

```

1  #include<stdio.h>
2
3  int maximum(int a , int b)
4  {
5      if (a>=b)
6          return a;
7      else
8          return b;
9  }
10 int Knapsack(int c[],int wt[],int n, int W)
11 {
12     int i,j,k,l;
13
14     int M[n+1][W+1];
15
16     for(i=0;i<=n;i++)
17     {
18         for(j=0;j<=W;j++)
19         {
20             if(i==0 || j==0)
21                 M[i][j]=0;
22             else if(wt[i-1]<=j)
23             {
24
25                 M[i][j]=maximum(M[i-1][j],M[i-1][j-wt[i-1]]+c[i-1]);
26             }
27             else
28                 M[i][j]=M[i-1][j];
29         }
30     }
31     for(i=0;i<=n;i++)
32     {
33         for(j=0;j<=W;j++)
34         {
35             printf("%d\t",M[i][j]);
36         }
37         printf("\n");
38     }
39 }
40
41 return M[n][W];
42 }
43 void main()
44 {
45     int val[] = {60, 100, 120};
46     int wt[] = {8, 20, 30};
47     int W = 58;
48     int n = sizeof(val)/sizeof(val[0]);
49     printf("final ans is %d", Knapsack(val,wt,n,W));
50 }

```