```
Code:
#include <stdio.h>
#include <conio.h>
void knapsack();
int max(int, int);
int i, j, n, m, p[10], w[10], v[10][10];
void main()
  printf("\nEnter the no. of items:\n");
  scanf("%d", &n);
  printf("\nEnter the weight of the each item:\n");
  for (i = 1; i \le n; i++)
  {
     scanf("%d", &w[i]);
  printf("\nEnter the profit of each item:\n");
  for (i = 1; i \le n; i++)
     scanf("%d", &p[i]);
  printf("\nEnter the knapsack's capacity:\n");
  scanf("%d", &m);
  knapsack();
  getch();
void knapsack()
  int x[10];
  for (i = 0; i \le n; i++)
  {
     for (j = 0; j \le m; j++)
        if (i == 0 || j == 0)
        {
           v[i][j] = 0;
        else if (j - w[i] < 0)
           v[i][j] = v[i - 1][j];
        }
        else
```

```
v[i][j] = max(v[i - 1][j], v[i - 1][j - w[i]] + p[i]);
        }
     }
   }
  printf("\nThe output is:\n");
  for (i = 0; i \le n; i++)
  {
     for (j = 0; j \le m; j++)
        printf("%d ", v[i][j]);
     printf("\n\n");
  printf("\nThe optimal solution is %d", v[n][m]);
  printf("\nThe solution vector is:\n");
  for (i = n; i >= 1; i--)
     if (v[i][m] != v[i - 1][m])
        x[i] = 1;
        m = m - w[i];
     }
     else
        x[i] = 0;
   }
  for (i = 1; i \le n; i++)
     printf("%d\t", x[i]);
  }
int max(int x, int y)
  if (x > y)
     return x;
  }
   else
     return y;
}
```

Output:

Observation:

```
Sort matted a) implement of knapsack. problem using dynamic Programming
          Loilloil , Coison , m, PCIOI, vorio, V[10];
          void knapsack ()
          ( Code to !
             for (1=0;) (=n; 1++)
             1 for (j=0; j <= m; j++)
               1 i+(i==0 11 i==0)
                     10 = (!)[!)~
                 else it (i - wai) (0)
                    1 [1][1-1] = 1[1-1][1]
                 else
                     (([1]) + [[1] w - [][1-1][] , V[1-1][] + P(1]);
           print & (" in The output is: in");
            for (1=0; ( <= n; (++)
               tor (j=0; j <= m; j++)
                 print + (" bolo ", vEIDCIJ);
                printf("mnn");
            print + 1" more optimal solution is "lod", VEND (m);
            print ("In The Solution vector is: 1m");
            for (1=0; 17=131--)
                (Cm)C1-1) v = (Cm) [1)v) +1
                  x(1)=1) 0A
                   i Cisa-m=m
                 else &
                    , 0= [i)x
              for (1=1] i(=n; i++)
                  Printel "by by how ) the Man
         (y tui, x tui) rong tui
             (2 (x) 4;
                return ";
              else return y;
```

```
toid mounts of
  1 10 1 (10)
   print of " in Enter the no. of ite rus: m");
   print ( 9th Enter the weight of out item: mu);
   for ( = 1) 1 = = 1 ( + + )
      Scan+ (440db, 4w(1));
   print+("In Futer profit of each item: in");
     Acr (1= 11 12= n; 1++)
         scant (world", GP(i));
   print & (" MEntor Knapsack's capacing: (");
   santinopod", 6m);
   Knap sacking
    getch ()
  1. [139 + [CHON : 3CH-13V, (DEL-DV) 11000 = 622 13V
output:
  Enter the noig items: 4
  Enter the weight of each item: 21
 Enter the profit of each item: 12 15 25 10
  Enter the knapsack's corpality: 5
  The output is:
            0 0 0 0
    0 0
    0 0 12 12 12 12.
                   27 27
               27
       15 15 27 40 40
    0 15 15 27 40 40.
  The Optimal Solution 11 40
  The solution vector is:
```