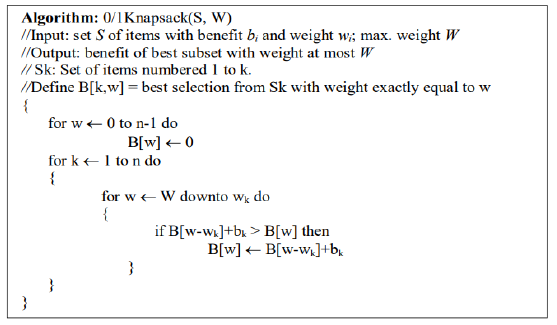
**Program 6:**

Design and implement C/C++ program to solve 0/1 Knapsack Problem using Dynamic Programming method.

**Algorithm:**

****

**Code:**

#include<stdio.h>

int n,m,p[10],w[10];

int max(int a, int b)

{

return(a>b?a:b);

}

void knapsack\_DP()

{

int V[10][10],i,j;

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

for(j=0;j<=m;j++)

if(i==0||j==0)

V[i][j]=0;

else

if(j<w[i])

V[i][j]=V[i-1][j];

else

V[i][j]=max(V[i-1][j],p[i]+V[i-1][j-w[i]]);

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

{

for(j=0;j<=m;j++)

printf("%d ",V[i][j]);

printf("\n");

}

printf("Items included are: ");

while(n>0)

{

if(V[n][m]!=V[n-1][m])

{

printf("%d ",n);

m=m-w[n];

}

n--;

}

}

int main()

{

int i;

printf("Enter the number of items: ");

scanf("%d",&n);

printf("Enter the weights of n items: ");

for(i=1;i<=n;i++)

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

printf("Enter the prices of n items: ");

for(i=1;i<=n;i++)

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

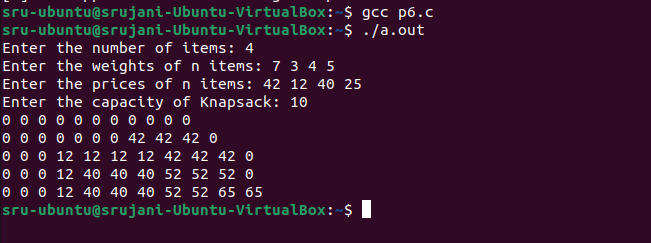
printf("Enter the capacity of Knapsack: ");

scanf("%d",&m);

knapsack\_DP();

}

**Output:**

****