In class Practice 9/27/2019

Group three or four people team.

For Part I questions, do solo first then discuss as a group.

For Part II and part III work as a team

3.for (size_t i{0}; i<myArray.size(); i++){
cout << "myArray[" << i << "]=" << myArray[i]<< endl;
}

Part I

1. const size_t arraySize{10};

1. Define a constant variable arraySize to represent the size of an array and initialize it to 10.
2. Declare an array call it *myArray* with arraySize elements of type float, and initialize the elements to 0. 2. array<float, arraySize> myArray {0.0};
3. Display all the myArray elements using a counter-controlled for statement. Define the integer variable i as a control variable for the loop.
4. Display all the myArray elements separated by spaces using a *range-based for* statement. 4. for(float item : myArray) {cout<< item <<" ";}
5. Declare the array call *mesh* to store int values and to have 5 rows and 5 columns. Assume that the constant variable *arraySize* has been defined to be 5. 5.array<array<int, arraySize>, arraySize> mesh;
6. To refer to a particular location or element within an array, we specify the name of the array and the value of the particular element. 6.False. The name of the array and the subscript of the array are specified
7. Create and initialize each of the 4 elements of one-dimensional integer array values to 8.
8. Determine and display the smallest and largest values contained in 55-element floating point array wk.
9. Write a nested *range-based for* statement and *auto* type inference that initializes each element of *temp* to zero.
10. What is the following do?

```
#include <iostream>
#include <array>
using namespace std;

const size_t arraySize{10};
void someFunction(const array<int, arraySize>&, size_t); // prototype

int main() {
   array<int, arraySize> a{1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

   cout << "The values in the array are:" << endl;
   someFunction(a, 0);
   cout << endl;
}

// What does this function do?
void someFunction(const array<int, arraySize>& b, size_t current) {
   if (current < b.size()) {
      someFunction(b, current + 1);
      cout << b[current] << "  ";
   }
}
```

10.print the array in reversed order

7. array<int, 4> values{8, 8, 8, 8};

8.double smallest {wk[0]};
double largest {wk[0]};
for (double element : wk){
if (element<smallest){
smallest= element;
} else if (element> largest){
largest= element;
}
}

9.for (auto &row： temp){
for(auto &element : row){
element = 0;
}
}

Part II

Find the errors of the following program segments

1. Find the error
   Off-by-one error

```
array<int, 10> b{};
for (size_t i{0}; i <= b.size(); ++i)
{ b[i] = 1; }
```

i < b.size()

2. Find the error(s) from the following program segment which creates 2-d array of *10* rows and *8* columns and initialize values like the following

0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14
8	9	10	11	12	13	14	15
9	10	11	12	13	14	15	16

```
#include <iostream>
#include <array>;
using namespace std
int main() {
        array<array<int, 10> , 8> b{};
        for (size_t i{0}; i <= b.size(); ++i) {
                for (size_t j{0}; j <=b[i].size(); ++j) {
                        b[i, j] = i+j;
                        cout << b[i,j] << "\t";
                }
                cout <<endl;
        }
        return 0;
}
```

array<int, 8> , 10

i < b.size()

j < b[i].size()

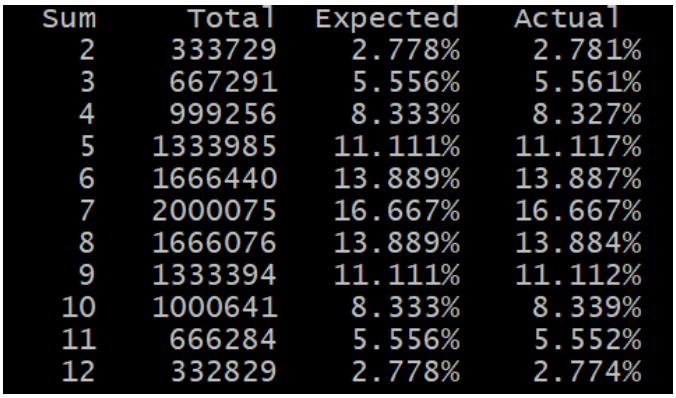
b[i][j]

b[i][j]

Part III

Q1

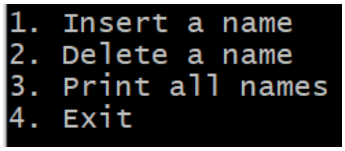
Write a program which will throw 2 dice for 12,000,000 times. Assuming uniform distribution. We like to output the statistic as the following. The first column is the sum from 2 to 12. Second column is the total number of occurrences for each sum. The third column is the expected percentage and the fourth column is the actual percentage. You must use C++ 11 default_random_engine to generate these random numbers (use time(0) as the seed) and uniform_int_distribution.



Sum	Total	Expected	Actual
2	333729	2.778%	2.781%
3	667291	5.556%	5.561%
4	999256	8.333%	8.327%
5	1333985	11.111%	11.117%
6	1666440	13.889%	13.887%
7	2000075	16.667%	16.667%
8	1666076	13.889%	13.884%
9	1333394	11.111%	11.112%
10	1000641	8.333%	8.339%
11	666284	5.556%	5.552%
12	332829	2.778%	2.774%

Q2.

Write a C++ program that uses vector to store names. Your system will output a menu as the following



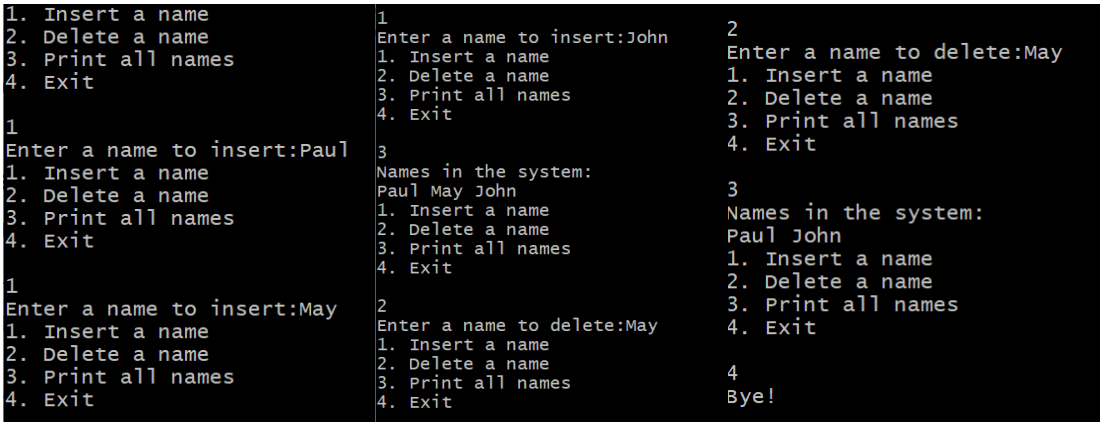
If user chose 1, the system will ask user to enter a name. This name will then be stored in the vector.

If user chose 2, the system will remove this name from the system.

If user chose 3, the system will print all names currently in the system.

If user chose 4, the program will end.

The followings are test runs of your system.



To remove an element from vector given element's value, you can use combination of algorithm std::remove and the erase member function of vector.

See https://en.wikipedia.org/wiki/Erase%E2%80%93remove_idiom