Market assignment

Y. Moustaghfir & S. S. Hamed
S2909758 & S2562677

September 15, 2015

1 Problem description

The program is required to print all the possible options of buying exactly 100 pieces of fruit with exactly 100 euros. We can choose between oranges, grapefruits and melons. The input of the program is the price of each fruit(orange, grapefruit and melons) in eurocents.

2 Problem analysis

We interpret the problem as follows: we first have to loop over the first kind of fruit. This means that we have to loop for example to 100 pieces of oranges. Within that first loop(let's call it the orange loop), we have to loop over the second kind of fruit(the grapefruit loop). After that we would have to make sure that the leftover fruit is set to melons, so that we can analyze the amount and price of that fruit as well. This interpretation is motivated by the following statements:

3 Design

First the prices for the oranges, grapefruits and melons are read from the standard input and stored (in the variables `op`, `gp` and `mp`). After that we begin with the first for loop with the variable `obought` which stands for the amount of oranges. Within the orange loop, we start another for loop with the variable `gbought`, which stands for the amount of grapefruit. At last we calculate the amount of melons, which is:

$$100 - obought - gbought$$

We store that value in the variable `mbought`. After we stored all the values of the amount of fruit, we begin to calculate the total amount of fruit and store that value in the variable
tt total. We also need the total price of the fruit to calculate if we have spent exactly 100 euros. This value is stored in the variable `totalp` with the expression:

$$totalp = (gbought * gp) + (obought * op) + (mbought * mp)$$

After all this looping, we start to select the statements we want to print. Each if statement checks if the price is exactly 10000 cents(because our input is in cents) and the total amount of fruit is 100. Each if statement prints a certain plural or singular form of the noun of the fruit. The if-statement check all the possibilities and print out the correct form of the noun(singular or plural). The oranges and grapefruit are unable to reach a value under 0, but
tt mbought is able to reach a value under 0. That is why a lot of if-statement checks also if the value of `mbought` is not smaller than 0.

4 Program code

market.c

```
1  /* File : market.c
2   * Author: Younes Moustaghfir
3   */
4
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(int argc, char *argv[]) {
9     int obought, gbought, mbought, op, gp, mp, total, totalp;
10
11    printf("price of an orange: ");
12    scanf("%d", &op);
13    printf("price of a grapefruit: ");
14    scanf("%d", &gp);
15    printf("price of a melon: ");
16    scanf("%d", &mp);
17
18    for(obought = 0; obought <= 100; obought++) {
19       for(gbought = 0; gbought <= 100; gbought++) {
20          mbought = 100 - obought - gbought;
21          total = obought + gbought + mbought;
22          totalp = (gbought*gp) + (obought*op) + (mbought*mp);
23          if(totalp == 10000 && total == 100 && obought != 1 && gbought != 1 && mbought !=
                1 && !(mbought < 0)) {
24             printf("%d oranges, ", obought);
25             printf("%d grapefruits, ", gbought);
26             printf("%d melons\n", mbought);
27          }
28          else if(totalp == 10000 && total == 100 && obought == 1 && mbought != 1 &&
                gbought != 1 && !(mbought < 0)) {
29             printf("%d orange, %d grapefruits, %d melons\n", obought, gbought, mbought);
30          }
31          else if(totalp == 10000 && total == 100 && mbought == 1 && gbought != 1 &&
                obought != 1) {
32             printf("%d oranges, %d grapefruits, %d melon\n", obought, gbought, mbought);
33          }
34          else if(totalp == 10000 && total == 100 && gbought == 1 && mbought != 1 &&
                obought != 1 && !(mbought < 0)) {
35             printf("%d oranges, %d grapefruit, %d melons\n", obought, gbought, mbought);
36          }
37          else if(totalp == 10000 && total == 100 && gbought == 1 && obought == 1 &&
                mbought != 1 && !(mbought <0)) {
38             printf("%d orange, %d grapefruit, %d melons\n", obought, gbought, mbought);
39          }
40          else if(totalp == 10000 && total == 100 && gbought == 1 && mbought == 1 &&
                obought != 1) {
41             printf("%d oranges, %d grapefruit, %d melon\n", obought, gbought, mbought);
42          }
43          else if(totalp == 10000 && total == 100 && obought == 1 && mbought == 1 &&
                gbought != 1) {
44             printf("%d orange, %d grapefruits, %d melon\n", obought, gbought, mbought);
45          }
46          else if(totalp == 10000 && total == 100 && obought == 1 && gbought == 1 &&
                mbought == 1) {
```

```
47              printf("%d orange, %d grapefruit, %d melon\n", obought, gbought, mbought);
48          }
49       }
50    }
51 }
```

5 Test results

- Input: (prices of the oranges, grapefruits and melons in cents)

```
88
99
102
```

Output:

```
1 orange, 62 grapefruits, 37 melons
4 oranges, 48 grapefruits, 48 melons
7 oranges, 34 grapefruits, 59 melons
10 oranges, 20 grapefruits, 70 melons
13 oranges, 6 grapefruits, 81 melons
```

- Input: (prices of the oranges, grapefruits and melons in cents)

```
0 0
4 8
9 1
5 8
```

Output:

```
The lines do not intersects each other.
```

- Input: (prices of the oranges, grapefruits and melons in cents)

```
0 0
9 9
5 5
9 1
```

Output:

```
The lines do not intersects each other.
```

6 Evaluation

The program was in fact easy to design to pass simple tests like the first test case, but when we had to check for, for example the value of `mbought`, the program logic of the if-statements became somewhat more complex. At first we had some trouble exactly figuring out what the exact logic should be, but with some help and a lot of thought we got the program to pass all the testcases from Justitia.

We have chosen to write down all the possible statements that we could print instead of splitting the printf statements. We could have checked for each part of the printf statement if it should be a plural or singular form.