TUNEET M.K
1BM19CS125

ADA LAB TEST - 2

⇒ Implement 0/1 knapsack problem using dynamic programming
Modification :- give the count of the items selected.

PROGRAM:-

```
#include <stdio.h>
int max(int, int);
int m, i, j, n, p[10], w[10], v[10][10], x[10], op_soln;
int knapsack();
void object_selected();
void main()
{
printf("Enter the number of objects\n");
  scanf("%d",&n);
printf("Enter the weights of N objects\n");
  for(i=1; i<=n; i++)
  scanf("%d", &w[i]);
printf("Enter the profit of N objects\n");
  for(i=1; i<=n; i++)
  scanf("%d", &p[i]);
  printf("Enter the capacity of knapsack\n");
  scanf("%d", &m);
  op_soln = knapsack(n, w, m, v, p);
  printf("The output is\n");
  for(i=0; i<=n; i++)
  {
  for(j=0; j<=m; j++)
  {
  printf("%d\t", v[i][j]);
  }
```

```
printf ("\n");
}
printf ("Optimal Solution = %d \n", op_soln);
object_selected ();
}
int max (int a, int b)
{
return (a>b ? a:b);
}
int knapsack ()
{
int 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 (w[i] > j)
v[i][j] = v[i-1][j];
else
v[i][j] = max(v[i-1][j], v[i-1][j - w[i]] + p[i]);
}
}
}
return v[n][m];
}
void object_selected ()
{
int count=0;
i=n;
j=m;
```

```
while (i != 0 && j != 0)
{
  if (v[i][j] != v[i-1][j])
  {
    x[i] = 1;
    j = j - w[i];
  }
  i--;
}
printf("objects selected \n");
for (i=1; i<=n; i++)
{
  if (x[i] == 1){
    printf("%d \t", i);
    count++;
  }
}
printf("\n");
printf("Total number of Items selected : %d", count);
}
```

Output:-

```
Enter the number of objects
4
Enter the weights of N objects
2 4 3 2
Enter the profits of N objects
12 10 20 15
Enter capacity of knapsack
5
```

The output is

0	0	0	0	0	0
0	0	12	12	12	12
0	0	12	12	12	12
0	0	12	20	20	32
0	0	15	20	27	35

Optimal Solution

35

Objects selected are

3 4

Total number of items selected : 2