# PROGRAMMING WITH C++

## LAB 7

Xi'an Jiaotong-Liverpool University
西交利物浦大学

# Example 1: Minimum Element in an Array

**Design and Write a function that finds the smallest element in an array of integers using the following header:**

**double min(double array[], int size)**

**Write a test program that prompts the user to enter ten numbers, invokes this function, and displays the minimum value.**

**Here is the sample run of the program:**

**Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2**

**The minimum number is: 1.5**

# Example 2: Number Of Letters In The String

**Write a function that counts the number of letters in the string using the following header:**

```
int countLetters(const char s[])
```

**Write a test program that reads a C-string and displays the number of letters in the string.**

**Here is a sample run of the program:**

**Enter a string: 2026 is coming**

**The number of letters in 2026 is coming is 8**

# Example 3: Analyze Numbers

Read one hundred numbers, compute their average, and find out how many numbers are above the average.

```
Enter a new number: 35
Enter a new number: 36
Average is 35.5
Number of elements above the average 1
```

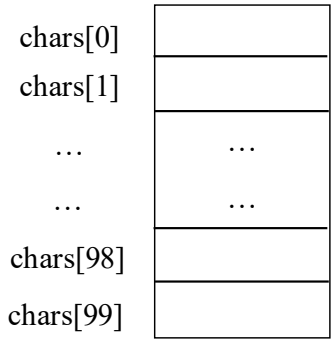
# AnalyzeNumbers.cpp

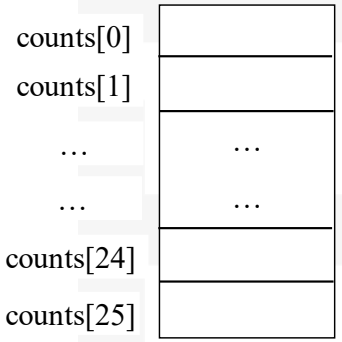
```
#include <iostream>
using namespace std;
int main()
{
  const int NUMBER_OF_ELEMENTS = 2;
  double numbers[NUMBER_OF_ELEMENTS];
  double sum = 0;
  for (int i = 0; i < NUMBER_OF_ELEMENTS; i++)
  {
    cout << "Enter a new number: ";
    cin >> numbers[i];
    sum += numbers[i];
  }
  double average = sum / NUMBER_OF_ELEMENTS;
  int count = 0; // The number of elements above average
  for (int i = 0; i < NUMBER_OF_ELEMENTS; i++)
    if (numbers[i] > average)
      count++;
  cout << "Average is " << average << endl;
  cout << "Number of elements above the average " << count << endl;
  return 0;
}
```

```
Enter a new number: 35
Enter a new number: 36
Average is 35.5
Number of elements above the average 1
```

# Example 4: Counting Occurrence of Each Letter

- Generate 100 lowercase letters randomly and assign to an array of characters.
- Count the occurrence of each letter in the array.




```
The lowercase letters are:
f c y f g c i n h s w g v r u h a t s g
k m a v z w t u f h g x m m x r m m s r
j d r f j e q d s b a v w j u z m s i t
x o x a i a k y e x b l z j x r h j z q
m j k s j a q k p h h z n k d e o o h m

The occurrences of each letter are:
6 a 2 b 2 c 3 d 3 e 4 f 4 g 7 h 3 i 7 j
5 k 1 l 8 m 2 n 3 o 1 p 3 q 5 r 6 s 3 t
3 u 3 v 3 w 6 x 2 y 5 z
```

# CountLetterInArray.cpp

```
#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std;
const int NUMBER_OF_LETTERS = 26;
const int NUMBER_OF_RANDOM_LETTERS = 100;
void createArray(char []);
void displayArray(const char []);
void countLetters(const char [], int []);
void displayCounts(const int []);

int main()
{
  // Declare and create an array
  char chars[NUMBER_OF_RANDOM_LETTERS];
  // Initialize the array with random lowercase letters
  createArray(chars);
  // Display the array
  cout << "The lowercase letters are: " << endl;
  displayArray(chars);
  // Count the occurrences of each letter
  int counts[NUMBER_OF_LETTERS];
  // Count the occurrences of each letter
  countLetters(chars, counts);
  // Display counts
  cout << "\nThe occurrences of each letter are: " << endl;
  displayCounts(counts);
  cout <<endl;
  return 0;
}
// Create an array of characters
void createArray(char chars[])
{
  // Create lowercase letters randomly and assign
  // them to the array
  srand(time(0));
  for (int i = 0; i < NUMBER_OF_RANDOM_LETTERS; i++)
    chars[i] = static_cast<char>('a' + rand() % ('z' - 'a' + 1));
}
// Display the array of characters
void displayArray(const char chars[])
{
  // Display the characters in the array 20 on each line
  for (int i = 0; i < NUMBER_OF_RANDOM_LETTERS; i++)
  {
    if ((i + 1) % 20 == 0)
      cout << chars[i] << " " << endl;
    else
      cout << chars[i] << " ";
  }
}
// Count the occurrences of each letter
void countLetters(const char chars[], int counts[])
{
  // Initialize the array
  for (int i = 0; i < NUMBER_OF_LETTERS; i++)
    counts[i] = 0;

  // For each lowercase letter in the array, count it
  for (int i = 0; i < NUMBER_OF_RANDOM_LETTERS; i++)
    counts[chars[i] - 'a'] ++;
}
// Display counts
void displayCounts(const int counts[])
{
  for (int i = 0; i < NUMBER_OF_LETTERS; i++)
  {
    if ((i + 1) % 10 == 0)
      cout << counts[i] << " " << static_cast<char>(i + 'a') << endl;
    else
      cout << counts[i] << " " << static_cast<char>(i + 'a') << " ";
  }
}
```

```
The lowercase letters are:
f c y f g c i n h s w g v r u h a t s g
k m a v z w t u f h g x m m x r m m s r
j d r f j e q d s b a v w j u z m s i t
x o x a i a k y e x b l z j x r h j z q
m j k s j a q k p h h z n k d e o o h m

The occurrences of each letter are:
6 a 2 b 2 c 3 d 3 e 4 f 4 g 7 h 3 i 7 j
5 k 1 l 8 m 2 n 3 o 1 p 3 q 5 r 6 s 3 t
3 u 3 v 3 w 6 x 2 y 5 z
```

# Example 5: Grading Multiple-Choice Test

Students' Answers to the Questions:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Student 0 | A | B | A | C | C | D | E | E | A | D |
| Student 1 | D | B | A | B | C | A | E | E | A | D |
| Student 2 | E | D | D | A | C | B | E | E | A | D |
| Student 3 | C | B | A | E | D | C | E | E | A | D |
| Student 4 | A | B | D | C | C | D | E | E | A | D |
| Student 5 | B | B | E | C | C | D | E | E | A | D |
| Student 6 | B | B | A | C | C | D | E | E | A | D |
| Student 7 | E | B | E | C | C | D | E | E | A | D |

- Objective: write a program that grades multiple-choice test.

Key to the Questions:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Key | D | B | D | C | C | D | A | E | A | D |

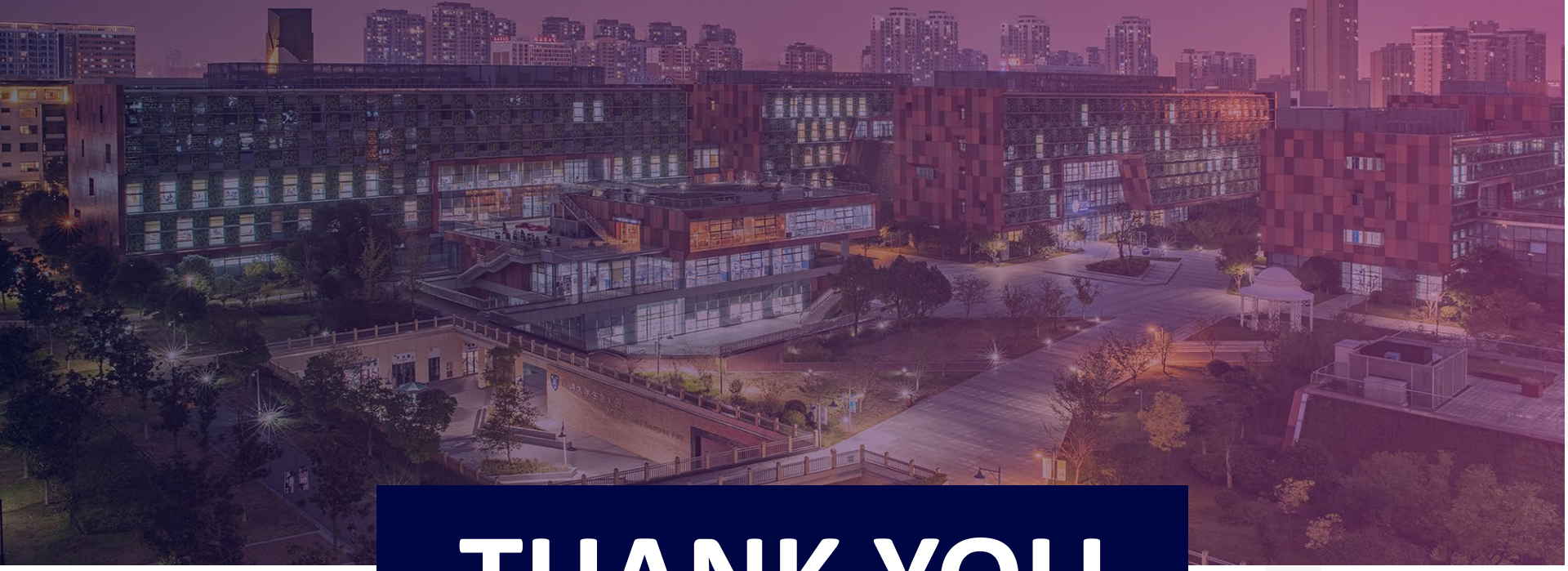
## GradeExam.cpp

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

int main()
{
  const int NUMBER_OF_STUDENTS = 8;
  const int NUMBER_OF_QUESTIONS = 10;

  // Students' answers to the questions
  char
answers[NUMBER_OF_STUDENTS][NUMBER_OF_QU
ESTIONS] =
  {
    {'A', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
    {'D', 'B', 'A', 'B', 'C', 'A', 'E', 'E', 'A', 'D'},
    {'E', 'D', 'D', 'A', 'C', 'B', 'E', 'E', 'A', 'D'},
    {'C', 'B', 'A', 'E', 'D', 'C', 'E', 'E', 'A', 'D'},
    {'A', 'B', 'D', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
    {'B', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
    {'B', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
    {'E', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'}
  };
  // Key to the questions
  char keys[] = {'D', 'B', 'D', 'C', 'C', 'D', 'A', 'E', 'A', 'D'};
  // Grade all answers
  for (int i = 0; i < NUMBER_OF_STUDENTS; i++)
  {
    // Grade one student
    int correctCount = 0;
    for (int j = 0; j < NUMBER_OF_QUESTIONS; j++)
    {
      if (answers[i][j] == keys[j])
        correctCount++;
    }
    cout << "Student " << i << "'s correct count is " <<
      correctCount << endl;
  }
  return 0;
}
```

```
Student 0's correct count is 7
Student 1's correct count is 6
Student 2's correct count is 5
Student 3's correct count is 4
Student 4's correct count is 8
Student 5's correct count is 7
Student 6's correct count is 7
Student 7's correct count is 7
```



# THANK YOU

## VISIT US

WWW.XJTLU.EDU.CN

## FOLLOW US

Xi'an Jiaotong-Liverpool University
西交利物浦大学