/*0/1 knapsack(not global)*/

```
#include<stdio.h>

int knapsack_0_1(int w[], int p[], int c, int n);

int main() {
   int i, n, c;
   printf("Enter the array size for both weight & profit:");
   scanf("%d", &n);
   int w[n], p[n];
   printf("\nEnter the weights and profits");
   for (i = 0; i < n; i++) {
      printf("\nWeight[%d]:", i + 1); //id starting from 0 but showing it's starting from 1
      scanf("%d", &w[i]);
      printf("Profit[%d]:", i + 1);  //id starting from 0 but showing it's starting from 1
      scanf("%d", &p[i]);
   }
   printf("\nEnter the capacity:");
   scanf("%d", &c);
   int max = knapsack_0_1(w, p, c, n);
   printf("Max profit: %d\n", max);

   return 0;
}

int knapsack_0_1(int w[], int p[], int c, int n) {
   int i, j,ct=0;  // Initialize the selected item count
   int ks[n + 1][c + 1];

   for (i = 0; i <= c; i++) {
      ks[0][i] = 0;
```

```
}

for (i = 0; i <= n; i++) {
   ks[i][0] = 0;
}

for (i = 1; i <= n; i++) {  //row       /*for each cols
   for (j = 1; j <= c; j++) {   //col     of each rows the loop is iterating*/
      if ((w[i - 1] <= j) && ((p[i - 1] + ks[i - 1][j - w[i - 1]]) > ks[i - 1][j])) {
         ks[i][j] = p[i - 1] + ks[i - 1][j - w[i - 1]];
      } else {
         ks[i][j] = ks[i - 1][j];
      }
   }
}

printf("The matrix is\n");
for (i = 0; i <= n; i++) {
   for (j = 0; j <= c; j++) {
      printf("%d ", ks[i][j]);
   }
   printf("\n");
}

// Backtrack to find the selected items
i = n;
j = c;
while (i > 0 && j > 0) {
   if (ks[i][j] != ks[i - 1][j]) {
      printf("Object %d selected (Weight: %d, Profit: %d)\n", i, w[i - 1], p[i - 1]);
      j -= w[i - 1];
```

```
            i--;
        } else {
            i--;
        }
    }

    return ks[n][c];
}
```