Object-oriented Programming – C++

Tutorial 3

1. Write a C++ program that dynamically allocates a vector large enough to hold a user defined number of test scores. Once all the scores are entered, the vector should be sorted in descending order. In addition, the program should also display the average score of the vector.

2. Write a C++ program that lets users enter an account number. The program should determine if the number is valid by checking for it in the following list:

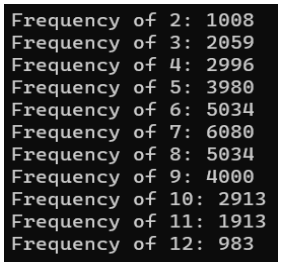
```
5658845 4520125 7895122 8777541 8451277 1302850
8080152 4562555 5552012 5050552 7825877 1250255
1005231 6545231 3852085 7576651 7881200 4581002
```

   The list of numbers above should be initialized in a vector. You should use the binary search algorithm from the STR library to check if the input account number appears in the vector. If the user enters a number in the vector, the program should display a message saying the number is valid. If the user enters a number not in the vector, the program should display a message indicating that the number is invalid.

   (Ref: https://cplusplus.com/reference/algorithm/binary_search/)

3. Write a C++ program to read in 20 numbers. As each number is read, validate it and store it in the vector only if it isn't a duplicate of a number already read. After reading all the values, display only the unique values that the user entered.

4. Write a C++ program that simulates the rolling of two dice. The sum of the two values should then be calculated and stored in a vector. [Note: Each die can show an integer value from 1 to 6, so the sum of the two values will vary from 2 to 12, with 7 being the most frequent sum and 2 and 12 being the least frequent sums.] Your program should roll the two dice 36,000 times. Use the vector to tally the number of times each possible sum appears. Print the results. *Observe if the totals are reasonable (i.e., there are six ways to roll a 7, so approximately one-sixth of all the rolls should be 7).*



Sample Output ***(frequency of sum can vary)***

Ref: https://www.bitdegree.org/learn/random-number-generator-cpp