**Name:** Purvi Singhvi

**Enrollment No.:** 180099

**PROGRAM-15**

**Aim:** Write an algorithm and program to implement 0/1 Knapsack.

**Algorithm:**

**KNAPSACK (n, W)**

1. for w = 0, W

2. do V [0,w] ← 0

3. for i=0, n

4. do V [i, 0] ← 0

5. for w = 0, W

6. do if (wi≤ w & vi + V [i-1, w - wi]> V [i -1,W])

7. then V [i, W] ← vi + V [i - 1, w - wi]

8. else V [i, W] ← V [i - 1, w]

**Source Code:**

#include<stdio.h>

#include<conio.h>

int max(int a, int b){

return (a > b) ? a : b;

}

int knapSack(int W, int wt[], int val[], int n){

if (n == 0 || W == 0)

return 0;

if (wt[n - 1] > W)

return knapSack(W, wt, val, n - 1);

else

return max(

val[n - 1] + knapSack(W - wt[n - 1], wt, val, n - 1),

knapSack(W, wt, val, n - 1));

}

main(){

int val[10],wt[10],i,W,n;

printf("Enter the number of elements (max 10): ");

scanf("%d",&n);

printf("Enter the value\n");

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

scanf("%d",&val[i]);

}

printf("Enter the weight\n");

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

scanf("%d",&wt[i]);

}

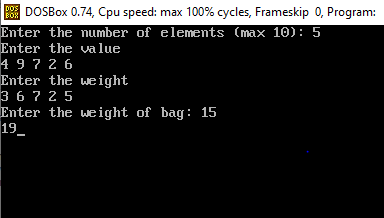
printf("Enter the weight of bag: ");

scanf("%d",&W);

printf("%d", knapSack(W, wt, val, n));

}

**Output:**



**Complexity:**

O(n\*W)

where n=no. of items, W=Capacity of Knapsack