银行队列ps：老师对不起，写这个代码耗尽了我的脑力，我已然一具行尸走肉，注释真的写不了了/吐血

#include <iostream>

#include <stdlib.h>

#include<time.h>

#include<windows.h>

using namespace std;

int now\_time=0;

int total\_member=0;

int wait\_time=0;

int gap\_time=0;

int process=0;

int last\_leave\_time=0;

int open\_time=0;//开门时间

int off\_time=0; //关门时间

int total\_time=0;

int cus\_ident=0;

class Customer

{

private:

int arrive\_time;

int process\_time;

int leave\_time;

public:

Customer();

Customer(int process,int last\_leave\_time):arrive\_time(now\_time),process\_time(process)

{

leave\_time=last\_leave\_time+process;

}

int get\_leave\_time()

{

return leave\_time;

}

int get\_wait\_time()

{

return leave\_time-arrive\_time;

}

int get\_arrive\_time()

{

return arrive\_time;

}

void modify\_leave(int VIPtime)

{

leave\_time+=VIPtime;

}

};

Customer::Customer()

{

arrive\_time=0;

process=0;

leave\_time=0;

}

typedef Customer Item;

class Queue

{

private:

typedef struct Node

{

Item item;

struct Node\*next;

} Node;

Node \*front;

Node \*rear;

int number;//当前队列的客户人数；

public:

Queue();

~Queue();

void enqueue(const Item &item);

void VIPenqueue(const Item & item,int process);

bool dequeue(int i);

void dispose(int i);

bool isempty()

{

return front==NULL;

}

Node\* get\_rear()

{

return rear;

}

Node\* get\_front()

{

return front;

}

int get\_num()

{

return number;

}

};

Queue::Queue()

{

front=rear=NULL;

number=0;

}

Queue::~Queue()

{

Node\*temp;

while(front!=NULL)

{

temp=front;

front=front->next;

delete temp;

}

}

void Queue::enqueue(const Item & item)

{

Node\*add=new Node;

add->item=item;

add->next=NULL;

if(front==NULL)

front=add;

else

rear->next=add;

rear=add;

number++;

}

void Queue::VIPenqueue(const Item&item,int process)

{

Node\*add=new Node;

add->item=item;

if(front==NULL)

{

front=rear=add;

add->next=NULL;

}

else

{

add->next=front->next;

front->next=add;

add=add->next;

while(add)

{

add->item.modify\_leave(process);

add=add->next;

}

}

number++;

}

void Queue::dispose(int i)

{

Node\*temp=front;

front=front->next;

total\_member++;

wait\_time+=temp->item.get\_wait\_time();

//打印

int arr\_h,arr\_m,leave\_h,leave\_m;

arr\_h=open\_time+temp->item.get\_arrive\_time()/60;

arr\_m=temp->item.get\_arrive\_time()%60;

leave\_h=open\_time+temp->item.get\_leave\_time()/60;

leave\_m=temp->item.get\_leave\_time()%60;

cout<<"办理窗口："<<i+1<<"号窗口 "<<"到达："

<<arr\_h<<":"<<arr\_m<<" 离开："<<leave\_h<<":"<<leave\_m<<" 逗留："<<temp->item.get\_wait\_time()<<"分钟"<<endl;

delete temp;

}

bool Queue::dequeue(int i)

{

if(front==NULL||front->item.get\_leave\_time()> now\_time)

return false;

Node\*temp=front;

front=front->next;

total\_member++;

wait\_time+=temp->item.get\_wait\_time();

//打印

int arr\_h,arr\_m,leave\_h,leave\_m;

arr\_h=open\_time+temp->item.get\_arrive\_time()/60;

arr\_m=temp->item.get\_arrive\_time()%60;

leave\_h=open\_time+temp->item.get\_leave\_time()/60;

leave\_m=temp->item.get\_leave\_time()%60;

cout<<"办理窗口："<<i+1<<"号窗口 "<<"到达："

<<arr\_h<<":"<<arr\_m<<" 离开："<<leave\_h<<":"<<leave\_m<<" 逗留："<<temp->item.get\_wait\_time()<<"分钟"<<endl;

number--;

delete temp;

}

int get\_short\_queue(Queue\*queue,int n)

{

int short\_num=queue[0].get\_num(),short\_mark=0,i;

for(i=1; i<n; i++)

if(queue[i].get\_num()<short\_num)

{

short\_num=queue[i].get\_num();

short\_mark=i;

}

return short\_mark;

}

void rand\_time()

{

srand((unsigned)time(NULL));

process= rand() % 30 + 1;

gap\_time=rand() % 5 + 1;

cus\_ident=rand() %10 ;

now\_time+=gap\_time;

Sleep(1000);

}

int main()

{

int n,i,j,short\_queue=0;

cout<<"请输入办理窗口的数目："<<endl;

cin>>n;

Queue queue[n];

cout<<"输入银行的24小时制营业时间:如营业时间为9:00--17:00，则应输入:9 17\n";

cin>>open\_time>>off\_time;

total\_time = (off\_time - open\_time) \* 60;//计算银行总营业多少分钟

cout<<"在银行关门前办理完的客户\n";

rand\_time();

while(now\_time<total\_time)

{

for(i=1; i<=5; i++)

for(j=0; j<n; j++)

queue[j].dequeue(j);

short\_queue=get\_short\_queue(queue,n);

last\_leave\_time=now\_time;

//普通客户

if(cus\_ident)

{

if(queue[short\_queue].get\_rear()!=NULL)

last\_leave\_time=queue[short\_queue].get\_rear()->item.get\_leave\_time();

Item cust(process,last\_leave\_time);

queue[short\_queue].enqueue(cust);

}

else

{

//VIP用户

if(queue[short\_queue].get\_front()!=NULL)

last\_leave\_time=queue[short\_queue].get\_front()->item.get\_leave\_time();

Item cust(process,last\_leave\_time);

queue[short\_queue].VIPenqueue(cust,process);

}

rand\_time();

}

cout<<"\n银行关门时间到不再接收客人\n\n";

for (int i = 0; i < n; i++)

{

queue[i].dispose(i);//输入在银行关门前还没有办理完业务的客户信息

}

cout<<"共接待客户人数："<<total\_member<<endl;

cout<<"客户平均逗留时长为："<<(float)wait\_time/total\_member<<"分钟";

return 0;

}



