3. **Q3_PART_1:** Implement a **class** called **FlightNum**; the object contains the numerical part of an alphanumerical flight code; this numerical part must be in the range [1, 9999], endpoints included. Write and include in your project the header/cpp files implementing class FlightNum with the following public member functions:

```
///---------- FlightNum public member functions ------------///
FlightNum(); // constructor;

// only accept input within [1, 9999]; object initialized only once;
void inputNum(int inp_val);

// get flight number; print error message if the object is uninitialized
int getNum();

//print flight number on screen; message if the object is uninitialized
void printNum();

// Set input value randomly within [1, 9999].
void inputRandomNum();
//--------------------------------------------------------------//
```

Test your implementation with the following routine that can be called from your main.

```
///Test routine; uncomment a line when corresponding function is ready/
void test_Q3_part_1()
{
    FlightNum test_object, test_object2;
    int val, inp_val=768;

    //test_object.inputNum(inp_val);

    //val = test_object.getNum();
    //cout << "inp Val was: " << val << endl;
    //test_object.printNum();

    // cout << " Random input val: " << endl;
    //test_object2.inputRandomNum();
    //test_object2.printNum();
}
//--------------------------------------------------------------//
```

Note: You may use the “helper” function below, with global scope, if/where needed:

```
///Function generates random int in [0,maxval] endpoints included
int generateRandValueUpToMax(int maxval)
{
      int result = 0;
      double result_double;
      if(maxval>0)
      {
            //random integer within [0 , RAND_MAX]
            result=rand();

            // random non-integer number within [0 , maxval]
            result_double=( ((double)result)/(RAND_MAX) )*maxval;

            // random integer number within [0 , maxval]
            result=(int) result_double;
      }
      return result;
}
//--------------------------------------------------------------//
```

[Marks breakdown: 3 marks for each member function implemented and tested correctly]

[PART_1: 15 marks]

Q3_PART_2: Implement a **class** called **FlightCode**; the object contains the alphabetical part (tag) of an alphanumerical flight code; this part identifies an airline and can only be one of the following six tags: "AF, BA, KL, EZY, VS, SK"; these tags are represented using an **enumerator** FlightCodeOptions. Write and include in your project the header file (optionally also a cpp file) implementing class FlightCode with the following (public) member functions:

```
///----------- FlightCode public member functions -------------///
FlightCode(); // constructor;

// only accept input within the allowed range; initialize only once;
void inputTag(FlightCodeOptions inp_val);


// get flight tag; print an error message if the object is uninitialized
FlightCodeOptions getTag();

//print flight code on screen; message if the object is un initialized
void printTag();

// Set input tag randomly within the allowed set of values.
void inputRandomTag();
//-----------------------------------------------------------------//
```

Write a routine `void test_Q3_part_2()` similar to the one provided for PART_1 to test your implementation.

[Marks breakdown: 2 marks for each member function implemented and tested correctly]
[PART_2: 10 marks]
[Q3: 25 marks]