```
// (9)-Aim: Write  a program to solve 0/1 Knapsack problem using DynamicProgramming
// Program:
#include<stdio.h>
#include<conio.h>
#define MAX 100
int main()
{
int n,flag[MAX]={0},v[MAX],w[MAX],m[MAX][MAX],W,i,j,k;
clrscr();
printf(”Enter the number of elements: ”);
scanf(”%d”,&n);
printf(”Enter the values: ”);
for(i=1;i<=n;i++)
scanf(”%d”,&v[i]);
printf(”Enter the weights: ”);
for(i=1;i<=n;i++)
scanf(”%d”,&w[i]);
printf(”Enter the capacity of knapsack: ”);
scanf(”%d”,&W);
for(j=0;j<=W;j++)
m[0][j]=0;
for(i=1;i<=n;i++)
{
for(j=0;j<=W;j++)
{
if(w[i]<=j)
{
if( m[i-1][j] > (m[i-1][j-w[i]]+v[i]) )
m[i][j]=m[i-1][j];
else
m[i][j]=m[i-1][j-w[i]]+v[i];
}
else
m[i][j]=m[i-1][j];
}
}
i=n;
k=W;
while(i>0 && k>0)
{
if(m[i][k]!=m[i-1][k])
{
flag[i]=1;
k=k-w[i];
i=i-1;
}
else
i--;
}
printf(”\n\t”);
for(i=0;i<=W;i++)
printf(”%d\t”,i);
printf(”\n”);
for(i=0;i<=10*W;i++)
printf(”-”);
printf(”\n”);
for(i=0;i<=n;i++)
{
printf(”%d |\t”, i);
for(j=0;j<=W;j++)
printf(”%d\t”,m[i][j]);
printf(”\n”);
}
printf(”\nThe resultant vector is ”);
printf(”( ”);
for(i=1;i<=n;i++)
printf(”%d ”,flag[i]);
printf(”)”);
printf(”\n\nThe total profit is %d”,m[n][W]);
printf(”\n”);
getch();
return 0;
}
```