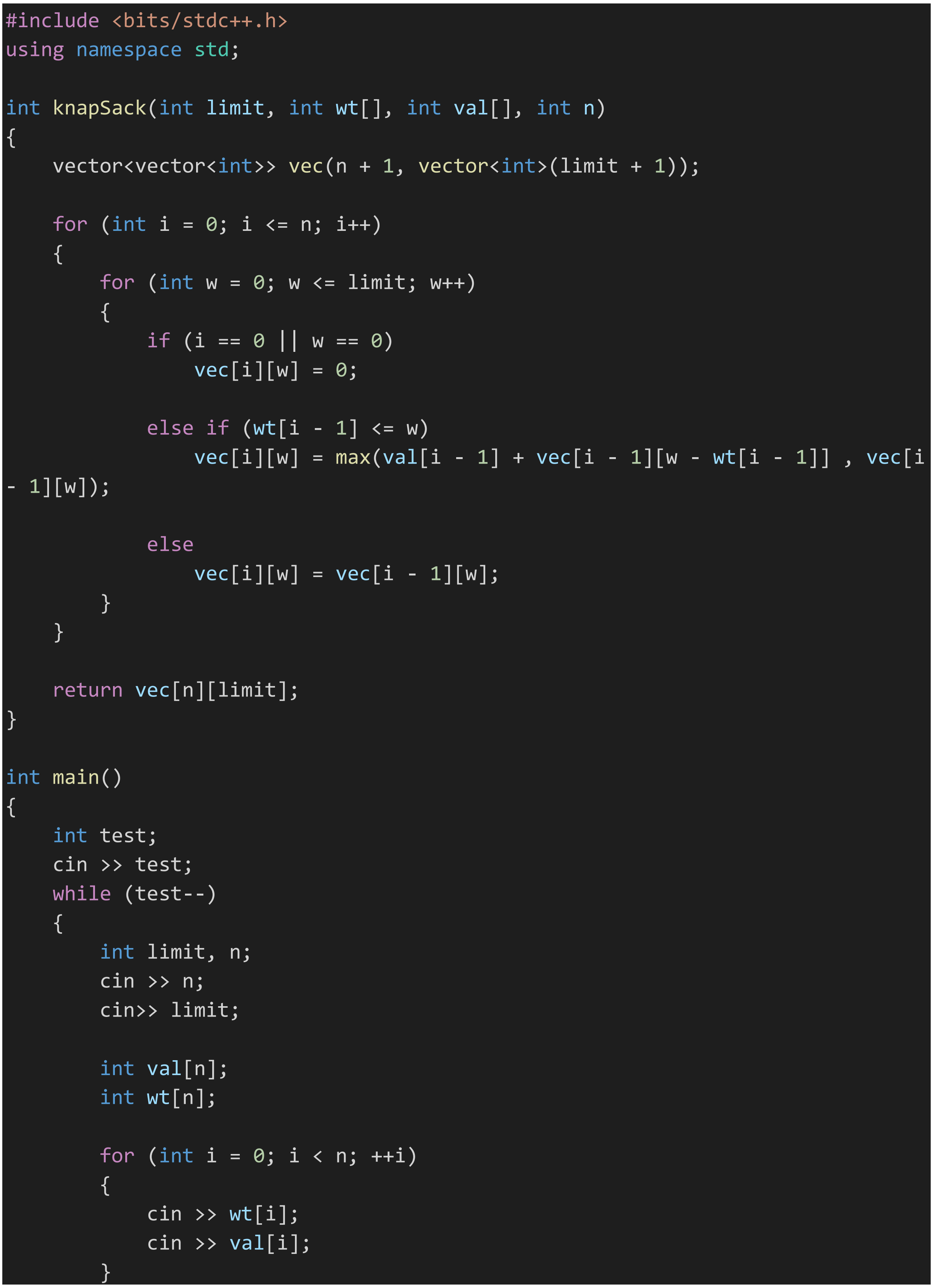
Output



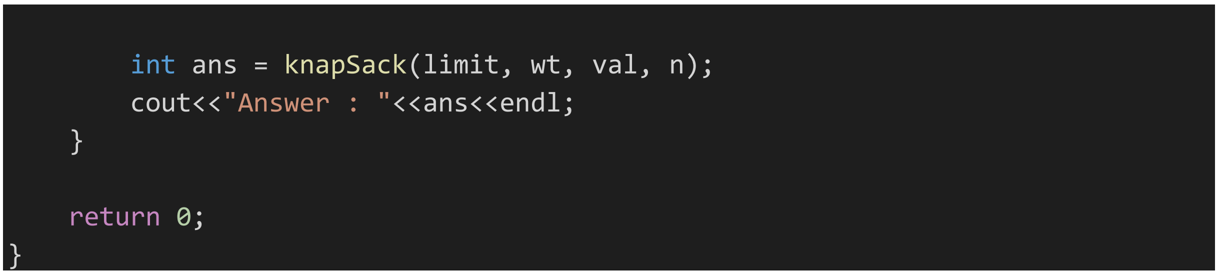
```
asdfghasdjkasdl
asd
3
Press any key to continue . . .
```

Answer 3

Code

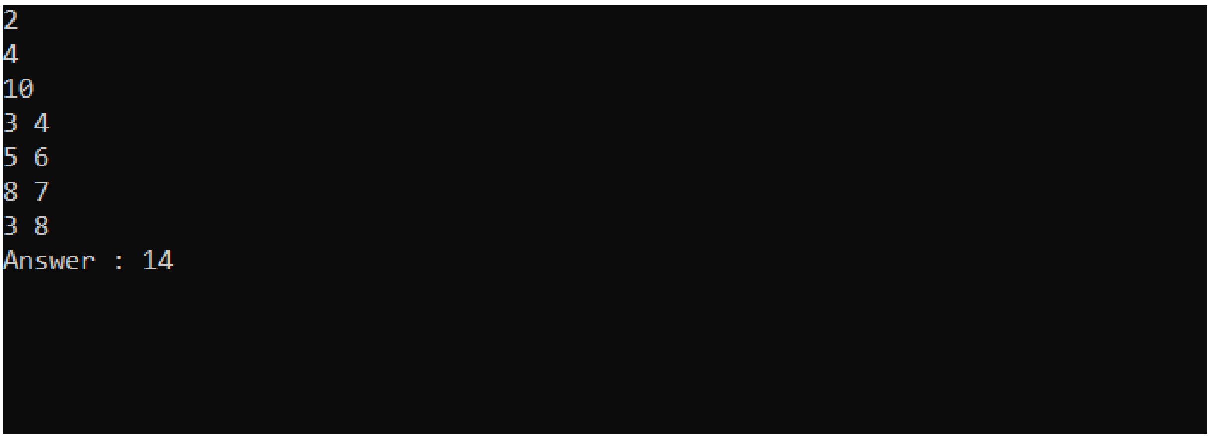


```
#include <bits/stdc++.h>
using namespace std;

int knapSack(int limit, int wt[], int val[], int n)
{
    vector<vector<int>> vec(n + 1, vector<int>(limit + 1));

    for (int i = 0; i <= n; i++)
    {
        for (int w = 0; w <= limit; w++)
        {
            if (i == 0 || w == 0)
                vec[i][w] = 0;

            else if (wt[i - 1] <= w)
                vec[i][w] = max(val[i - 1] + vec[i - 1][w - wt[i - 1]] , vec[i
- 1][w]);

            else
                vec[i][w] = vec[i - 1][w];
        }
    }

    return vec[n][limit];
}

int main()
{
    int test;
    cin >> test;
    while (test--)
    {
        int limit, n;
        cin >> n;
        cin>> limit;

        int val[n];
        int wt[n];

        for (int i = 0; i < n; ++i)
        {
            cin >> wt[i];
            cin >> val[i];
        }
```



```
        int ans = knapSack(limit, wt, val, n);
        cout<<"Answer : "<<ans<<endl;
    }

    return 0;
}
```

Output



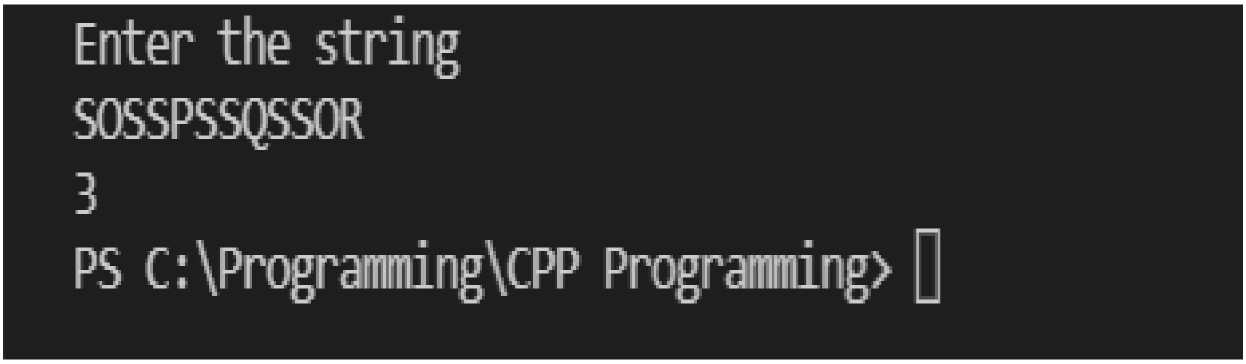
```
2
4
10
3 4
5 6
8 7
3 8
Answer : 14
```

Answer 4

Code

```
#include<bits/stdc++.h>

using namespace std;


int marsExploration(string s){

    int i, ans = 0;

        for (i = 0; s[i] != '\0'; i += 3){

            if(s[i] != 'S'){

                ++ans;

            }

            if(s[i + 1] != 'O'){

                ++ans;

            }

            if(s[i + 2] != 'S'){

                ++ans;

            }

        }

    return ans;

}

int main(){

    cout<<"Enter the string ";

    string s;    cout<<endl;

    cin>>s;

    cout<< marsExploration(s);

}
```

Output

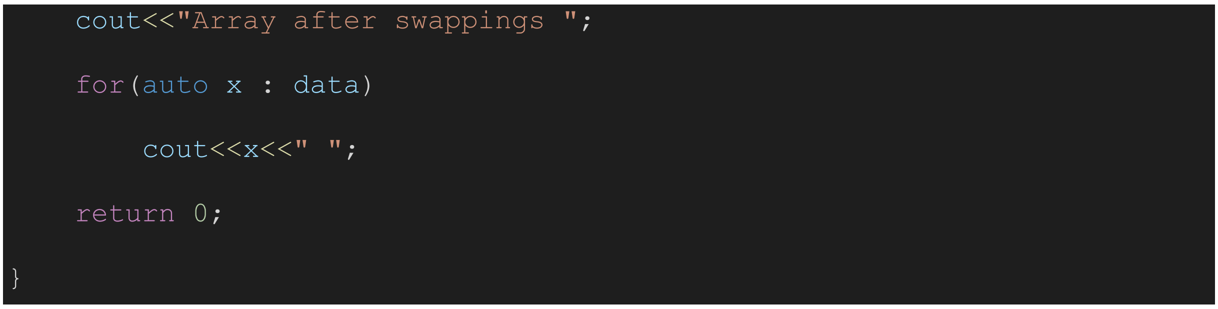


```
Enter the string
SOSSPSSQSSOR
3
PS C:\Programming\CPP Programming> 
```

Answer 5

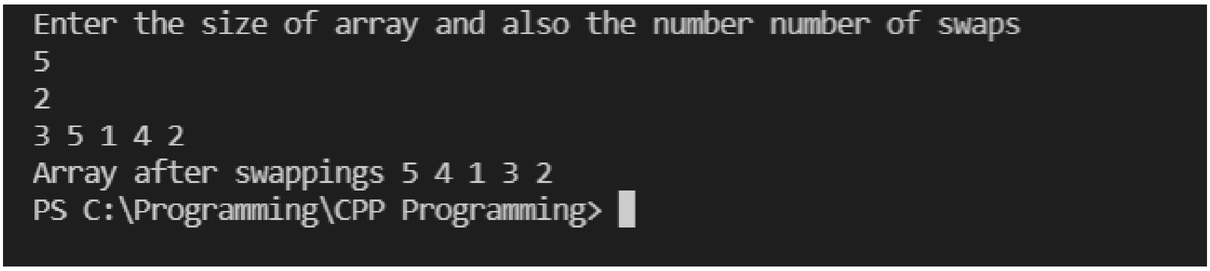
Code

```cpp
#include<bits/stdc++.h>
using namespace std;

int main(){
    int n,k;
    cout<<"Enter the size of array and also the number number of swaps"<<endl;
    cin >>n>>k;
    vector<int> data(n);
    for (int i=0;i<n; ++i)
        cin >> data.at(i);
    vector<int> idx(1000);

    for (int i = 0; i<n;++i)
        idx.at(data.at(i))=i;

    for (int i=n;i>=1 && k>0;i--){
        int max_val = n-i;
        if (max_val == idx[i])
            continue;
        int p = idx[i];
        swap(idx[i], idx[data[max_val]]);
        swap(data[max_val], data[p]);

        k--;
    }
```



```
    cout<<"Array after swappings ";

    for(auto x : data)

        cout<<x<<" ";

    return 0;

}
```

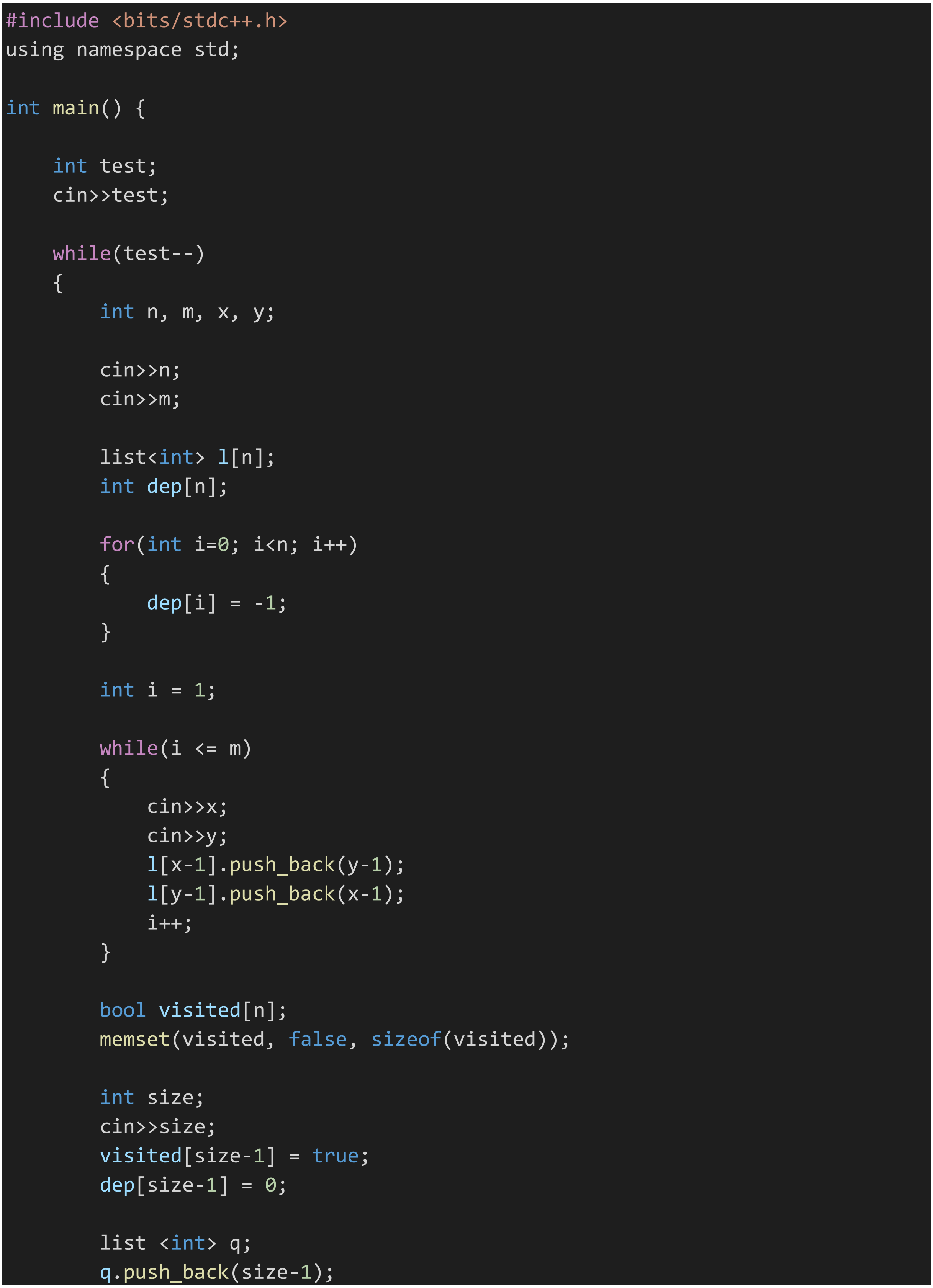
Output



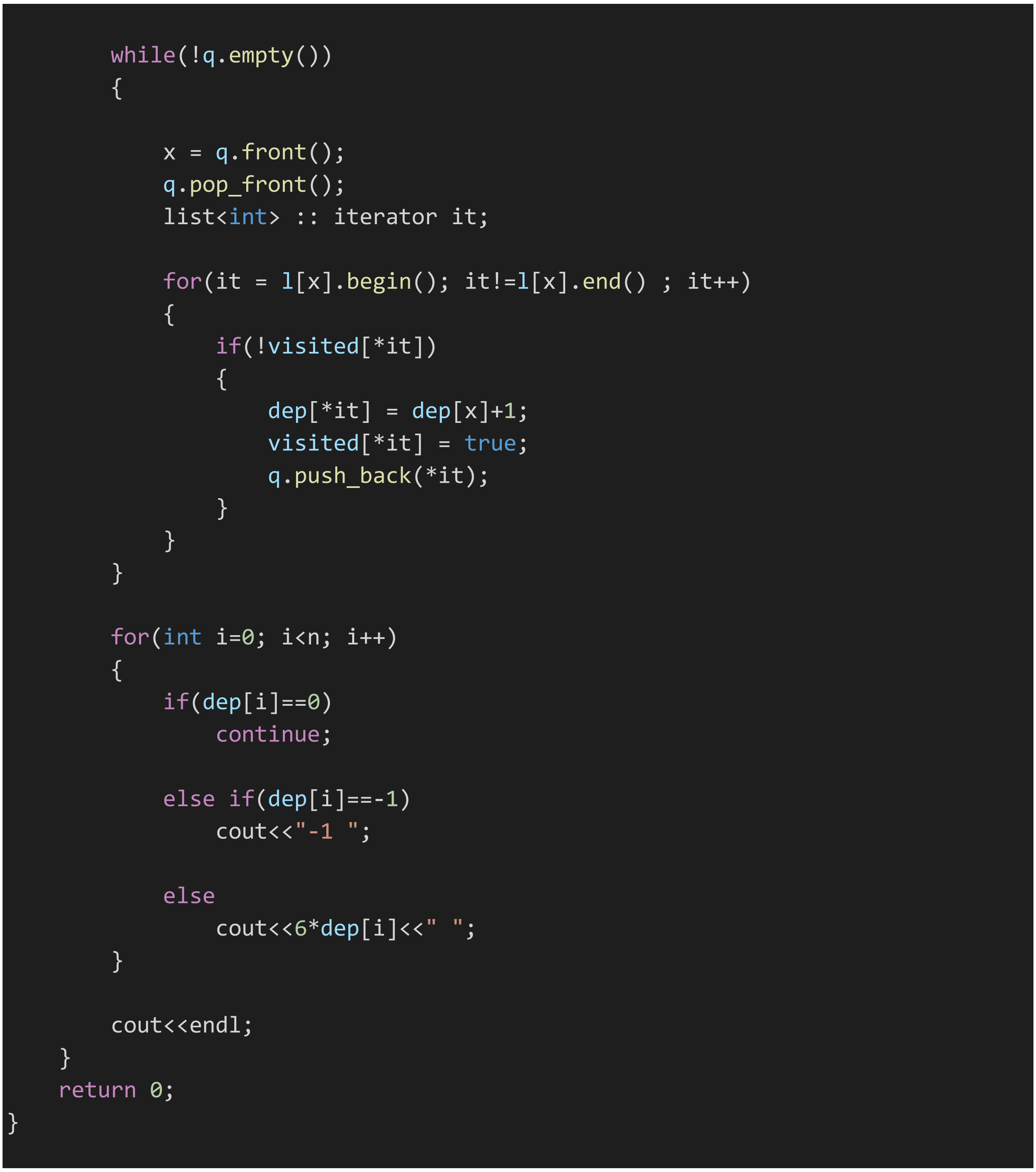
```
Enter the size of array and also the number number of swaps
5
2
3 5 1 4 2
Array after swappings 5 4 1 3 2
PS C:\Programming\CPP Programming>
```

Answer 6

Code

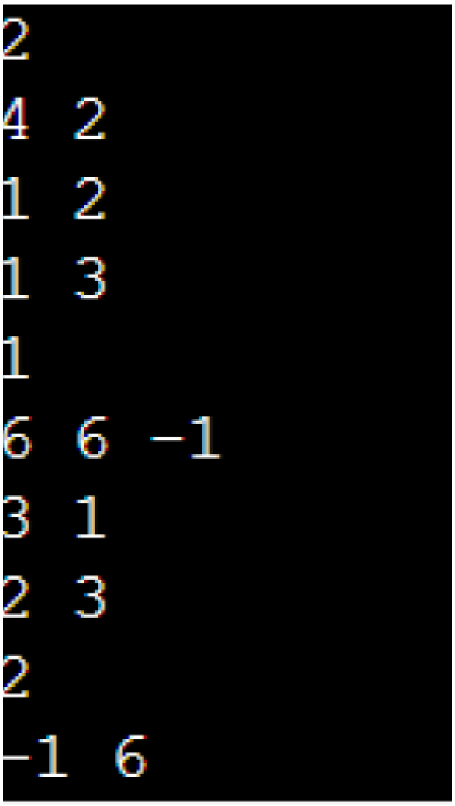


```
#include <bits/stdc++.h>
using namespace std;

int main() {

    int test;
    cin>>test;

    while(test--)
    {
        int n, m, x, y;

        cin>>n;
        cin>>m;

        list<int> l[n];
        int dep[n];

        for(int i=0; i<n; i++)
        {
            dep[i] = -1;
        }

        int i = 1;

        while(i <= m)
        {
            cin>>x;
            cin>>y;
            l[x-1].push_back(y-1);
            l[y-1].push_back(x-1);
            i++;
        }

        bool visited[n];
        memset(visited, false, sizeof(visited));

        int size;
        cin>>size;
        visited[size-1] = true;
        dep[size-1] = 0;

        list <int> q;
        q.push_back(size-1);
```



```
        while(!q.empty())
        {

            x = q.front();
            q.pop_front();
            list<int> :: iterator it;

            for(it = l[x].begin(); it!=l[x].end() ; it++)
            {
                if(!visited[*it])
                {
                    dep[*it] = dep[x]+1;
                    visited[*it] = true;
                    q.push_back(*it);
                }
            }
        }

        for(int i=0; i<n; i++)
        {
            if(dep[i]==0)
                continue;

            else if(dep[i]==-1)
                cout<<"-1 ";

            else
                cout<<6*dep[i]<<" ";
        }

        cout<<endl;
    }
    return 0;
}
```

Output

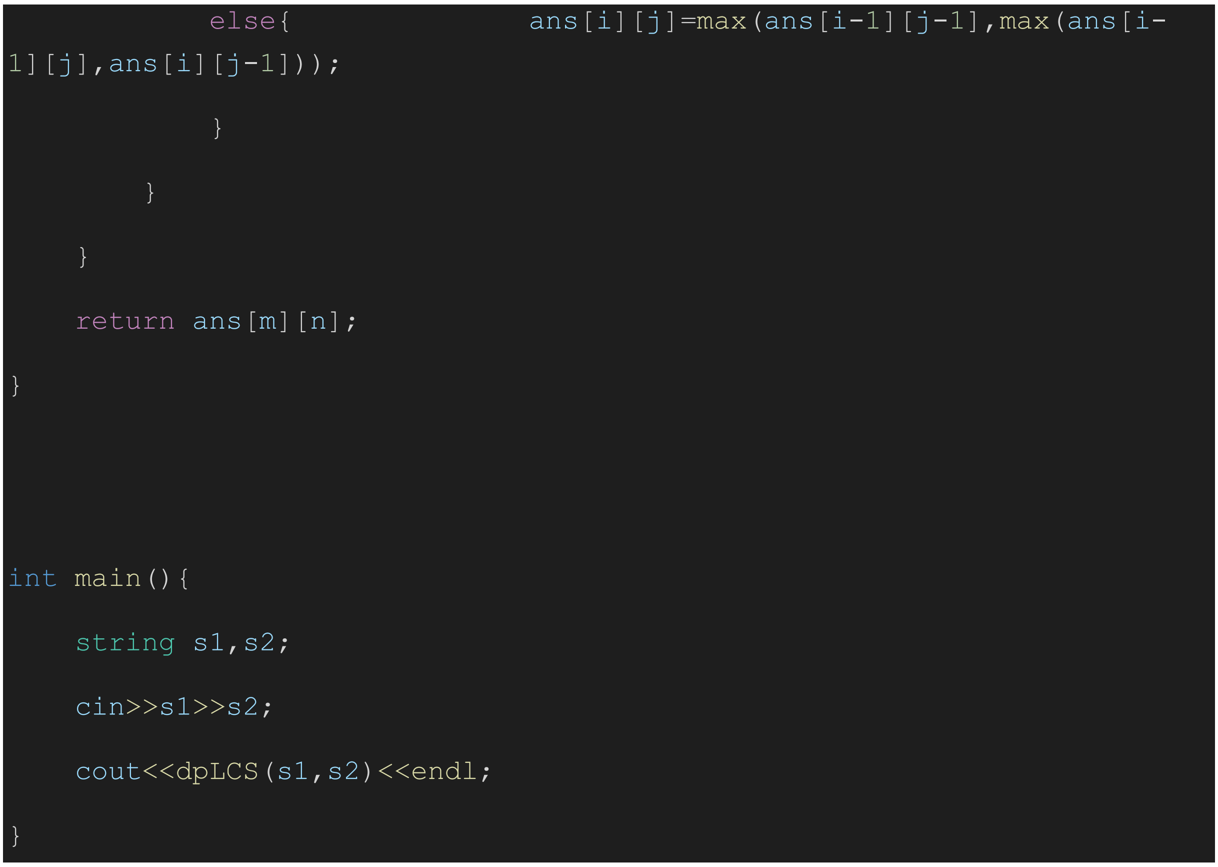


```
2
4 2
1 2
1 3
1
6 6 -1
3 1
2 3
2
-1 6
```

Answer 7

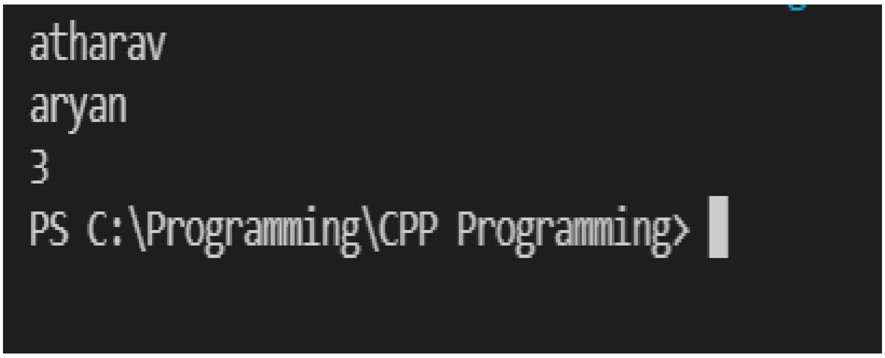
Code

```cpp
#include<bits/stdc++.h>
using namespace std;

int dpLCS(string s1, string s2){

    int m=s1.size();
    int n=s2.size();

    int**ans=new int*[m+1];

    for(int i=0;i<m+1;i++){
        ans[i]=new int [n+1];
    }

    for(int i=0;i<m+1;i++){
        ans[i][0]=0;
    }

    for(int j=0;j<n+1;j++){
        ans[0][j]=0;
    }
    for(int i=1;i<m+1;i++){
        for(int j=1;j<n+1;j++){
            if(s1[m-i]==s2[n-j]){
                ans[i][j]=1+ans[i-1][j-1];
            }
```



```
            else{                    ans[i][j]=max(ans[i-1][j-1],max(ans[i-
1][j],ans[i][j-1]));
            }
        }
    }
    return ans[m][n];
}


int main(){
    string s1,s2;
    cin>>s1>>s2;
    cout<<dpLCS(s1,s2)<<endl;
}
```

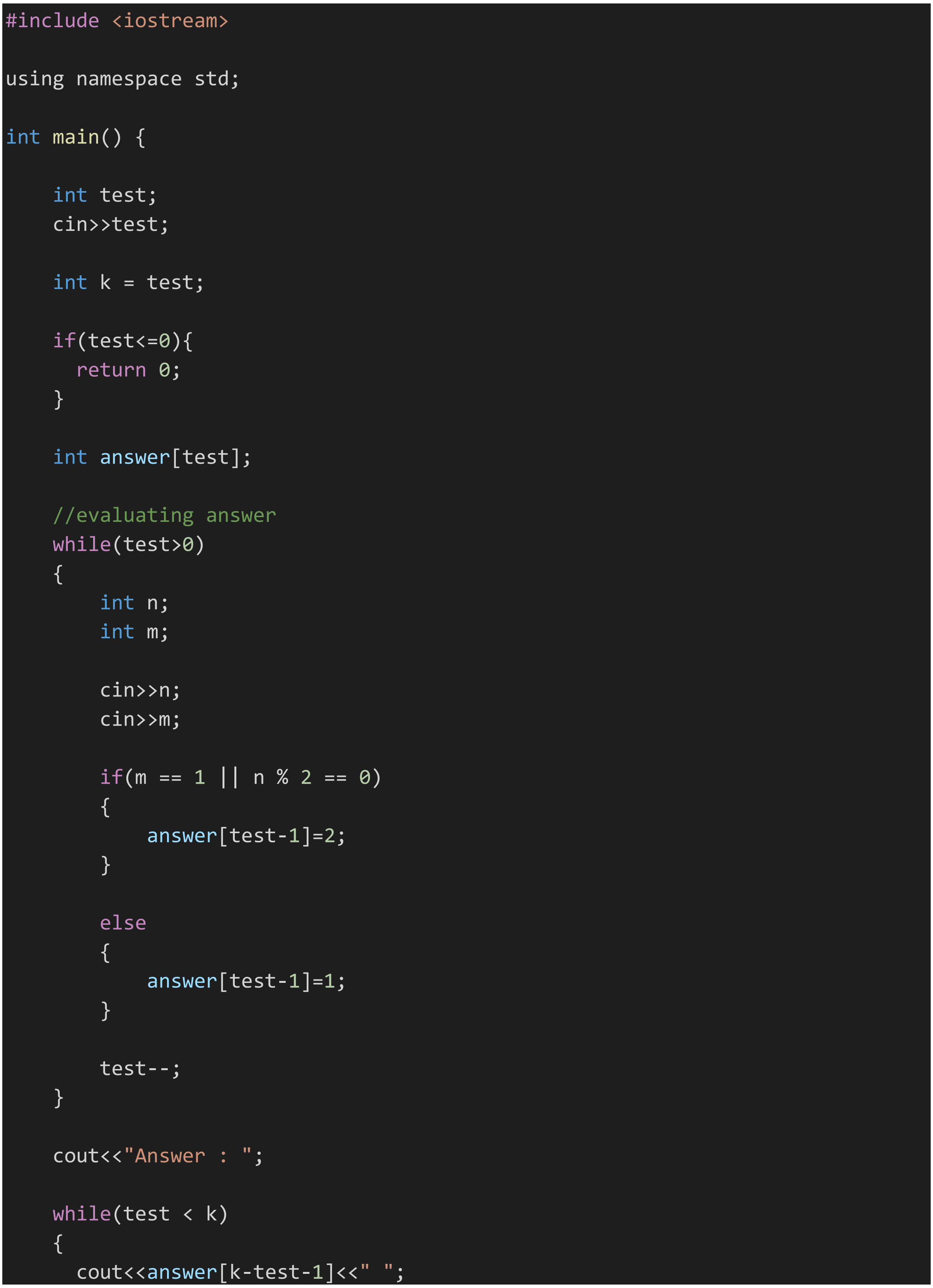
OUTPUT



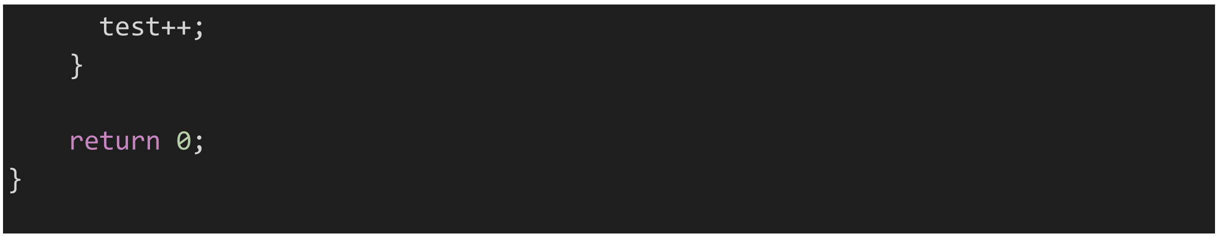
```
atharav
aryan
3
PS C:\Programming\CPP Programming>
```

Answer 8

Code

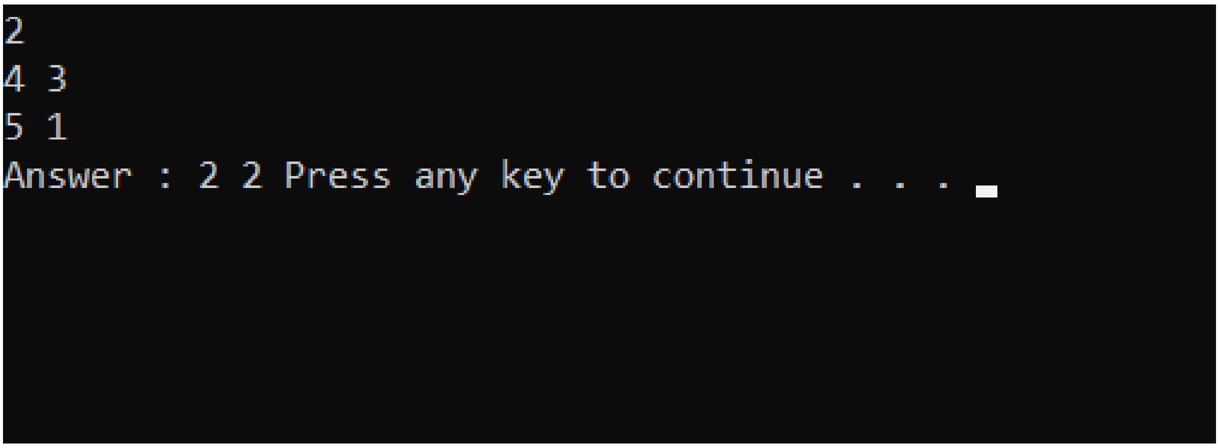


```
#include <iostream>

using namespace std;

int main() {

    int test;
    cin>>test;

    int k = test;

    if(test<=0){
      return 0;
    }

    int answer[test];

    //evaluating answer
    while(test>0)
    {
        int n;
        int m;

        cin>>n;
        cin>>m;

        if(m == 1 || n % 2 == 0)
        {
            answer[test-1]=2;
        }

        else
        {
            answer[test-1]=1;
        }

        test--;
    }

    cout<<"Answer : ";

    while(test < k)
    {
      cout<<answer[k-test-1]<<" ";
```



```
      test++;
    }

    return 0;
}
```

Output



```
2
4 3
5 1
Answer : 2 2 Press any key to continue . . .
```