Report For Ansys

Name:Wayne
Platform: Ubuntu 16.04
IDE: Eclipse for C++
C++ Version: C++ 11

Part1.Pre-condition.

Firstly, I am confused with the input of this program.

For example, if the variable amountPaid is 30$, there are many different combinations, such as 10$,10$,10$ or 20$ 10$ or 20$,5$,5$, the combination of the input will affect the number of coins in the till.

However, in the real life, salesperson usually return the difference for one specific banknote. **So I assume the customer always pay with one coin/bill.**

The other problem is that we don’t know the number of coins in till when initialization. So, **I assume the initialized number is 10 for every coins.**

Part2.Algorithm

I applied the greedy algorithm to solve the problem. The algorithm will select the biggest one in till if the biggest one exist. There are several test cases:

Case1:
Paid 100$ Own 50$ change 50$
The coins in till: <100$,0><50$, 1> <20$, 2> <10$, 2>
The output: 1 50$
The till will change: <100$,1> <50$, 0> <20$,2><10$,2>

Case2:
Paid 100$, Own 50$, Change 50$
The coins in till: <100$,1> <50$, 0> <20$,2><10$,2>
The output: 2 20$, 1 10$
The till will change: <100$,2> <50$, 0> <20$,0><10$,1>

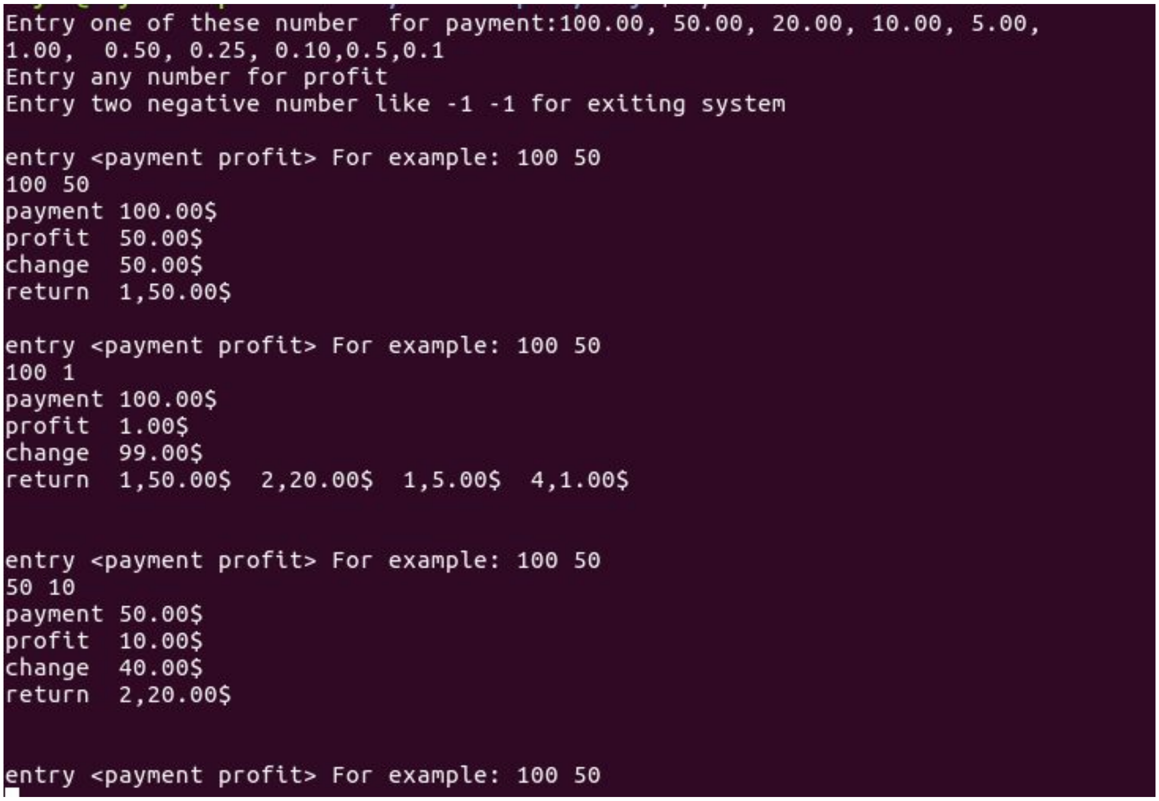
Case3: the total money are not enough.
Paid 100$, Own 50$, Change 50$
The coins in till: <100$,0> <50$, 0> <20$,0><10$,1>
Out:the money is not enough

Case4:The total money is enough but there is no suitable coins.
Paid 100$, Own 50$, Change 50$
The coins in till: <100$,1> <50$, 0> <20$,1><10$,1>
Output:money is enough but coins/bills are not enough

Part3, Improvement

I don’t know the combination of the payment, so I assume customer just can only use one coin/bill to pay.
if customer wants to make payment become random, just remove related code.

Part4:result



```
Entry one of these number  for payment:100.00, 50.00, 20.00, 10.00, 5.00,
1.00,  0.50, 0.25, 0.10,0.5,0.1
Entry any number for profit
Entry two negative number like -1 -1 for exiting system

entry <payment profit> For example: 100 50
100 50
payment 100.00$
profit  50.00$
change  50.00$
return  1,50.00$

entry <payment profit> For example: 100 50
100 1
payment 100.00$
profit  1.00$
change  99.00$
return  1,50.00$  2,20.00$  1,5.00$  4,1.00$


entry <payment profit> For example: 100 50
50 10
payment 50.00$
profit  10.00$
change  40.00$
return  2,20.00$


entry <payment profit> For example: 100 50
```