**ADA LAB-12**

* **Knapsack problem using dynamic programming.**
  + - * **Program**

#include<stdio.h>

#include<conio.h>

int i,j,n,m,p[10],w[10],v[10][10];

int max(int x,int y)

{

if(x>y)

{

return x;

}

else

{

return y;

}

}

void knapsack()

{

int x[10];

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]<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("\noutput:\n");

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

{

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

{

printf("%d\t",v[i][j]);

}

printf("\n\n");

}

printf("\noptimal solution: %d\n",v[n][m]);

printf("\nsolution vector:\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<=n;i++)

{

printf("%d\t",x[i]);

}

}

int main()

{

printf("\nenter the no. of items:\t");

scanf("%d",&n);

printf("\nenter the weight of the each item:\n");

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

{

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

}

printf("\nenter the profit of each item:\n");

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

{

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

}

printf("\nenter the knapsack's capacity:\t");

scanf("%d",&m);

knapsack();

getch();

}

* + - * **Output**



