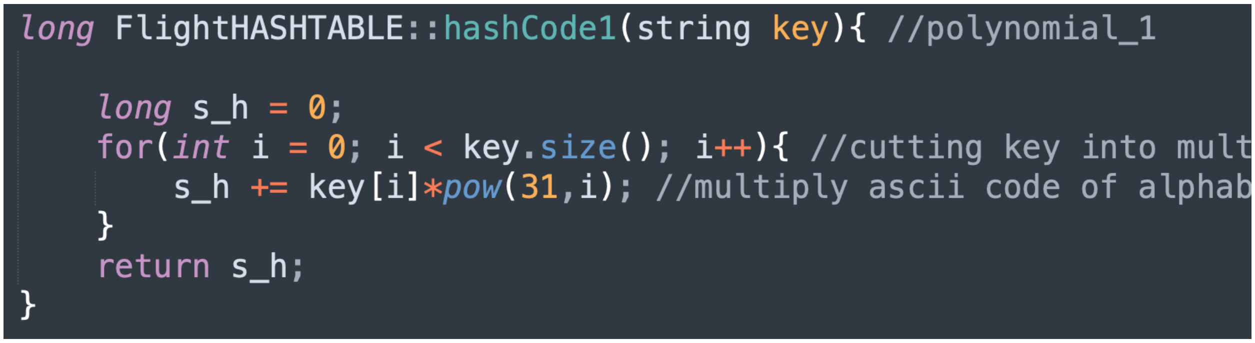
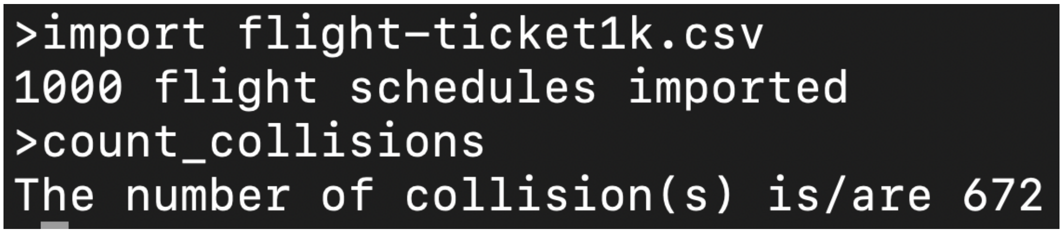
HashCode function #1(Polynomial)



```
long FlightHASHTABLE::hashCode1(string key){ //polynomial_1

    long s_h = 0;
    for(int i = 0; i < key.size(); i++){ //cutting key into mult
        s_h += key[i]*pow(31,i); //multiply ascii code of alphab
    }
    return s_h;
}
```

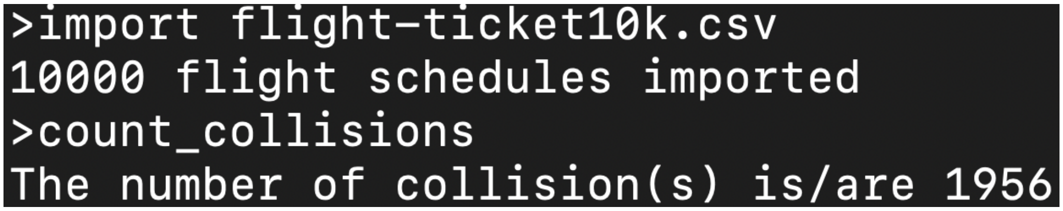
File #1(Flight-ticket1k.csv)



```
>import flight-ticket1k.csv
1000 flight schedules imported
>count_collisions
The number of collision(s) is/are 672
```

=> Avg. number of collisions per entry: 0.672(67.2%)

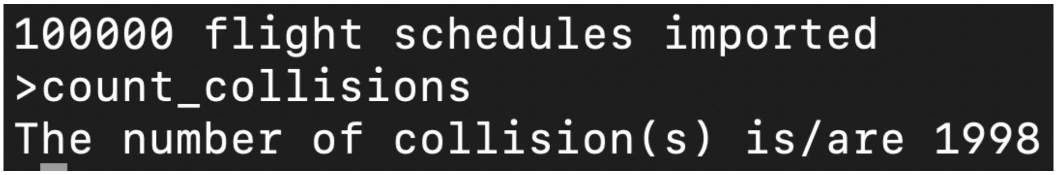
File #2(Flight-ticket10k.csv)



```
>import flight-ticket10k.csv
10000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1956
```

=> Avg. number of collisions per entry: 0.1956(19.56%)

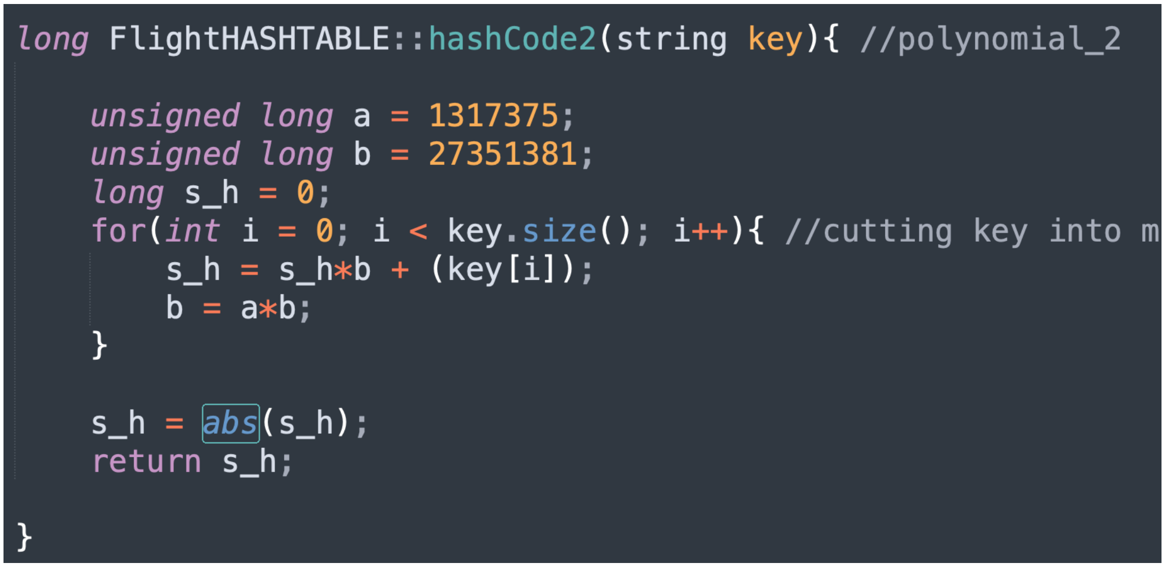
File #3(Flight-ticket100k.csv)



```
100000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1998
```

=> Avg. number of collisions per entry: 0.01998(1.998%)

HashCode function #2(polynomial,multiplication)



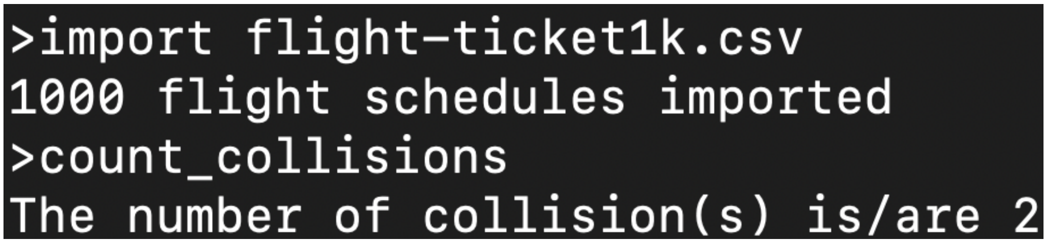
```
long FlightHASHTABLE::hashCode2(string key){ //polynomial_2

    unsigned long a = 1317375;
    unsigned long b = 27351381;
    long s_h = 0;
    for(int i = 0; i < key.size(); i++){ //cutting key into m
        s_h = s_h*b + (key[i]);
        b = a*b;
    }

    s_h = abs(s_h);
    return s_h;

}
```

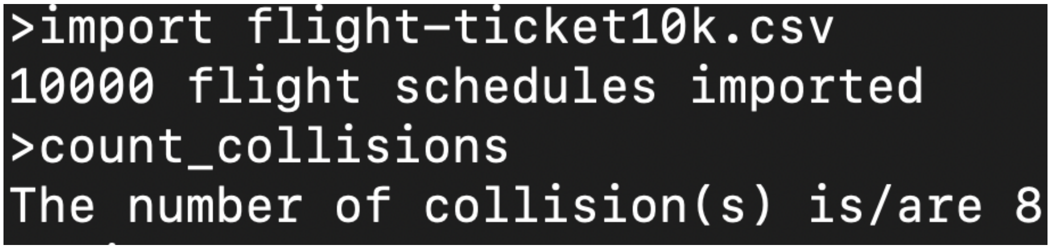
File #1(Flight-ticket1k.csv)



```
>import flight-ticket1k.csv
1000 flight schedules imported
>count_collisions
The number of collision(s) is/are 2
```

=> Avg. number of collisions per entry: 0.002(0.2%)

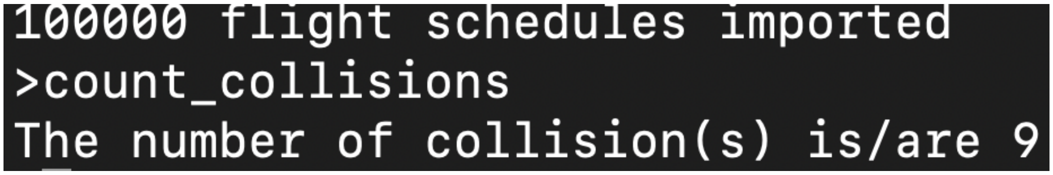
File #2(Flight-ticket10k.csv)



```
>import flight-ticket10k.csv
10000 flight schedules imported
>count_collisions
The number of collision(s) is/are 8
```

=> Avg. number of collisions per entry: 0.008(0.8%)

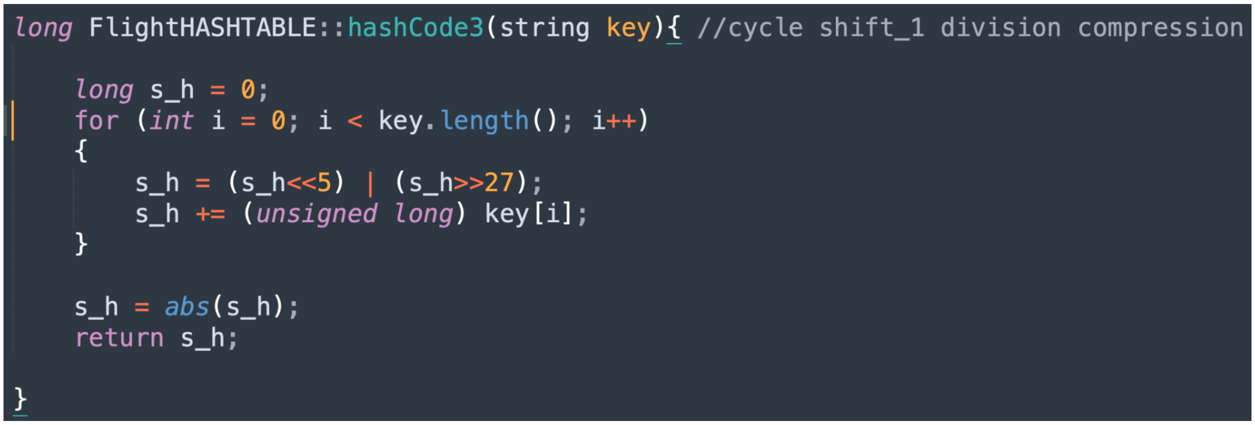
File #3(Flight-ticket100k.csv)



```
100000 flight schedules imported
>count_collisions
The number of collision(s) is/are 9
```

=> Avg. number of collisions per entry: 0.00009(0.009%)

HashCode function #3(cycle shift)



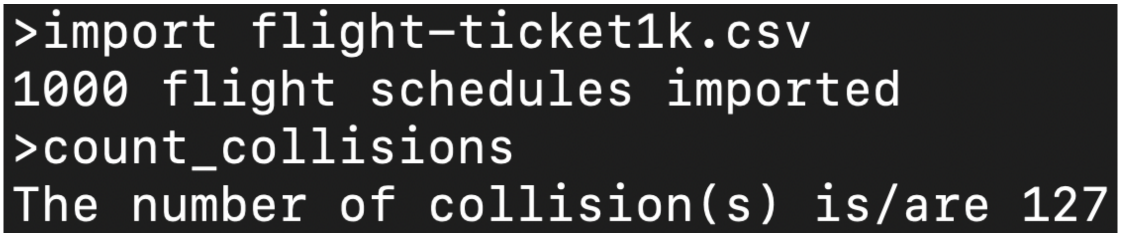
```
long FlightHASHTABLE::hashCode3(string key){ //cycle shift_1 division compression

    long s_h = 0;
    for (int i = 0; i < key.length(); i++)
    {
        s_h = (s_h<<5) | (s_h>>27);
        s_h += (unsigned long) key[i];
    }

    s_h = abs(s_h);
    return s_h;

}
```

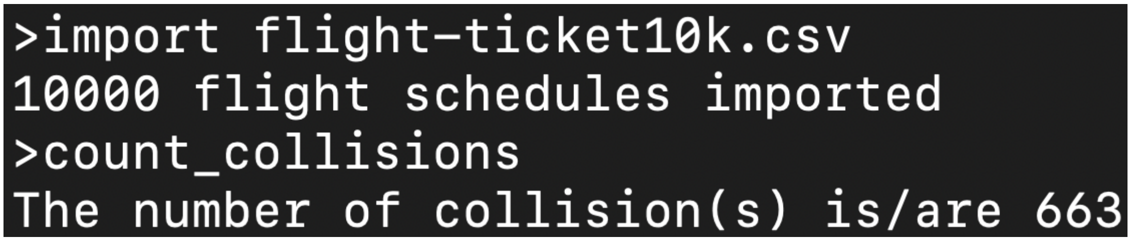
File #1(Flight-ticket1k.csv)



```
>import flight-ticket1k.csv
1000 flight schedules imported
>count_collisions
The number of collision(s) is/are 127
```

=> Avg. number of collisions per entry: 0.127(12.7%)

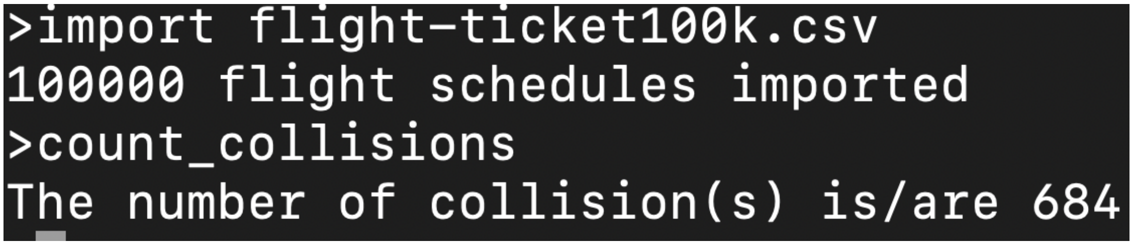
File #2(Flight-ticket10k.csv)



```
>import flight-ticket10k.csv
10000 flight schedules imported
>count_collisions
The number of collision(s) is/are 663
```

=> Avg. number of collisions per entry: 0.0663(6.63%)

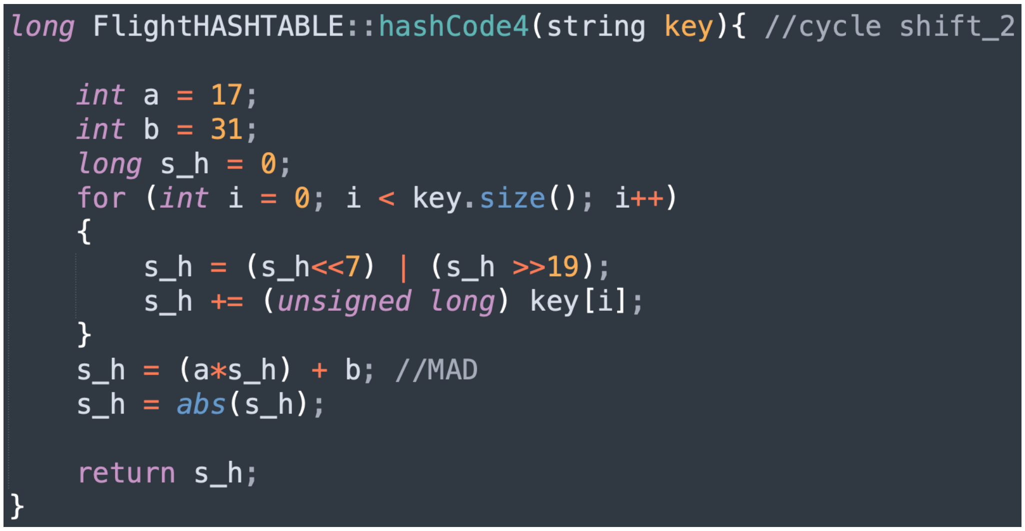
File #3(Flight-ticket100k.csv)



```
>import flight-ticket100k.csv
100000 flight schedules imported
>count_collisions
The number of collision(s) is/are 684
```

=> Avg. number of collisions per entry: 0.00684(0.684%)

HashCode function #4(cycle shift + MAD)

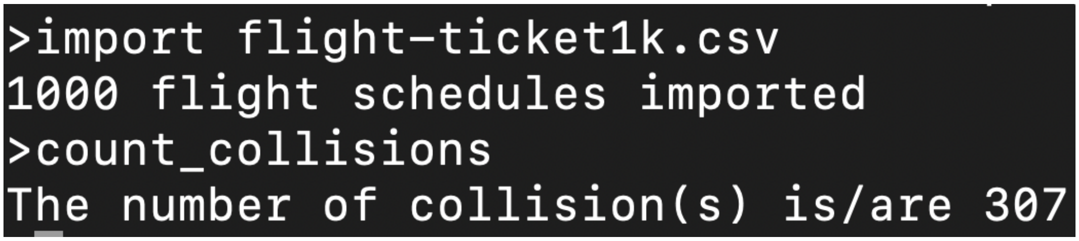


```
long FlightHASHTABLE::hashCode4(string key){ //cycle shift_2

    int a = 17;
    int b = 31;
    long s_h = 0;
    for (int i = 0; i < key.size(); i++)
    {
        s_h = (s_h<<7) | (s_h >>19);
        s_h += (unsigned long) key[i];
    }
    s_h = (a*s_h) + b; //MAD
    s_h = abs(s_h);

    return s_h;
}
```

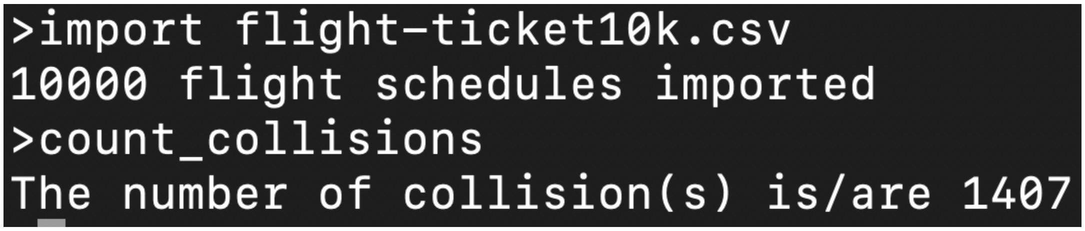
File #1(Flight-ticket1k.csv)



```
>import flight-ticket1k.csv
1000 flight schedules imported
>count_collisions
The number of collision(s) is/are 307
```

=> Avg. number of collisions per entry: 0.307(30.7%)

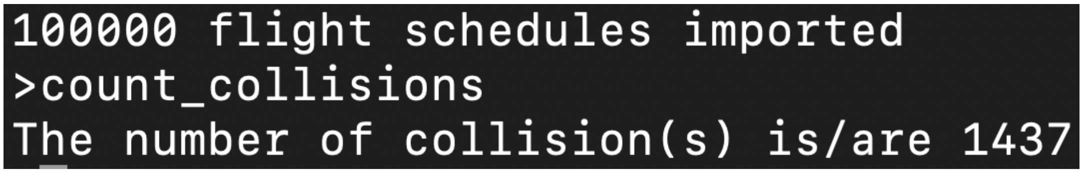
File #2(Flight-ticket10k.csv)



```
>import flight-ticket10k.csv
10000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1407
```

=> Avg. number of collisions per entry: 0.1407(14.07%)

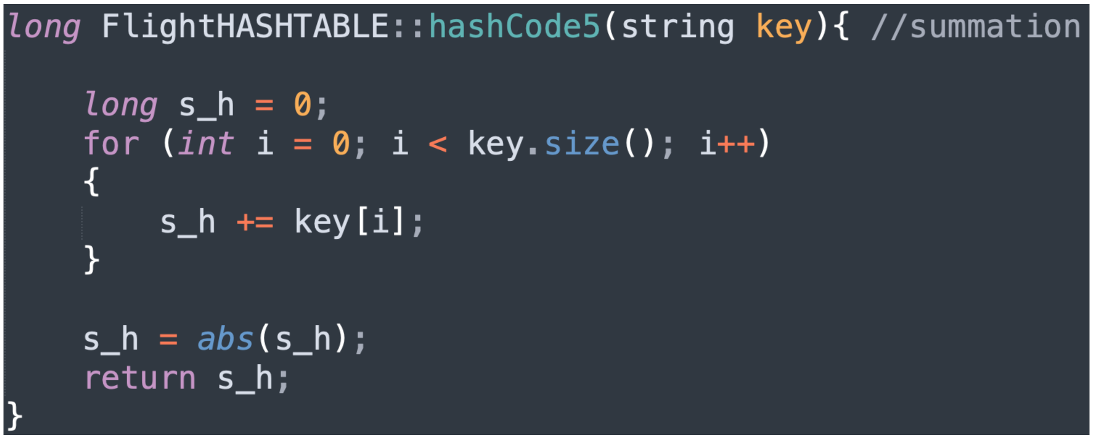
File #3(Flight-ticket100k.csv)



```
100000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1437
```

=> Avg. number of collisions per entry: 0.01437(1.437%)

HashCode function #5 (summation)

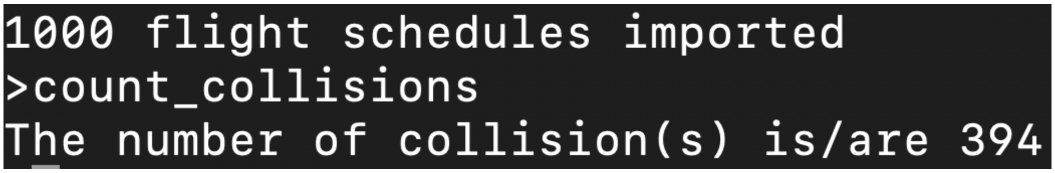


```
long FlightHASHTABLE::hashCode5(string key){ //summation

    long s_h = 0;
    for (int i = 0; i < key.size(); i++)
    {
        s_h += key[i];
    }

    s_h = abs(s_h);
    return s_h;
}
```

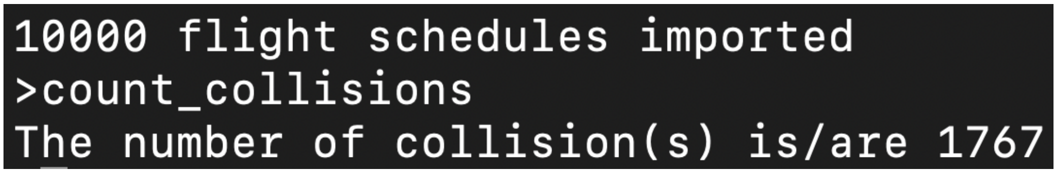
File #1(Flight-ticket1k.csv)



```
1000 flight schedules imported
>count_collisions
The number of collision(s) is/are 394
```

=> Avg. number of collisions per entry: 0.394(39.4%)

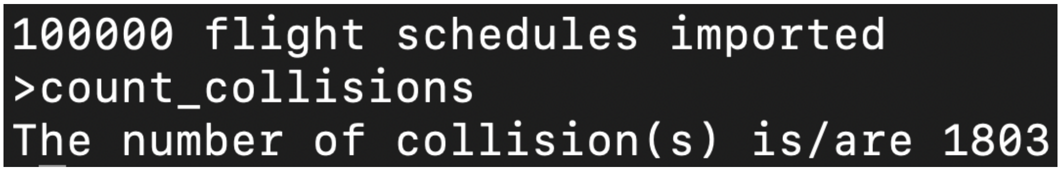
File #2(Flight-ticket10k.csv)



```
10000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1767
```

=> Avg. number of collisions per entry: 0.1767(17.67%)

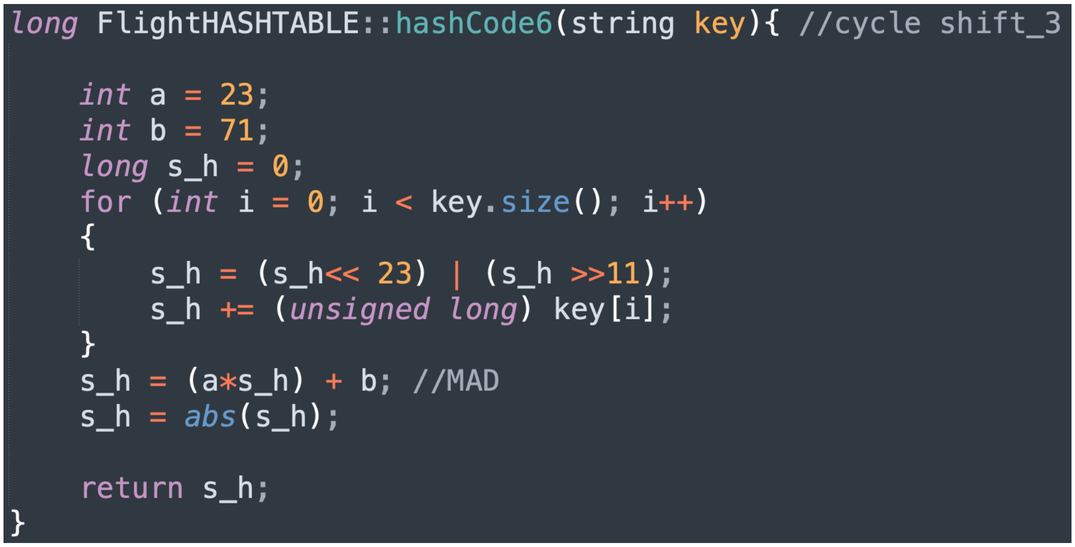
File #3(Flight-ticket100k.csv)



```
100000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1803
```

=> Avg. number of collisions per entry: 0.01803(1.803%)

HashCode function #6 (cycle shift + MAD)

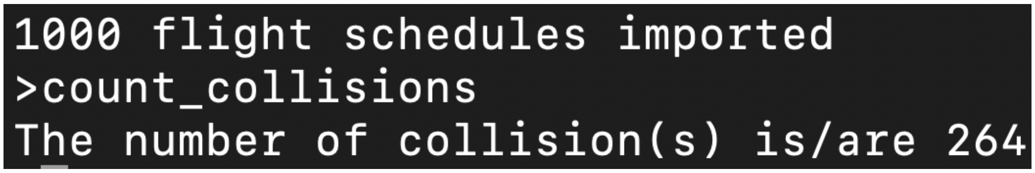


```
long FlightHASHTABLE::hashCode6(string key){ //cycle shift_3

    int a = 23;
    int b = 71;
    long s_h = 0;
    for (int i = 0; i < key.size(); i++)
    {
        s_h = (s_h<< 23) | (s_h >>11);
        s_h += (unsigned long) key[i];
    }
    s_h = (a*s_h) + b; //MAD
    s_h = abs(s_h);

    return s_h;
}
```

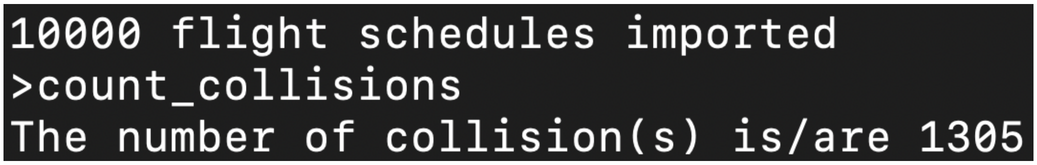
File #1(Flight-ticket1k.csv)



```
1000 flight schedules imported
>count_collisions
The number of collision(s) is/are 264
```

=> Avg. number of collisions per entry: 0.264(26.4%)

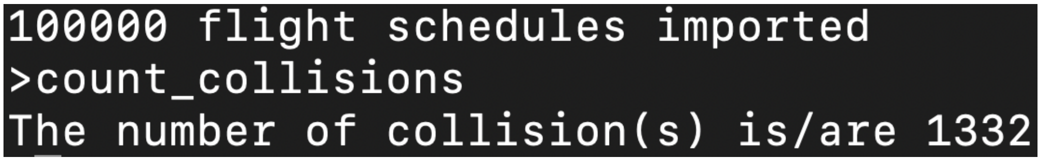
File #2(Flight-ticket10k.csv)



```
10000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1305
```

=> Avg. number of collisions per entry: 0.1305(13.05%)

File #3(Flight-ticket100k.csv)



```
100000 flight schedules imported
>count_collisions
The number of collision(s) is/are 1332
```

=> Avg. number of collisions per entry: 0.01332(1.332%)

Results

Since hashCode2 has the lowest average number of collisions per entry for all three files, it will be used in the final version of the assignment.