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.

Given 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.

**Note: If no such object could be picked, then the answer is obviously 0.**

Input:

* The first line of input contains T, number of test cases in a file.
* The next line contains three integers, N, C and K, as described in the problem statement.
* The next line contains N space separated integers, denoting Wi, i.e.

weight of the object.

Output: For test case, maximum number of objects you can pick. Input

2

5 5 5

5 4 3 2 1

5 5 4

1 4 1 1 1

Output

2

2

Program:

#include<iostream>

using namespace std;

int maximum(int max\_obj,int count)

{

if(max\_obj>count)

{

return max\_obj;

}

else

{

return count;

}

}

void bag(int n,int c,int k,int w[])

{

int i,max\_obj=0;

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

{

int total=0;

int count=0;

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

{

total+=w[j];

count+=1;

if(total>=k && total<=c)

{

max\_obj=maximum(max\_obj,count);

}

if(total>c)

{

break;

}

}

}

cout<<"Maximum number of objects are : "<<max\_obj;

}

int main()

{

cout<<"This program is prepared by JHIL 22CE009\n\n";

int n,c,k;

int w[20];

cout<<"Enter the number of objects :";

cin>>n;

cout<<"Enter the maximum capacity of knapsack :";

cin>>c;

cout<<"Enter the minimum capacity of knapsack :";

cin>>k;

cout<<"Enter the weight of the objects"<<endl;

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

{

cout<<"Enter the weight of object "<<i+1<<": ";

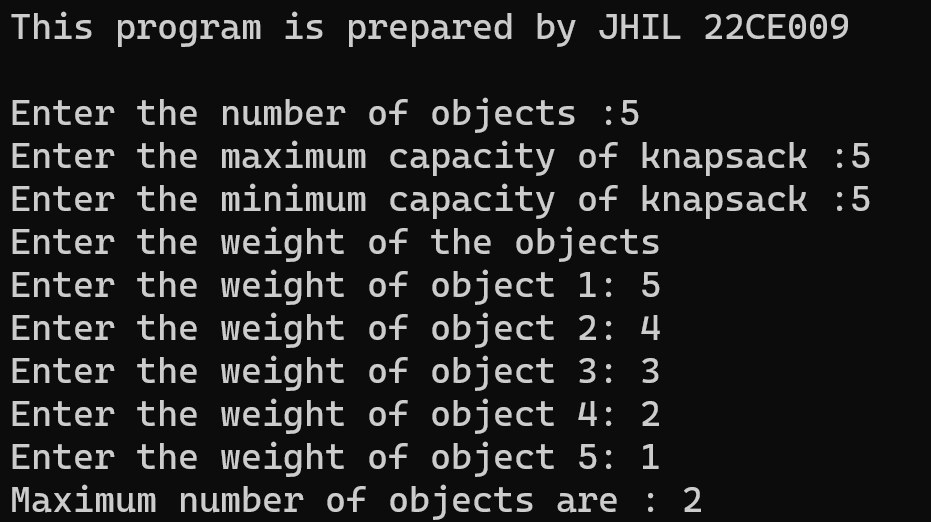
cin>>w[i];

}

bag(n,c,k,w);

}

Output:-



CONCLUSION:-

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

Student Signature Faculty Signature Marks