INTRODUCTION

Program structure design

- I organizes Main() variations inside a namespace “Programs”

```
10  namespace Programs
11  {
12      void main_1();
13      void main_2();
14      void main_3();
15      void main_4();
16      void main_5();
17  }
```

- Option Picker

```
char Utilities::options()
{
    char opt = null;
    while (opt == null || opt == '\n')
    {
        std::cout << "1.Run P1 version of Airport simulator\n2.Run P2 ve
        opt = std::getchar();
    }
    return opt;
}
```

- Main() function

```
int main()
{
    switch(Utilities::options())
    {
        case '1':
            Programs::main_1();
            break;
        case '2':
            Programs::main_2();
            break;
        case '3':
            Programs::main_3();
            break;
        case '4':
            Programs::main_4();
            break;
        case '5':
            Programs::main_5();
            break;
        default:
            std::cout << "Wrong input, exit program." << std::endl;
    }
}
```

METHOD

Using test cases for solving and testing algorithms for each programs.

Test cases

- Case 1: 2, 40, 0.7, 0.3

- Case 2: 3, 40, 0.7, 0.3

RESULTS

P1.

- On first program, I reuse program on source code to testing, header and source code files.

- On first testing, Runway will refuse new Plane if the random rate of landing and/ or takeoff are high. For example, on simulation with input values (Waiting queue: 2, units of time: 40, landing rate: 0.7, takeoff rate: 0.2), the runway had to ignore 1 plane that want to land:

```
- small_airport: 0: Runway is idle.
- small_airport: 1: Runway is idle.
Plane number 0 ready to land.
- small_airport: Fuels: 0, wait 0, 2: Plane number 0 landed after 0 time units in the landing queue.
Plane number 1 ready to land.
Plane number 2 ready to land.
Plane number 3 ready to land.
Plane number 3 directed to another airport
- small_airport: Fuels: 0, wait 0, 3: Plane number 1 landed after 0 time units in the landing queue.
- small_airport: Fuels: 0, wait 1, 4: Plane number 2 landed after 1 time unit in the landing queue.
Plane number 4 ready to land.
```

- On second testing, I ran 2 simulations with different waiting size: 2 and 3, but not change other input values.

Output

- On first result, when size of queue is only 2, the runway had to refuse 4 planes and the rate of idle is 25%

```
Simulation has concluded after 40 time units.
Total number of planes processed 32
Total number of planes asking to land 28
Total number of planes asking to take off 4
Total number of planes accepted for landing 26
Total number of planes accepted for takeoff 4
Total number of planes refused for landing 2
Total number of planes refused for takeoff 0
Total number of planes that landed 26
Total number of planes that took off 4
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 25%
Average wait in landing queue 0.423077 time units
Average wait in takeoff queue 1.75 time units
Average observed rate of planes wanting to land 0.7 per time unit
Average observed rate of planes wanting to take off 0.1 per time unit
```

- On second result, when size of queue increase to 3, the runway accepted all planes and the rate of idle is 20%

```
Simulation has concluded after 40 time units.
Total number of planes processed 32
Total number of planes asking to land 28
Total number of planes asking to take off 4
Total number of planes accepted for landing 28
Total number of planes accepted for takeoff 4
Total number of planes refused for landing 0
Total number of planes refused for takeoff 0
Total number of planes that landed 28
Total number of planes that took off 4
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 20%
Average wait in landing queue 0.678571 time units
Average wait in takeoff queue 2 time units
Average observed rate of planes wanting to land 0.7 per time unit
Average observed rate of planes wanting to take off 0.1 per time unit
```

P2.

Modify

- Add one runway and change them to landing_runway and takeoff_runway

```
Runway landing_runway(queue_limit);
Runway takeoff_runway(queue_limit);
```

- Activity checking for takeoff_runway.

```
    std::cout << "- takeoff_runway: ";
switch (takeoff_runway.activity(current_time, moving_plane)) { //
  //  Let at most one Plane onto the Runway at current_time.
case land:
    moving_plane.land(current_time);
    break;
case take_off:
    moving_plane.fly(current_time);
    break;
case idle:
    Utilities::run_idle(current_time);
}
```

- Shutdown for second runway.

```
std::cout << "- takeoff_runway -" << std::endl;
takeoff_runway.shut_down(end_time);
```

Test case: 1

Output

- By using only for landing, this runway only worked in 65% of running time. However, it is not mean that using 2 runways is more efficiency. I believe that it would work more efficiency, if landing and takeoff rates are nearly equal. So, amount of planes will be separated equally.

```
- landing_runway -
Simulation has concluded after 40 time units.
Total number of planes processed 28
Total number of planes asking to land 28
Total number of planes asking to take off 0
Total number of planes accepted for landing 26
Total number of planes accepted for takeoff 0
Total number of planes refused for landing 2
Total number of planes refused for takeoff 0
Total number of planes that landed 26
Total number of planes that took off 0
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 35%
Average wait in landing queue 0.423077 time units
Average wait in takeoff queue -nan time units
Average observed rate of planes wanting to land 0.7 per time unit
Average observed rate of planes wanting to take off 0 per time unit
```

- Unlike landing_runway, this runway only worked in 10% of running time.

```
- takeoff_runway -
Simulation has concluded after 40 time units.
Total number of planes processed 4
Total number of planes asking to land 0
Total number of planes asking to take off 4
Total number of planes accepted for landing 0
Total number of planes accepted for takeoff 4
Total number of planes refused for landing 0
Total number of planes refused for takeoff 0
Total number of planes that landed 0
Total number of planes that took off 4
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 90%
Average wait in landing queue -nan time units
Average wait in takeoff queue 0 time units
Average observed rate of planes wanting to land 0 per time unit
Average observed rate of planes wanting to take off 0.1 per time unit
```

P3

Modify

- Add 1 more IF condition before refuse the plane, so it will ask can_land() landing_runway and takeoff_runway before refuse the plane.

//landing_runway -> takeoff_runway -> refuse

```
Plane current_plane(flight_number++, current_time, arriving);
if (landing_runway.can_land(current_plane) != success)
if (takeoff_runway.can_land(current_plane) != success)
current_plane.refuse();
```

//takeoff_runway -> landing_runway -> refuse

```
Plane current_plane(flight_number++, current_time, departing);
if (takeoff_runway.can_depart(current_plane) != success)
if (landing_runway.can_depart(current_plane) != success)
current_plane.refuse();
```

Test case: 1

Output

- Landing runway still refuses to land, however, the refused planes were moved to takeoff runway.

```
- landing_runway:
Simulation has concluded after 40 time units.
Total number of planes processed 28
Total number of planes asking to land 28
Total number of planes asking to take off 0
Total number of planes accepted for landing 26
Total number of planes accepted for takeoff 0
Total number of planes refused for landing 2
Total number of planes refused for takeoff 0
Total number of planes that landed 26
Total number of planes that took off 0
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 35%
Average wait in landing queue 0.423077 time units
Average wait in takeoff queue -nan time units
Average observed rate of planes wanting to land 0.7 per time unit
Average observed rate of planes wanting to take off 0 per time unit
- takeoff_runway:
Simulation has concluded after 40 time units.
Total number of planes processed 6
Total number of planes asking to land 2
Total number of planes asking to take off 4
Total number of planes accepted for landing 2
Total number of planes accepted for takeoff 4
Total number of planes refused for landing 0
Total number of planes refused for takeoff 0
Total number of planes that landed 2
Total number of planes that took off 4
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 85%
Average wait in landing queue 0 time units
Average wait in takeoff queue 0 time units
Average observed rate of planes wanting to land 0.05 per time unit
Average observed rate of planes wanting to take off 0.1 per time unit
```

P4

Modify

- Liked P3, however on this time, I added a new runway named backup_runway.

So, there are 2 unchanged runways and a flexible one.

```
Runway landing_runway(queue_limit); // add landing runway
Runway takeoff_runway(queue_limit); // add takeoff runway
Runway backup_runway(queue_limit); // add takeoff runway

for (int current_time = 0; current_time < end_time; current_time++) { //  loop over time intervals
   int number_arrivals = variable.poisson(arrival_rate);  //  current arrival requests
   for (int i = 0; i < number_arrivals; i++) {
      Plane current_plane(flight_number++, current_time, arriving);
      if (landing_runway.can_land(current_plane) != success)
      if (backup_runway.can_land(current_plane) != success)
      current_plane.refuse();
   }

   int number_departures= variable.poisson(departure_rate); //  current departure requests
   for (int j = 0; j < number_departures; j++) {
      Plane current_plane(flight_number++, current_time, departing);
      if (takeoff_runway.can_depart(current_plane) != success)
      if (backup_runway.can_depart(current_plane) != success)
      current_plane.refuse();
   }
```

Test case: 1

Output

```
- landing_runway:
Simulation has concluded after 40 time units.
Total number of planes processed 28
Total number of planes asking to land 28
Total number of planes asking to take off 0
Total number of planes accepted for landing 26
Total number of planes accepted for takeoff 0
Total number of planes refused for landing 2
Total number of planes refused for takeoff 0
Total number of planes that landed 26
Total number of planes that took off 0
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 35%
Average wait in landing queue 0.423077 time units
Average wait in takeoff queue -nan time units
Average observed rate of planes wanting to land 0.7 per time unit
Average observed rate of planes wanting to take off 0 per time unit
```

```
- takeoff_runway:
Simulation has concluded after 40 time units.
Total number of planes processed 4
Total number of planes asking to land 0
Total number of planes asking to take off 4
Total number of planes accepted for landing 0
Total number of planes accepted for takeoff 4
Total number of planes refused for landing 0
Total number of planes refused for takeoff 0
Total number of planes that landed 0
Total number of planes that took off 4
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 90%
Average wait in landing queue -nan time units
Average wait in takeoff queue 0 time units
Average observed rate of planes wanting to land 0 per time unit
Average observed rate of planes wanting to take off 0.1 per time unit
```

```
- backup_runway:
Simulation has concluded after 40 time units.
Total number of planes processed 2
Total number of planes asking to land 2
Total number of planes asking to take off 0
Total number of planes accepted for landing 2
Total number of planes accepted for takeoff 0
Total number of planes refused for landing 0
Total number of planes refused for takeoff 0
Total number of planes that landed 2
Total number of planes that took off 0
Total number of planes left in landing queue 0
Total number of planes left in takeoff queue 0
Percentage of time runway idle 95%
Average wait in landing queue 0 time units
Average wait in takeoff queue -nan time units
Average observed rate of planes wanting to land 0.05 per time unit
Average observed rate of planes wanting to take off 0 per time unit
```

P5

Modify

- Create an overloading operator for Plane and 1 Constructors for adding fuel.

- Add fuel configuration for default constructor.

```
Plane() {
    flt_num = -1;
    clock_start = -1;
    fuel = -1;
    state = null;
}

Plane operator= (const Plane &copy){
    this -> fuel = copy.fuel;
    this -> flt_num = copy.flt_num;
    this -> clock_start = copy.clock_start;
    this -> state = copy.state;
    return *this;
}

Plane(int flt, int time, Plane_status status, int fuel)
/*
Post:  The Plane data members flt_num, clock_start, state
       and fuel are set to the values of the parameters flt,
       time and status, respectively.
*/

{
    this -> fuel = fuel;
    this -> flt_num = flt;
    this -> clock_start = time;
    this -> state = status;
    std::cout << "Plane number " << flt << " ready to ";
    if (status == arriving)
        std::cout << "land." << std::endl;
    else
        std::cout << "take off." << std::endl;
}
```

- Add a swap function for swapping order of Planes.

```
void Runway::swap_plane (Plane *a, Plane *b)
{
    Plane t = *a;
    *a = *b;
    *b = t;
}
```

- Add a random value of fuel for landing Planes.

```
  fuel = variable.random_integer(MIN_FUEL, MAX_FUEL);
```

- Add a fuel checker in Runway class. For explanation, I will use an array of Plane to temporary hold the Plane in queue. Then, a loop will find out an “Out of fuel” Plane by checking Plane’s fuel and waiting time. if there is only one unit of time left for it to land. The emergency landing will run, and take it out off order and move it to the front of array. At the end, the array will be appended back to main queue.

```
bool Runway::enough_fuel(int time)
{
   Plane moving[queue_limit];
   Plane temp;
   bool status = false;
   int order = 0;

   while (landing.size() != 0) {
      landing.serve_and_retrieve(moving[order]);
      if(!status
         && order != 0 // If Plane order = 0, false
         && moving[order].check_fuel() - (time - moving[order].started()) <= 1)
         { //Emergency landing if fuel - time < 1
         std::cout << "(#Emergency Landed# - fuel: "
                   << (int)moving[order].check_fuel()
                   << ", " << time - moving[order].started()
                   << ") ";
         swap_plane(&moving[0], &moving[order]);
         for (int i = order; i > 1; --i)
         {
            swap_plane(&moving[i], &moving[i-1]);
         }
         status = true;
      }
      order++;
   }

   if (order != 0)
   for (int i = 0; i < order; i++)
   {  std::cout << "(" << moving[i].get_fly_num() << ") ";
      landing.append(moving[i]);
   }

   return status;
}
```

Test case: 2

Output

Additional: I had added some code for printing flight numbers, it will have to track the result.

- Obviously, the order of this should be 13 - 14 - 15. However, the fuel in 14th plane is 1, so it cannot wait for 13th plane landing first. Emergency alert was called and 14th Plane landed.

```
Plane number 13 ready to land.
Plane number 14 ready to land.
Plane number 15 ready to land.
(#Emergency Landed# - fuel: 1, 0) (14) (13) (15) Fuels: 1, wait 0, 15: Plane number 14 landed after 0 time units in the l
anding queue.
(#Emergency Landed# - fuel: 2, 1) (15) (13) Fuels: 2, wait 1, 16: Plane number 15 landed after 1 time unit in the landing
 queue.
```