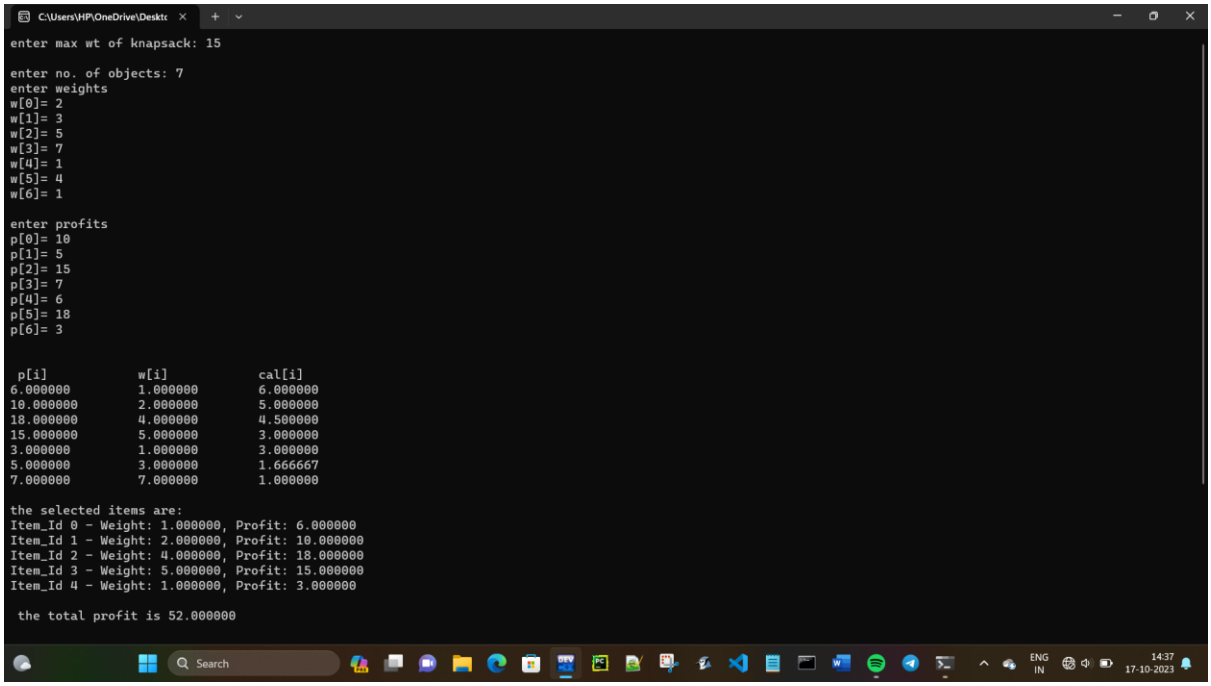
/*knapsack*/

```
#include <stdio.h>

void knapsack(float cal[], float p[], float w[]);

int m = 0, n = 0;

int main() {
   int i;
   float cal[20], p[20], w[20];
   printf("enter max wt of knapsack: ");
   scanf("%d", &m);
   printf("\nenter no. of objects: ");
   scanf("%d", &n);

   printf("enter weights\n");
   for (i = 0; i < n; i++) {
      printf("w[%d]= ", i);
      scanf("%f", &w[i]);
   }

   printf("\nenter profits\n");
   for (i = 0; i < n; i++) {
      printf("p[%d]= ", i);
      scanf("%f", &p[i]);
   }
   knapsack(cal, p, w);
   return 0;
}

void knapsack(float cal[], float p[], float w[]) {
```

```
int i, j;
float total_profit = 0;
int selected_items[20]; // To keep track of selected items
int ct = 0; // Initialize the selected item count

for (i = 0; i < n; i++) {
   cal[i] = p[i] / w[i];
}

// Sort the items based on the ratio of profit to weight (cal)
for (i = 0; i < n; i++) {
   for (j = i + 1; j < n; j++) {
      if (cal[i] < cal[j]) {
         float temp = cal[i];
         cal[i] = cal[j];
         cal[j] = temp;

         temp = w[i];
         w[i] = w[j];
         w[j] = temp;

         temp = p[i];
         p[i] = p[j];
         p[j] = temp;
      }
   }
}

printf("\n\n p[i]\t\t w[i]\t\t cal[i]\n");
for (i = 0; i < n; i++) {
   printf("%f\t %f\t %f\t\n", p[i], w[i], cal[i]);
```

```
    }

    for (i = 0; i < n; i++) {
        if (m > 0 && w[i] <= m) {
            m -= w[i];
            total_profit += p[i];
            selected_items[ct++] = i; // Store the index of the selected item
        } else {
            break;
        }
    }

    // If there is still space in the knapsack, add a fraction of the next item
//    if (m > 0 && i < n) {
//        float fraction = (float)m / w[i];
//        total_profit += p[i] * fraction;
//        selected_items[ct] = i; // Store the index of the selected item
//    }

    printf("\nthe selected items are: \n");
    for (i = 0; i < ct; i++) {
        printf("Item_Id %d - Weight: %f, Profit: %f\n", selected_items[i], w[selected_items[i]],
p[selected_items[i]]);
    }
    printf("\n the total profit is %f\n", total_profit);
}
```


C:\Users\HP\OneDrive\Deskto
enter max wt of knapsack: 15
enter no. of objects: 7
enter weights
w[0]= 2
w[1]= 3
w[2]= 5
w[3]= 7
w[4]= 1
w[5]= 4
w[6]= 1
enter profits
p[0]= 10
p[1]= 5
p[2]= 15
p[3]= 7
p[4]= 6
p[5]= 18
p[6]= 3
p[i] w[i] cal[i]
6.000000 1.000000 6.000000
10.000000 2.000000 5.000000
18.000000 4.000000 4.500000
15.000000 5.000000 3.000000
3.000000 1.000000 3.000000
5.000000 3.000000 1.666667
7.000000 7.000000 1.000000
the selected items are:
Item_Id 0 - Weight: 1.000000, Profit: 6.000000
Item_Id 1 - Weight: 2.000000, Profit: 10.000000
Item_Id 2 - Weight: 4.000000, Profit: 18.000000
Item_Id 3 - Weight: 5.000000, Profit: 15.000000
Item_Id 4 - Weight: 1.000000, Profit: 3.000000
the total profit is 52.000000
Search
ENG
IN
14:37
17-10-2023