

2024秋招大一题目

背景与假定

华科中散布着很多自动售货机，同学们经常在售货机上买东西吃，同时，同学们对自动售货机是如何工作的感到好奇，我们希望你根据此背景，联系实际，模拟出自动售货机的工作流程。

题目中会给出我们希望你实现的功能，你需要将功能实现并且按售货机的工作流程串联起来。对于其中的一些题目，并没有绝对的标准答案，只要有合理的依据即可。我们希望看到你学习C语言与独立通过查找资料等方式编写程序的过程。

注：1、所有的货物均用一个大写英文字母代替（如A,B,C...X,Y,Z）

2、假定这个售货机只有5个通道用来摆放货物，每个通道只能放一种货物，且每个通道最多放入50个货物；不同的通道可以放相同种类的货物

3、假定每种货物的单价均在10元以下，且均为整数

要求

1、只允许使用C语言

2、需要有适当的用户输入输出提示词，但不要求使用图形化界面

3、对于已你提交的工程，我们希望通过gcc等工具一键编译、运行，你也可以使用VS等IDE一键编译并提交sln文件，并且拥有可执行文件（.exe）。

4、如果你没有做全部题目也没有关系，相比于完成度，我们更看重你的学习态度与能力：你需要给出一份学习文档，记录学习过程中你认为的重点和疑点并整理简要文档，但请确保文档上写的知识你完全理解，该文档会作为答辩提问的来源之一。

题目

Level0-1

学会使用Git，并提交自己的远程仓库链接（完成后请填写招新群内的收集表，提交链接）

参考资料：Git 教程 | 菜鸟教程 (runoob.com) 不会科学上网的建议使用gitee代替github。（不要花费太长时间停留在这里，可以在完成下列任务的时候加深理解，来修改补充文档）

Level0-2

1. 请自行学习关于C语言知识包括：基本数据类型，程序控制语句，函数，数组，文件读写

2. 学习C程序从源代码到可执行文件的整个过程及对应的编译器命令：

- 我们知道，C语言属于高级编程语言，我们需要将C转化为计算机可以看懂的机器语言，再转化为可执行文件。你需要学习（以gcc为例）整个编译过程，掌握每一步的指令，自己动手实验将.c代码转化为可执行文件，并在文档中记录你的实验流程与收获。
- 选做：感兴趣的同学还可以学习一下编译系统CMake

Level1-1

售货机上必须有货物才能进行售卖，所以首先需要模拟在售货机上摆放货物的功能。用户通过手动输入货物种类、货物所在的通道标号、货物单价以及货物个数，实现在售货机某一通道摆放货物。

对于输出，需要展示摆放的结果

例如：输入

```
1 A 3 7 5 //表示 货物A摆放在3号通道，单价为7元，摆放了5个
```

输出

```
1 3: AAAAA 7 //3为3号通道，放了5个货物A，所以输出5个A，7表示单价7元
```

Level1-2

售货机上有了货物之后用户就可以进行购买操作。请编写程序模拟此购买功能（购买功能分为投币、找零两部分），投币功能只能输入1元、2元和5元，当投币的金额小于想要购买货物的总价（总价=单价*数量）时，需要一直投币，直到投币金额大于等于想要购买货物的总价时，结束投币，开始找零。

用户手动输入购买货物需要的总价，以及每次投币的金额，最终输出找零的金额

（选做内容）：将输入由手动输入改为从文件输入。

例如：输入

```
1 7 // 总价为7元
2 1 // 第一次投入1元
3 2 // 第二次投入2元
4 5 // 第三次投入5元，此时投币金额足够，执行找零
```

输出

```
1 1 // 找零1元
```

Level1-3

在完成Level1-1和Level1-2之后，你已经可以搭建一个简易的自动售货机的工作流程，所以请将前两个Level的功能串联起来实现一个简单的售货机，要求除了正常实现前两个Level的功能，还需要允许用户在货物卖空之前重复购买。（可以学习 状态机 相关内容）

此时，由于工作流程开始复杂，用户的输入输出也发生了变化：摆放货物功能的输入格式不变并且只需要输入一次即可，购买功能的第一次输入由输入总价，变为输入需要购买的货物、货物所在的通道以及购买的数量。输出只需要输出找零金额。

例如：输入

```
1 A 3 7 5 // 货物A摆放在3号通道，单价为7元，摆放了5个
2 A 3 1 // 购买3号通道的A货物，买1个
3 1 // 第一次投入1元
4 2 // 第二次投入2元
5 5 // 第三次投入5元，此时投币金额足够，执行找零
6 ...
```

输出

```
1 1 // 找零1元，之后开始下一次购买
2 ...
```

注：1、在该Level只考虑一种货物一个通道的情况

2、考虑到实际情况，用户肯定会出现误操作的情况，会输入一些不合理的内容，我们希望你对这些不合理的内容进行处理。

3、可以输出一些提示信息，让你的工作流程更加一目了然

Level2-1

面对同学们日益增长的购物需求，售货机增加了货物的种类，并且将通道数增加到了5个（编号1~5），你需要升级你的售货机以适应新功能：摆放货物功能需要支持不同种类的货物分别摆放在不同的通道中，并且不同的通道可以放相同种类的货物，先摆放完所有货物，再选择要购买的货物，当选择完所有需要购买的货物时，再进行投币。（可以学习 状态机 相关内容）

输入：

```
1 A 3 7 5 // 货物A摆放在3号通道，单价为7元，摆放了5个
2 B 2 3 5 // 摆放B
```

```
3 B 1 3 3 // 摆放B
4 ...
5 END // 摆放结束
6 A 3 1 // 购买3号通道的A货物，买1个
7 B 2 1 // 选择B
8 ...
9 END // 选择结束
10 1 // 第一次投入1元
11 5 // 第二次投入5元
12 5 // 第三次投入5元，此时投币金额足够，执行找零
13 ...
```

输出：

```
1 1 // 找零1元，之后开始下一次购买
2 ...
```

注：你需要自定义某种操作，来判断是否摆放完所有货物，以及是否选择完所有需要购买的货物。

Level2-2

为了让售货机的操作更人性化，学校对所有售货机进行了升级，新增了回退功能。每当用户对自己的操作感到后悔的时候就可以选择回退功能，撤销自己之前的操作，回退操作可以重复使用，但最多撤销三步操作。

用户输入输出格式同Level2-1，同理，你需要自定义回退功能的操作标志（比如输入"back"就执行回退操作），并且应当判断是否可以回退操作。

参考文献

vscode的下载和配置<https://blog.csdn.net/toby54king/article/details/105453603>

状态机参考<https://blog.csdn.net/JMW1407/article/details/108458569>