

飞机场的模拟：

考虑一个小型繁忙的飞机场，该飞机场只有一条飞机跑道。在每个单位时间内，只有一架飞机可以着陆或者只有一架飞机可以起飞，不允许同时着陆和起飞。飞机的到达和起飞时是随机的，因此在任何时刻，该机场跑道可能是空闲的，或者有一架飞机正在着陆或起飞，也可能有若干架飞机在等待着陆或等待起飞。

我们需要创建的类有：

飞机类：

这个类描述的对象为单个飞机。

首先枚举飞机的状态通过以下语句：

```
enum Plane_status{null, arriving, departing};
```

然后描述两个类分别如下，

```
class Plane{
public:
    Plane();
    Plane(int flt, int time, Plane_status status);
    void refuse() const;
    void land(int time) const;
    void fly(int time) const;
    int started() const;
private:
    int flt_num;
    int clock_start;
    Plane_status state;
};
```

跑道类：

```
enum Runway_activity {idle, land, takeoff}
```

```
class Runway{
public:
    Runway(int limit);
    Error_code can_land(const Plane& current);
    Error_code can_land(const Plane& current);
    Runway_activity activity(int time, Plane &moving);
    void shut_down(int time) const;
private:
    Extended_queue landing;
```

```

Extended_queue takeoff;
int queue_limit;//队列的最大长度
int num_land_requests; //要求降落的飞机数目
int num_takeoff_requests;//要求起飞的飞机数目
int num_landing;//已降落的飞机数目
int num_takeoffs;//已起飞的飞机数目
int num_land_accepted;//在降落队列里的飞机数目
int num_takeoff_accepted;//在起飞队列里的飞机数目
int num_land_refused;//被拒绝的要降落飞机数目
int num_takeoff_refused;//被拒绝的要起飞飞机数目
int land_wait; //飞机等待降落的总时间
int takeoff_wait;//飞机等待起飞的总时间
int idle_time;//机场处于空闲状态的总时间
};

```

其中

关闭函数定义如下

```
void Runway::shut_down(int time) const
```

```
//跑道数据在这里总结计算并打印出来
```

```

{
    cout<< ``Simulation has concluded after"<<time<<"time units"<<endl
        <<`` Total number of planes processed"
        <<(num_land_requests+num_takeoff_requests)<<endl
        <<``Total number of planes asking to land"
        << num_land_requests<<endl
        <<``Total number of planes asking to taking off"
        << num_takeoff_requests<<endl
        <<``Total number of planes accepted for landing"
        << num_land_accepted<<endl
        <<``Total number of planes accepted for takeoff"
        << num_takeoff_accepted<<endl
        <<``Total number of planes refused for landing"
        << num_takeoff_accepted<<endl
        <<``Total number of planes refused for takeoff"
        << num_takeoff_accepted<<endl
        <<``Total number of planes that landed"
        << num_takeoff_accepted<<endl
        <<``Total number of planes that took off"
        << num_takeoff_accepted<<endl
        <<``Total number of planes left landing queue'
        << landing.size<<endl
        <<``Total number of planes left in takeoff queue"
        << takeoff.size<<endl;
    cout << "Percentage of time runway idle "
        << 100.0 * ((float) idle_time)/((float) time) << "%" << endl;
    cout << "Average wait in landing queue "
        << ((float) land_wait)/((float) num_landings) << " time units";
}

```

```

cout << endl << "Average wait in takeoff queue "
<< ((float) takeoff_wait)/((float) num_takeoffs)
<< " time units" << endl;
cout << "Average observed rate of planes wanting to land "
<< ((float) num_land_requests)/((float) time)
<< " per time unit" << endl;
cout << "Average observed rate of planes wanting to take off "
<< ((float) num_takeoff_requests)/((float) time)
<< " per time unit" << endl;
}

```

主函数

```

void main()
{
    int end_time;
    int queue_limit;
    int flight_number = 0;
    double arrival_rate, departure_rate;
    initialize(end_time, queue_limit, arrival_rate, departure_rate);
    Random variable;
    Runway small_airport(queue_limit);
    for (int current_time = 0; current_time < end_time; current_time++){
        int number_arrivals = variable.poisson(arrival_rate);
        for (int i = 0; i < number_arrivals; i++){
            Plane current_plane(flight_number++, current_time, arriving);
            if (small_airport.can_land(current_plane) != success)
                current_plane.refuse();
        }
        int number_departures = variable.poisson(departure_rate);
        for (int j = 0; j < number_departures; j++){
            Plane current_plane(flight_number++, current_time, departing);
            if (small_airport.can_depart(current_plane) != success)
                current_plane.refuse();
        }
        Plane moving_plane;
        switch (small_airport.activity(current_time, moving_plane)){
            case land:
                moving_plane.land(current_time);
                break;
            case land:
                moving_plane.fly(current_time);
                break;
            case idle;

```

```

        run_idle(current_time);
    }
}
small_airport.shut_down(end_time);
}

```

其中初始函数如下：

```

void initialize(int &end_time, int &queue_limit, double &arrival_rate,
double&departure_rate)
//用户在此声明仿真所需的time units（时间单位）的数量，队列的最大长度和机场的平
均到达率和离开率。
{
    cout << "This program simulates an airport with only one runway." << endl
    << "One plane can land or depart in each unit of time." << endl;
    cout << "Up to what number of planes can be waiting to land "
    << "or take off at any time? " << flush;
    cin >> queue_limit;
    cout << "How many units of time will the simulation run?" << flush;
    cin >> end_time;
    bool acceptable;
    do {
        cout << "Expected number of arrivals per unit time?" << flush;
        cin >> arrival_rate;
        cout << "Expected number of departures per unit time?" << flush;
        cin >> departure_rate;
        if (arrival_rate < 0.0 || departure_rate < 0.0)
            cerr << "These rates must be nonnegative." << endl;
        else
            acceptable = true;
        if (acceptable && arrival_rate + departure_rate > 1.0)
            cerr << "Safety Warning: This airport will become saturated. " << endl;
    } while (!acceptable);
}

```

main 函数跑完后窗口出现的结果如下：

This program simulates an airport with only one runway.
One plane can land or depart in each unit of time.
Up to what number of planes can be waiting to land or take off at any time ? 5
How many units of time will the simulation run ? 1000
Expected number of arrivals per unit time ? .48
Expected number of departures per unit time ? .48
Plane number 0 ready to take off.
0: Plane 1 landed; In queue 0 units.
Plane number 0 took off after 0 time units in the takeoff queue.
Plane number 1 ready to take off.
1: Plane number 1 took off after 0 time units in the takeoff queue.
Plane number 2 ready to take off.
Plane number 3 ready to take off.
2: Plane number 2 took off after 0 time units in the takeoff queue.
Plane number 4 ready to land.
Plane number 5 ready to take off.

3: Plane number 4 landed after 0 time units in the takeoff queue.
Plane number 6 ready to land.
Plane number 7 ready to land.
Plane number 8 ready to take off.
Plane number 9 ready to take off.

4: Plane number 6 landed after 0 time units in the takeoff queue.
Plane number 10 ready to land.
Plane number 11 ready to take off.

5: Plane number 7 landed after 1 time unit in the takeoff queue.
Plane number 12 ready to land.

6: Plane number 10 landed after 1 time unit in the takeoff queue.

7: Plane number 12 landed after 1 time unit in the takeoff queue.
Plane number 13 ready to land.
Plane number 14 ready to take off.

takeoff queue is full Plane number 14 told to try to takeoff again later.

8: Plane number 13 landed after 0 time units in the takeoff queue.

9: Plane number 3 took off after 7 time units in the takeoff queue.

10: Plane number 5 took off after 7 time units in the takeoff queue.

11: Plane number 8 took off after 7 time units in the takeoff queue.
Plane number 15 ready to take off.

12: Plane number 9 took off after 8 time units in the takeoff queue.
Plane number 16 ready to land.
Plane number 17 ready to land.

13: Plane number 16 landed after 0 time units in the takeoff queue.
Plane number 18 ready to land.

14: Plane number 17 landed after 1 time unit in the takeoff queue.

15: Plane number 18 landed after 1 time unit in the takeoff queue.
Plane number 19 ready to land.
Plane number 20 ready to take off.

16: Plane number 19 landed after 0 time units in the takeoff queue.

17: Plane number 11 took off after 12 time units in the takeoff queue.

18: Plane number 15 took off after 6 time units in the takeoff queue.

19: Plane number 20 took off after 3 time units in the takeoff queue.

both queues are empty 20: Runway is idle.

Simulation has concluded after 1000 time units.

Total number of planes processed	970
Total number of planes asking to land	484
Total number of planes asking to take off	486
Total number of planes accepted for landing	484
Total number of planes accepted for takeoff	423
Total number of planes refused for landing	0
Total number of planes refused for takeoff	63
Total number of planes that landed	483
Total number of planes that took off	422
Total number of planes left in landing queue	1
Total number of planes left in takeoff queue	1
Percentage of time runway idle	9.5 %
Average wait in landing queue	0.36646 time units
Average wait in takeoff queue	4.63744 time units
Average observed rate of planes wanting to land	0.484 time units
Average observed rate of planes wanting to take off	0.486 time units