1. 本程序使用了Qt Creator 2.4.1（基于 Qt 4.7.4）进行编写，如果想要自己把工程编译一遍，请下载相应程序，然后打开project文件夹中的 Restaurant.pro进行编译即可。
2. 如果只是正常运行，不需安装Qt，打开release文件夹中的exe文件即可。
3. 如果测试运行中出现故障，很有可能是参数设置有问题，比如**应该输入整数时输入了小数，单位理解错误**等。
4. 如果觉得测试的结果数据不太符合现实或者出现了莫名其妙的数据，请先检查参数设置是否合理，可能情况同上。
5. 阅读源代码，请打开src文件夹，里面注释较为详细，可自行阅读。本程序的结构如下：
   1. Consumer类：代表顾客，执行顾客相关行为操作（consumer.h、consumer.cpp）
   2. Restaurant类：代表餐厅，执行餐厅相关操作，如制作食品、丢弃食品等。（restaurant.cpp、restaurant.h）
   3. Simulation类：模拟类，用来保存相关模拟数据，和模拟餐厅与顾客的交互过程，一次模拟过程的实现（以分钟为单位）在simulate()函数中。（simulation.cpp、simulation.h）
   4. MainWindow.cpp：主窗口程序，主要看on\_startButton\_clicked()函数即可，在该函数中调用了simulation类进行循环模拟。
   5. 其余代码均是Qt自动生成的GUI构件，不用管。
6. 关于本次的题目，在你们的算法的基础上根据实际情况进行了简化和修改，具体请看下面你们的算法中红字标出的部分。

（ 基本分数12分）快餐店运行模拟

某快餐店供应若干种快餐和饮料（5种以上），早晨6：00开始营业，晚上11：00打烊。前一天已经安排了若干工人上班，快餐店的用餐位是固定的，每种食物的成本和销售价格是确定的，每种食物的总量是确定的，储存事先准备好的各种食物的空间是确定的，顾客人数和顾客到达的时间是随机的，每人点餐的时间为一分钟，顾客用餐的时间是随机的。营业时间里，+其中7：00—9：00、11：00—13：00、17：00—19：00是用餐高峰期。

食物可以事先准备好，但是每种食物有不同的保鲜期，超过保鲜期的食物将被丢弃。顾客到达后排队点餐，可能因为想要的某种食物没有准备好而等待，也可能因为准备好的食物时间较长（未过保鲜期）而等待现做的；**（这个功能我省略了，因为不是很重要实现起来却非常耗时间，而且对输出结果并没有什么影响。）**可能因为食物售完或要等待而换成其他食物，抑或离开。如果店里没有空位，顾客可选择等待或离开。

编写一个模拟该快餐店一天运行情况的应用程序，要求：

* 设计系统需要的输入数据，并通过文本文件输入这些数据；
* 生成一天模拟运行（数据信息）的报告，该报告存储在数据文件中，具体信息包括：

1. 每种食物的销售数量、丢弃数量和利润（扣除成本后的收入）以及快餐店一天的总利润；
2. 顾客因为食物售完或其他原因离开的人数，因为没有用餐位离开的人数；
3. 顾客等待用餐位置的时间；
4. 顾客等待食物的时间（每种食物一个数据）；**（等待食物应该是从点餐之后算起，因此排队时间并不计算在内，这样理解应该没错）**
5. 每种食物不能满足顾客点餐要求的次数；
6. 高峰期和非高峰期餐位的利用率。

* 程序测试时应该应用多组参数，并产生不同的结果，从中判断系统结果是否符合实际，如增加用餐位，顾客等待用餐位置的时间会减少，非高峰期餐位的利用率会降低等

总体思想：  
按分钟为单位通过循环模拟一天的运营情况；

固定的数据设定一个默认值，由外部文本文件导入，可以在运行前改变参数（如没有位子时等待和离开的概率、某种食物的价格、成本），通过改变不同参数值与决定概率的函数参量进行测试得出合理的参数。

某事件是否发生、其结果由随机函数产生（概率事先已设定）

即时在界面上显示运营情况，每小时（60分钟）后暂停运行，询问是否要显示即时的运营情况（销售额、利润等）,最后将所有运营情况输出至文本文件生成运营日志。

**（6：00~23：00共有一千多分钟，每分钟都输出运行情况的话将非常耗时，因此最终实现时仅将最后结果输出，而且题目要求也是如此。）**

一天的模拟后将总营业情况报告输出到文本文件中（在题目中已给出需要输出的内容）

如店内有S（默认15）个餐位，高峰点每分钟每种食物各做出来一份，非高峰点每种食物每10分钟各做出来一份，主食（汉堡、薯条、蛋挞、鸡块）10分钟未售出要扔掉，饮料（可乐、橙汁、咖啡）不会变质扔掉；某种食物准备的数量累积至10份后便不再做

**（这些参数都可以在程序中自行修改，下同）**

每分钟内可能发生的事件及顺序（括号内为在屏幕上随着时间显示的改变的量）：

1)新的食物做好 （食物剩余情况：……）

高峰时每分钟各种食物各做出来一份，非高峰每十分钟各做出来一份

2)等待食物的顾客拿到已点食物，店内就餐\离开就餐 （剩余食物情况：……）

选择店内就餐\离开就餐概率分别为b1/b2(默认0.7/0.3)

3)顾客排队等待时选择离开 （排队点餐人数：……）

**选择离开概率为此时等待人数乘以c（默认0.05），即排队人数不会超过1/c（20）人**

4)顾客用餐结束离开 （空位：）

用餐时间调用函数A

**（我并没有严格按照你们的算法写函数，但实现效果与你们的想法一致，具体实现请阅读源代码，其他函数同理）**

5）等待空位的顾客就坐 （空位：）

6)顾客到达，排队等候/开始点餐/无空位选择离开 （排队等候点餐人数：……）

**高峰\非高峰时每分钟顾客到达人数调用函数B\C，顾客有人排队时是否选择等待同（3），无空位时选择离开的概率为50%（有人排队时先判断是否因为需要排队离开，若不离开再判断是否因为没有空位离开）。**

7）顾客点了……，拿到食物，店内就餐\离开就餐；

**选择店内就餐\离开就餐概率分别为b1/b2(默认0.7/0.3)**

**默认每份订单点主食份数1~5的概率各为0.2，再根据函数D决定这X份主食点汉堡\薯条\蛋挞\鸡块份数，每份订单点饮料份数的概率各为0.25，这X份饮料点可乐\橙汁\咖啡份数的函数调用E，（每份点单至少包括一份主食、一份饮料）**

8）或XX剩余不足，顾客选择等待/放弃点餐/将X份XX换成X份XX （空位：……，排队等待人数：……，食物剩余情况：……）

**某种食物不足顾客选择等待概率d1，放弃点餐概率d2，换餐概率d3(默认0.4/0.2/0.4)，换餐随机将不足的X份主食换成另外X份有剩余主食（不一定是一种），饮料同理；如果剩余主食或饮料份数不够选择等待\放弃点餐概率分别为50%**

**（这里我简化了，只要顾客点的所有食品中有一种不足，顾客就会考虑换餐）**

9）顾客开始点餐 （排队等候人数:……）

10）食物保保鲜期已到，扔掉…… （剩余食物情况：……） （只可能发生在非高峰期）

算法：

用一个结构/类储存一个顾客的数据，产生新顾客则new出一个结构数据，结构中包括：顾客编号、到达时间、以及上述的选择结果、等待时间等数据，将对最后报告可能产生影响的数据储存。顾客离开后删除这个结构

每分钟开始决定是否产生\产生几个新顾客，并执行所有存在顾客的事件。

店内情况（空位、排队人数、食物剩余情况等）用一个数组或结构储存，也可以与其他报告中数据一起用一个类表示

例如输出结果：  
6:00开业 （空位15/15，无人排队等待点餐，食物剩余情况：汉堡0，薯条0，蛋挞0，鸡块0，可乐0，橙汁0，咖啡0）

。。。

6:14 顾客1到达，开始点餐 （排队点餐人数：1）

6:15 顾客1点了1份汉堡，1份可乐，店内就餐 （空位14/15，无人排队等待点餐，食物剩余情况：汉堡0，可乐0）

顾客2到达，开始点餐（排队点餐人数：1）

。。。

7:13 新的食物做好 （食物剩余情况：汉堡1，薯条5，蛋挞2，鸡块1，可乐1，橙汁5，咖啡3）

顾客20拿到点的一份鸡块，离开就餐 （食物剩余情况：鸡块：0）

顾客17用餐结束离开 （空位1/15）

顾客24到达，排队等待 （排队等待点餐人数：4）

顾客21点了一份薯条，1份橙汁，店内就餐 （空位0/15，排队等待点餐人数：3，食物剩余情况：……）

顾客22开始点餐 （排队等待点餐人数：3）

。。。

7:16 新的食物做好（食物剩余情况：……）

顾客29排队等待时选择离开 (排队等待点餐人数：4）

顾客25点了1份蛋挞，2份咖啡； 咖啡剩余不足，顾客25选择等待（排队点餐人数：3，食物剩余情况：蛋挞2，咖啡0）

。。。

7：17 新的食物做好（食物剩余情况：……）

顾客27点了1份汉堡，一份咖啡；咖啡剩余不足，顾客26放弃点餐，选择离开 （排队点餐人数：5）

顾客28开始点餐 （排队点餐人数：5）

。。。