

[Send Feedback](#)

Line 1 : N i.e. number of items
Line 2 : N Integers i.e. weights of items separated by space
Line 3 : N Integers i.e. values of items separated by space
Line 4 : Integer W i.e. maximum weight thief can carry

Line 1 : Maximum value V

 $1 \leq N \leq 10^4$ $1 \leq w_i \leq 100$ $1 \leq v_i \leq 100$

4
1 2 4 5
5 4 8 6
5

13

```

1  #include<cstring>
2  using namespace std;
3  int knapsack(int* wt, int* val, int n, int W){
4
5  // matrix to store final result
6  int mat[2][W+1];
7  memset(mat, 0, sizeof(mat));
8
9  // iterate through all items
10 int i = 0;
11 while (i < n) // one by one traverse each element
12 {
13     int j = 0; // traverse all weights j <= W
14
15     // if i is odd that mean till now we have odd
16     // number of elements so we store result in 1th
17     // indexed row
18     if (i%2!=0)
19     {
20         while (++j <= W) // check for each value
21         {
22             if (wt[i] <= j) // include element
23                 mat[1][j] = max(val[i] + mat[0][j-wt[i]],
24                                 mat[0][j]);
25             else // exclude element
26                 mat[1][j] = mat[0][j];
27         }
28     }
29
30     // if i is even that mean till now we have even number
31     // of elements so we store result in 0th indexed row
32     else
33     {
34         while(++j <= W)
35         {
36             if (wt[i] <= j)
37                 mat[0][j] = max(val[i] + mat[1][j-wt[i]],

```

