Practical-2

AIM: In a far away Galaxy of Tilky Way, there was a planet Tarth where the sport of Competitive Coding was very popular. According to legends, there lived a setter known for loving knapsack type problems. N objects in a row, with weights W1,W2,…,WN, you need to find the maximum number of consecutive objects you can fill in a bag of maximum capacity C such that the total weight of objects taken is at least K. In other words, pick objects such that-The total weight of collected objects is at least K. The total weight does not exceed C. The objects picked must be consecutive (i.e. a subarray of the objects need to be picked) The number of objects is maximized. You need to print this maximum value.

* Program

#include<bits/stdc++.h>

using namespace std;

int main(){

int n, k, c;

int maxi = 0;

int total, count;

cin >> n >> c >> k;

int w[n];

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

cin >> w[i];

}

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

total = 0;

count = 0;

for(int j=i; j<n; j++){

total = total + w[j];

count++;

if(total > c){

break;

}

else if(total >= k and total <= c){

maxi = max(maxi, count);

}

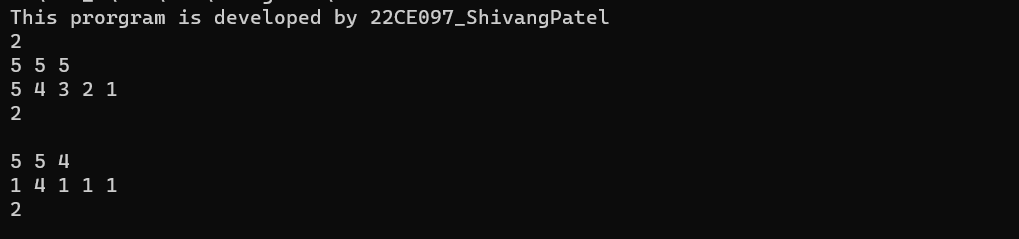
}

}

cout << maxi;

}

Output



* Conclusion

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Student Signature Faculty Signature Marks