

1. 程式碼

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int max(int a, int b)
6  {
7      return (a > b) ? a : b;
8  }
9
10 int knapSack(int W, int weight[], int p[], int pw)
11 {
12     if (pw == 0 || W == 0)
13         return 0;
14     if (weight[pw - 1] > W)
15         return knapSack(W, weight, p, pw - 1);
16
17     else
18         return max(
19             p[pw - 1]
20             + knapSack(W - weight[pw - 1],
21                       weight, p, pw - 1),
22             knapSack(W, weight, p, pw - 1));
23 }
24
25
26 int main()
27 {
28     int p[] = { 20, 30, 35, 12, 3 };
29     int weight[] = { 2, 5, 7, 3, 1 };
30     int W = 9;
31     int pw = sizeof(p) / sizeof(p[0]);
32     printf("%d", knapSack(W, weight, p, pw));
33     return 0;
34 }
```

2.輸出結果

```
PS C:\Users\7RED\Desktop\C - 複製> ./Knapsack.exe  
55
```

3.解釋

第一段副程式 `max` 用來判斷拿與不拿誰的價值高。

第二段副程式 `knapsack` 將所有值輸入後進行判斷，選出最高價值並且不會過重的結果回傳。

主程式輸入所有變數以及呼叫副程式後得到結果。