Lecture 13
Declaring a 2D Array

• Model: type name[row_size][column_size]

• Example:
  
  int grades[10][20];

  string students[10][20];
2D Array data structure

• Say we have the following array:

• int grades[4][8];

• Here is the graphical representation:
Accessing Elements in the 2D Array

- If we have an 2D array declare as the following: int grades[4][8];

- The elements of the array are as follows: grades[1], grades[1], grades[1], grades[3]

- Each first dimension “element” represent an array of 8 elements

- Then We can assign values to grades[0] as follows: grades[0][0]=0, grades[0][1]=1, grades[0][2]=2, grades[0][3]=3, grades[0][4]=4, grades[0][5]=5, grades[0][6]=6, grades[0][7]=7;
Print 2D array

• We usually use nested for loop to printing out elements of the 2D array. Code would look like this for grades[4][8]:

```cpp
for( int i = 0 ; i < 4 ; ++i ){
    for( int j = 0 ; j < 8 ; ++j )
        cout << grades[i][j] << " ";
    cout << endl;
}
```
Initializing the 2D array

- Sometimes we want to pre-initialize the array, we can do the following:

  ```c
  int grades[3][2]= {{97,93},{87,83},{77,73}};
  ```

- Sometimes we want to initialize the entire array to zero, we can do the following:

  ```c
  int grades[5][10]= {0};
  ```

- `{0}` is a special code to C++, `{1}` doesn’t work.
2D Arrays and Functions

• Like regular arrays, two dimensional arrays can be passed into sub-functions, and they are always passed by reference.

• It is important to note: If the function is trying to access the entire 2D array or an element of the 2D array, which is the 1D array.
Passing 2D array

• To pass entire 2D array into the function
  int gradesSet[10][20];
  printAllScore(gradesSet,rowsize,colsSize);
  void printAllScore(int gradesSet[][20], int row, int col)

• The COLUMN SIZE of the 2D array MUST be provided while row size is optional.

• To pass 1 element of the 2D array into the function
  int gradesSet[10][20];
  printRowScore(gradesSet[0],colsSize);
  void printRowScore(int gradesSet[][20], int col)
Exercise

• Exercise 1: Create a 2-dimensional array with 4 rows and 30 columns. Row represents sections of a course and column represents the students, value inside each position of the array is the Final exam grade for each students. Fill the array with random numbers between 40 and 100. Calculate the total, average, maximum, minimum for each section.

• Exercise 2: Create a 2-dimensional array with 10 rows and 10 columns. Fill the array with random 3 digit integers. (rand()%900 + 100). Print out the column with the largest sum. (If two or more columns share the largest sum, print out one of them only.)

• Exercise 3: Write a complete C++ program that does the following.
  a. It asks the user to enter a positive integer value, r that is at most 100.
  b. The program reads a value entered by the user. If the value is not in the right range, the program should terminate.
  c. The program reads and stores r integers from the user and then prints a pattern of r rows of stars, the lengths of which are the other integers entered by the user.
For example, the following represents one run of the program. How many rows? 4
Enter 4 row lengths: 2 7 1 5
**
******
*
*****
Exercise

• Exercise 4: Write a complete C++ program that does the following.
  a. It asks the user to enter a 5-digit integer value, n.
  b. The program reads a value entered by the user. If the value is not in the right range, the program should terminate.
  c. The program calculates and stores the 5 individual digits of n.
  d. The program outputs a “bar code” made of 5 lines of stars that represent the digits of the number n.
For example, the following represents one run of the program. (The user chooses the number 16384.)
Enter a 5 digit integer: 16384
* 
******
***
*******
****

• Exercise 5: Write a complete C++ program that does the following.
  1. It asks the user to enter 9 integers as the entries of a 3 x 3 table.
  2. The program reads the 9 entries, row by row and prints the table.
  3. If every row and column of the table have the same sum then the program adds the message: MAGIC.
Here is an example of how the program should work: Enter 9 entries of a 3 x 3 table: 10 14 18 15 16 11 17 12 13
output:
10 14 18
15 16 11
17 12 13
MAGIC
Homework 10

• Question 1: Write a program that takes a word from user until the user input “stop”, then output all the word backward which the user previous input. Please use array to do it, not the recursion. (Max word capacity is 10, so user does not allow to input more than 10 words.) Example output:

Enter a word ('stop' to end program): guozhen
Enter a word ('stop' to end program): am
Enter a word ('stop' to end program): I
Enter a word ('stop' to end program): Hello
Enter a word ('stop' to end program): stop
Hello I am guozhen

• Question 2: Write a program that reads in ten numbers and displays distinct numbers. If a number appears multiple times, it is displayed only once. For example: if user enters 1,1,2,3,4,4,5,1,0,9. You should output: 1,2,3,4,5,0,9. Order doesn’t matter.
Question 3: Eight queens are to be placed on an 8 x 8 chessboard in such a way that one queen is to be in each row. A program will store an array x[] with capacity 8 to represent such a configuration. If x[r] has value c then in row r there is a queen in column c. Write a program that asks a user to enter the columns that contain queens in the 8 rows. The program makes sure there are no queens in the same column. The program then prints the board. For example, if the user enters: 0,3,4,0,0,7,6,7 the program should print: No Good. (Hint: Use nested for loop, if column c == array[r] Print ‘Q’, otherwise Print ‘.’) For example, if the user enters: 1,3,4,5,2,8,6,7 the program should print:

Q......
..Q....
...Q...
....Q...
.Q......
.......Q
.....Q
....Q.
......Q.