Assignment 4

A.Jyothi

M22CS007

1. #include <algorithm>

#include <iostream>

using namespace std;

int station(float arrival[], float departure[], int size)

{

    sort(arrival, arrival + size);

    sort(departure, departure + size);

    int need = 1, result = 1;

    int i = 1, j = 0;

    while (i < size && j < size)

    {

        if (arrival[i] <= departure[j])

        {

            need++;

            i++;

        }

        else if (arrival[i] > departure[j])

         {

            need--;

            j++;

        }

        if (need > result)

            result = need;

    }

    return result;

}

int main()

{

    float arrival[] = { 2.05, 2.10, 3.10, 3.20, 3.50, 5.10 };

    float departure[] = { 2.35, 3.40, 3.25, 4.30, 4.10, 5.40 };

    int size = sizeof(arrival) / sizeof(arrival[0]);

    cout << station(arrival, departure, size);

    return 0;

}

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

using namespace std;

int multiply(int array[], int n)

{

    sort(array, array+n);

    return min(array[0]\*array[1]\*array[2], array[0]\*array[n-1]\*array[n-2]);

}

int main()

{

    int array[] = { 4, -1, 4, 5, 9 };

    int size = sizeof(array)/sizeof(array[0]);

    if(size<3)

    {

        cout<<"Invalid..!!"<<endl;

        return 0;

    }

    cout<<"Min product of triplet is : "<<multiply(array, size);

    return 0;

}

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

using namespace std;

void jobs(int start[], int end[], int n)

{

    if(n <= 0)

    {

        cout<<"Invalid..!!"<<endl;

        return;

    }

    int compare = end[0];

    cout<<"Jobs are : "<<endl;

    cout<<"{"<<start[0]<<", "<<compare<<"} ,";

    int i = 1;

    while(i < n)

    {

        if(start[i] > compare)

        {

            compare = end[i];

            cout<<"{"<<start[i]<<", "<<compare<<"}, ";

        }

        i++;

    }

    return;

}

int main(){

    int start[] = {1, 3, 0, 5, 3, 5, 6, 8, 8, 2, 12};

    int end[] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};

    int n = sizeof(end)/sizeof(start[0]);

    jobs(start, end, n);

    return 0;

}

Section 2

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

using namespace std;

int main()

{

    int first\_card,second\_card;

    cout<<"First Friend number of Cards : ";

    cin>>first\_card;

    int A[first\_card];

    cout<<"First Friend card's Strength : ";

    for(int i = 0; i<first\_card; i++)

        cin>>A[i];

    cout<<"\nSecond Friend number of Cards : ";

    cin>>second\_card;

    int B[second\_card];

    cout<<"Second Friend card's Strength : ";

    for(int i = 0; i<second\_card; i++)

        cin>>B[i];

    sort(B,B+second\_card);

    long int sum = 0;

    for(int i = 0; i<first\_card; i++)

    {

        long x = B[second\_card-1]+1;

        if(x-A[i]>0)sum+=x-A[i];

    }

    cout<<"\nAmount of money first friend needed to win game : "<<sum<<endl;

    return 0;

}

Section 3

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

using namespace std;

int maxCutseg(int n, int x, int y, int z)

{

    int s[n + 1];

    memset(s, -1, sizeof(s));

    s[0] = 0;

    for (int i = 0; i < n; i++)

    {

        if (s[i] != -1)

        {

            if(i + x <= n )

                    s[i + x] = max(s[i] + 1, s[i + x]);

            if(i + y <= n )

                    s[i + y] = max(s[i] + 1, s[i + y]);

            if(i + z <= n )

                    s[i + z] = max(s[i] + 1, s[i + z]);

        }

    }

    return s[n];

}

int main()

{

    int n, x, y, z;

    cout<<"Enter n, x, y, z"<<endl;

    cin>>n>>x>>y>>z;

    cout <<"Maximum possible cuts are : "<< maxCutseg(n, x, y, z);

    return 0;

}

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

using namespace std;

bool twoParts(int array[], int n)

{

    int res = 0;

    for(int i = 0; i<n;i++)

    {

        res = res + array[i];

    }

    if(res%2!=0)

        return false;

    res = res / 2;

    bool s[res+1] = {false};

    s[0] = true;

    for(int i=1; i <= n; i++)

    {

        for(int j = res; j>=0; j=j-1)

        {

            if(j-array[i-1] >= 0)

            {

                s[j] = s[j - array[i-1]] or s[j];

            }

        }

    }

    return s[res];

}

int main()

{

    int array[] = {1, 5, 11, 5 };

    int size = sizeof(array)/sizeof(array[0]);

    if(twoParts(array, size))

    {

        cout<<"Yes..!!"<<endl;

        return 0;

    }

    cout<<"No..!!"<<endl;

    return 0;

}

Section 4

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

using namespace std;

int s[50][50];

int product(int\* arr, int i, int j)

{

    if (i == j)

        return 0;

    if (s[i][j] != -1)

        return s[i][j];

    s[i][j] = INT\_MAX;

    for (int k = i; k < j; k++)

        s[i][j] = min( s[i][j], product(arr, i, k) + product(arr, k + 1, j) + arr[i - 1]\*arr[k]\*arr[j] );

    return s[i][j];

}

int MatrixChainOrder(int\* arr, int n){

    int i = 1, j = n - 1;

    return product(arr, i, j);

}

int main()

{

    int arr[] = { 1, 2, 3, 4 };

    int n = sizeof(arr) / sizeof(arr[0]);

    memset(s, -1, sizeof s);

    cout << "Number of product is : "<< MatrixChainOrder(arr, n);

    return 0;

}