**Practical 8 (Part B)**

**Two dimensional Array**

1. Create a C++ program to perform the following:
   1. Declare a two-dimensional array named studs, which is used to store up to 10 student names and student IDs as shown in Figure 1 below. Initialize the array with empty string.
   2. Declare an integer array named marks, which is used to store each student’s mark. Initialize the marks array elements to 0.
   3. Prompt user to enter up to 10 students’ names, IDs and their marks. Store the data into the correspondent studs array and marks array.

Remark: You may use a while (or do-while) loop. Ensure stop prompting for input after 10 iterations. Example of input:

Enter Student Name: -> Leng Wii Wii

Enter Student ID: -> WAA12345

Enter Mark: -> 98

Press [Y] for more record, press [N] to stop

Option: -> Y

Enter Student Name: ->

* 1. Tabularize all the student names, student IDs, and their marks retrieved from the studs and marks arrays. Example of output screen is as follows:

No. Student Name Student ID Mark

=== =========== ======== ====

1 Leng Wii Wii WAA12345 98

2 Chong Yen Yen WAA23456 90

3 Laurence Lim WAA34567 88

4 Chris Leong WAA45678 80

5 Alan Wong WAA56789 65

6 Tham Hsing Hsing WAA67890 76

7 Wayne Wong WAA78901 97

8 Wong Yong Yong WAA89012 68

End of Data

Figure 1

* 1. Use nested *for* loop to obtain and display the highest mark and the minimum mark of the above. Display the correspondent student’s name. Lastly, your program should calculate and display the average marks of the whole class.

1. Suppose the weekly hours for all employees are stored in a two-dimensional array. Each row records an employee’s seven-day work hours with seven columns. For example, the following array stores the work hours for eight employees. Write a C++ program that displays employees and their total hours worked. Initialize the array with the values shown in the following table.

Sun Mon Tue Wed Thu Fri Sat

Employee 0 2 4 3 4 5 8 8

Employee 1 7 3 4 3 3 4 4

Employee 2 3 3 4 3 3 2 2

Employee 3 9 3 4 7 3 4 1

Employee 4 3 5 4 3 6 3 8

Employee 5 3 4 4 6 3 4 4

Employee 6 3 7 4 8 3 8 4

Employee 7 6 3 5 9 2 7 9

1. Write a C++ program that reads ten numbers (store in a floating-point array size 10), computes their average, and finds out how many numbers are above the average. Use for loops wherever possible**.**
2. Write a C++ program that repeatedly prompts the user to enter a capital for a country. Upon receiving the user input, the program reports whether the answer is correct. A sample output is shown below:

What is the capital of AUSTRALIA : canberra

Your answer is correct

Do you want to continue (Y/N) : y

What is the capital of VIETNAM : hanoi

Your answer is correct

Do you want to continue (Y/N) : n

You have 2 correct answers

Assuming that there are ten countries and their capitals are stored in a two-dimensional array. The program prompts the user to answer the countries’ capital and displays the total correct count.

1. Trace and understand the following program:

#include <iostream>

#define NUM\_ROWS 3

#define NUM\_COLS 5

using namespace std;

void totalElement (int [][NUM\_COLS]);

void sumRows (int [][NUM\_COLS]);

void sumColumns (int [][NUM\_COLS]);

int main()

{ int numbers[NUM\_ROWS][NUM\_COLS] = { {2, 7, 9, 6, 4},

{6, 1, 8, 9, 4},

{4, 3, 7, 2, 9} };

totalElement(numbers);

sumRows(numbers);

sumColumns(numbers);

return 0;

}

//Summing all the elements of a two-dimensional array

void totalElement (int numbers[][NUM\_COLS])

{ int total = 0;

for (int row=0; row<NUM\_ROWS; row++)

{

for (int col=0; col<NUM\_COLS; col++)

{

cout << numbers[row][col] << " ";

total += numbers[row][col];

}

cout << endl;

}

cout << "\nThe total is : " << total << endl << endl;

}

//Summing the rows of a two-dimensional array

void sumRows (int numbers[][NUM\_COLS])

{ int total = 0;

for (int row=0; row<NUM\_ROWS; row++)

{

total = 0;

for (int col=0; col<NUM\_COLS; col++)

total += numbers[row][col];

cout << "The total for row " << row + 1 << " is : "

<< total << endl;

}

cout << endl;

}

//Summing the columns of a two-dimensional array

void sumColumns (int numbers[][NUM\_COLS])

{ int total = 0;

for (int col=0; col<NUM\_COLS; col++)

{

total = 0;

for (int row=0; row<NUM\_ROWS; row++)

total += numbers[row][col];

cout << "The total for column " << col + 1 << " is : "

<< total << endl;

}

cout << endl;

}

**Extra exercise**

1. Below shows three different test marks for the students in a tuition centre.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name | English | Malay | Maths |
| Ali | 68 | 89 | 76 |
| Bobby | 87 | 73 | 79 |
| Chong | 59 | 55 | 90 |
| David | 85 | 70 | 68 |
| Elisa | 88 | 83 | 90 |

1. Declare a two-dimensional array named testMarks and initialize it with the marks above. Please note that each row of the array is correspondent to one student’s results.
2. Create a one-dimensional array named studNames and initialize it with the student names as shown above. Similarly create another array named tests and initialize with the test names (English, Malay and Maths).
3. Create a function named search\_stud\_index, which will accept a student name. If the student name is found in the studNames array, then returns the index of the item in the array, else returns -1.
4. Create a function named get\_highest\_mark that will accept a string value. This function will call the search\_test\_index to obtain the correspondent tests index based on the given string. If the index is not -1, then based on the index, it will search and display the correspondent student name (from the studNames array) with the highest mark of the given test.

Example of the function call:

get\_highest\_mark("English");

Example of output of the function :

Highest Mark for English: 88 Name: Elisa

If search\_test\_index is not successful (returns -1), then it will call the search\_stud\_index to obtain the correspondent studNames index. If the index is not -1, then based on the index, it will search and display the highest mark of the tests.

Example of the function call:

get\_highest\_mark("Ali");

Example of output of the function :

Highest Mark for Ali: 89 Test: Malay

Display “Invalid parameter” if the parameter of the function is invalid.

1. Create a function named get\_average\_mark that will accept string. This function will display the correspondent average mark of a specific subject or a specific student.

Example of the function output that calculates the average mark for English is as below:

77.40

Example of the function output that calculates the average mark for Ali is as below

77.67

1. Create a function named display\_record that will tabularize the data as below. Call the get\_average\_mark functions that you have defined above to display the record of the last row and the last column below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student Name | English | Malay | Maths | Average |
| Ali | 68 | 89 | 76 | 77.67 |
| Bobby | 87 | 73 | 79 | 79.67 |
| Chong | 59 | 55 | 90 | 68.00 |
| David | 85 | 70 | 68 | 74.33 |
| Elisa | 88 | 83 | 90 | 87.00 |
| Average | 77.40 | 74.00 | 80.60 |  |

1. Write the main function that displays the highest mark and the lowest mark for each student and for each test. Then display the student mark summary by calling the display\_record function you have written above.