**实验一 队列仿真**

1. 实验题目：用队列结构仿真若干窗口前的排队情况，并求平均等待时间。
2. 解题思路：

1）首先创建k个队列，设置秒，学生数，总服务时间计数器。

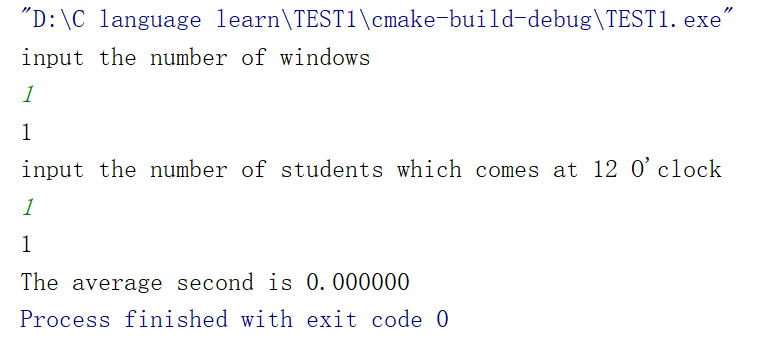
2）每秒循环一次，每到60的整数倍，按照高斯函数求出的人数入队，对每个学生入最短队，元素的值赋为随机产生的服务时间，每个学生的服务时间为队列里面前面所有元素之和，更新总服务时间。每秒更新队列状态(首元素减一，直至减到零删除)，同时更新学生数。

3) 将总时间除以学生数，得到平均等待服务时间。

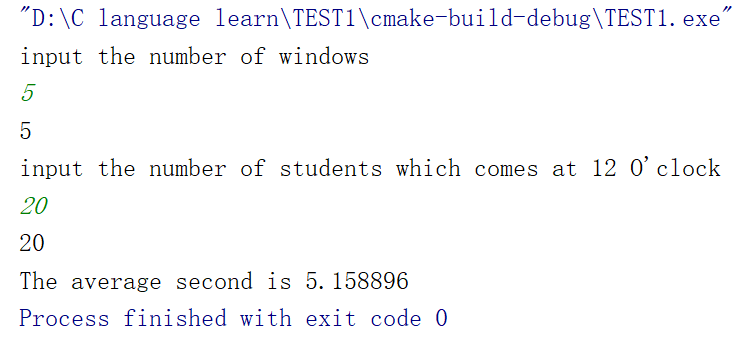
1. 代码
2. #include **<iostream>**#include **<stdio.h>**#include **<math.h>**#include **<time.h>**#include **<stdlib.h>**#define **Maxsize** 1000  
   #define **pi** 3.1415926  
   **typedef struct** {  
    **int** front,rear;  
    **int** q[**Maxsize**];  
   }Queue;  
   **void** InitQ(Queue &q);  
   **bool** EnQ(Queue &q,**int** x);  
   **bool** DeQ(Queue &q,**int** &e);  
   **int** QueueLength(Queue q);  
   **int** random(**int** i,**int** j);  
   **int** Guass(**int** x,**int** N);  
   **int** shortest(Queue window[],**int** k);  
   **float** getAveTime(**int** N,**int** k);  
   **int** main()  
   {  
    **int** k,N;  
    printf(**"input the number of windows\n"**);  
    scanf(**"%d"**,&k);  
    printf(**"input the number of students which comes at 12 O'clock\n"**);  
    scanf(**"%d"**,&N);  
    printf(**"The average second is %f"**,getAveTime(N,k));  
   }

**void** InitQ(Queue &q)  
{  
 q.rear=q.front=0;  
}  
**bool** EnQ(Queue &q,**int** x)  
{  
 **if**((q.rear+1)%**Maxsize**==q.front)  
 **return false**;  
 q.q[q.rear]=x;  
 q.rear=(q.rear+1)%**Maxsize**;  
 **return true**;  
}  
**bool** DeQ(Queue &q,**int** &e)  
{  
 **if**(q.front==q.rear)  
 **return false**;  
 e=q.q[q.front];  
 q.front=(q.front+1)%**Maxsize**;  
 **return true**;  
}  
**int** QueueLength(Queue q)  
{  
 **return**(q.rear-q.front);  
}  
**int** random(**int** i,**int** j)  
{  
 **return** rand()%(j+1-i)+i;  
}  
**int** Guass(**int** x,**int** N)  
{  
 **double** y=x/120.0-0.5;  
 **return int**(N \* exp(-0.5\*sqrt(2\***pi**) \* y \* y));  
}  
**int** shortest(Queue window[],**int** k)  
{  
 **int** i,j=0;  
 **int** temp=QueueLength(window[0]);  
 **for**(i=1;i<k;i++)  
 {  
 **if**(QueueLength(window[i])<temp)  
 {  
 temp=QueueLength(window[i]);  
 j=i;  
 }  
 }  
 **return** j;  
}  
  
**float** getAveTime(**int** N,**int** k)  
{  
 Queue window[k];  
 **int** i,j,e;  
 **for**(i=0;i<k;i++)  
 {  
 InitQ(window[i]);  
 }  
 **int** sec=0;  
 **int** serve=0;  
 **int** stu=0;  
 **for**(sec=0;sec<7200;sec++)  
 {  
 **if**(sec%60==0)  
 {  
 **for**(i=0;i<Guass(sec/60,N);i++)*//对Guass个学生入队* {  
**你 for**(j=window[shortest(window,k)].front;j<window[shortest(window,k)].rear;j++)  
 serve=serve+window[shortest(window,k)].q[j];*//计算等待时间* srand((**unsigned int**)(time(**NULL**)));*//生成随机种子* **if**(EnQ(window[shortest(window,k)],random(4,6))== **false**)  
 printf(**"There is a student who cannot be in a queue\n"**);  
 stu++;*//计算学生数* }  
 }  
 **for**(i=0;i<k;i++)*//更新队列状态* {  
 你 **if**((QueueLength(window[i])!=0)&&(window[i].q[window[i].front]!=0))  
 window[i].q[window[i].front]--;  
 **if**(window[i].q[window[i].front]==0)  
 DeQ(window[i],e);  
 }  
 }  
 **return float**(serve)/**float**(stu);  
}

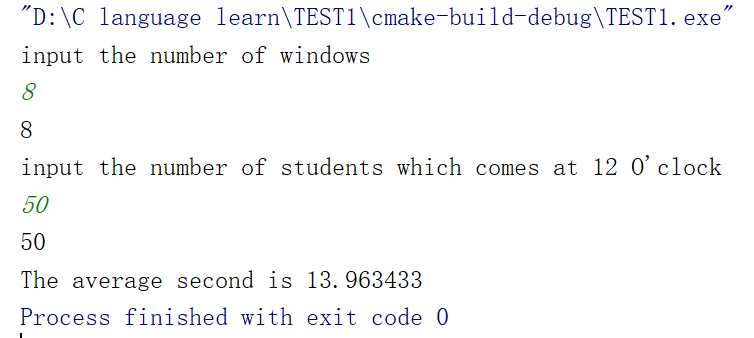
1. 运行结果

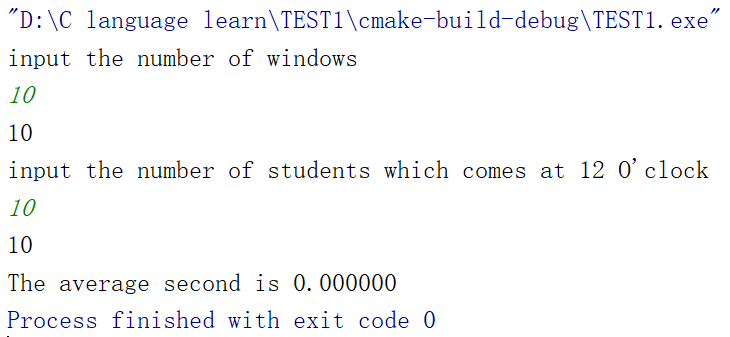


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 心得体会
2. 产生a到b的随机数：

首先用srand函数产生随机种子

产生随机数rand( ) %(b+1-a)+a  
 2）队列结构可以用来仿真先进先出的情况，对现实生活中的一些事件做出 科学的预测。