Experiment 5

Fractional Knap Sack using Greedy Method

Program:

```
#include <stdio.h>

int n = 5; /* The number of objects */
int c[10] = {12, 1, 2, 1, 4}; /* c[i] is the *COST* of the ith object; i.e. what
        YOU PAY to take the object */
int v[10] = {4, 2, 2, 1, 10}; /* v[i] is the *VALUE* of the ith object; i.e.
        what YOU GET for taking the object */
int W = 15; /* The maximum weight you can take */

void simple_fill() {
   int cur_w;
   float tot_v;
   int i, maxi;
   int used[10];

   for (i = 0; i < n; ++i)
      used[i] = 0; /* I have not used the ith object yet */

   cur_w = W;
   while (cur_w > 0) { /* while there's still room*/
      /* Find the best object */
      maxi = -1;
      for (i = 0; i < n; ++i)
         if ((used[i] == 0) &&
            ((maxi == -1) || ((float)v[i]/c[i] > (float)v[maxi]/c[maxi])))
            maxi = i;

      used[maxi] = 1; /* mark the maxi-th object as used */
      cur_w -= c[maxi]; /* with the object in the bag, I can carry less */
      tot_v += v[maxi];
      if (cur_w >= 0)
         printf("Added object %d (%d$, %dKg) completely in the bag. Space left: %d.\n", maxi +
1, v[maxi], c[maxi], cur_w);
      else {
         printf("Added %d%% (%d$, %dKg) of object %d in the bag.\n", (int)((1 +
(float)cur_w/c[maxi]) * 100), v[maxi], c[maxi], maxi + 1);
         tot_v -= v[maxi];
         tot_v += (1 + (float)cur_w/c[maxi]) * v[maxi];
```

```
        }
    }

    printf("Filled the bag with objects worth %.2f$.\n", tot_v);
}

int main(int argc, char *argv[]) {
    simple_fill();

    return 0;
}
```

Output:

```
Added object 5 (10$, 4Kg) completely in the bag. Space left: 11.
Added object 2 (2$, 1Kg) completely in the bag. Space left: 10.
Added object 3 (2$, 2Kg) completely in the bag. Space left: 8.
Added object 4 (1$, 1Kg) completely in the bag. Space left: 7.
Added 58% (4$, 12Kg) of object 1 in the bag.
Filled the bag with objects worth 17.33$.


=== Code Execution Successful ===
```