8-07-21

ADA - Lab Test-2

3. Implement 0/1 knapsack problem using dynamic programming

```
#include <stdio.h>
#include <conio.h>
int max (int a, int b)
 {
  if a>b
   return a;
   else return b;
 }
void knapsack (int w[], int v[], int s, int n)
 {
   int k[n+1][s+1];
   int i, j, res=0;
   int count = 0, weight=0;
   for (i=0; i<=n ; i++)
   for (j=0 ; j<=s; j++)
     {
       if (i==0 || j==0)
         k[i][j] = 0;
       else if (w[i-1] <= j)
       k[i][j] = max (v[i-1]+k[i-1][j-w[i-1]]k[i-1][j]);
       else
       k[i][j] = k[i-1][j];
```

3 0

```
res = k[n][s];
printf("\n Maximum value that can be obtained
    is : %d", res);
j = s;
printf("\n And the objects with the respective
    weights selected are :");
);
for (i=n; i>0 && res>0; i--)
{
    if (res == k[i-1][j])
        continue;
    else
    {
printf("%d", w[i-1]);
res = res - v[i-1];
j = j - w[i-1];
count++;
weight = weight + w[i-1];
}
}
printf("\n The count of the items selected is : %d
    count);
printf("\n The Total weight of the items
    selected : %d", weight);
}
```

```
int main( )
{
 int w[10], v[10], s, n, i;
 printf ("\n Enter the numbers of objects :");
 scanf ("%d", &n);
 printf ("\n Enter the weights of the objects :");
 for (i=0; i<n; i++)
  scanf ("%d", &w[i]);
 printf ("\n Enter the value of the objects :");
  for (i=0; i<n; i++)
  scanf ("%d", &v[i]);
  printf ("\n Enter the size of knapsack :");
  scanf ("%d", &s);
  knapsack (w, v, s, n);
  }
```