

Programming Assignment 6

Prof. Zhang

Fall 2018

Write a program that assigns passenger seats in an airplane.

main function:

1. Prompt the user to enter the model of the airplane, and the number of rows and columns in this model.
2. Allocate and initialize a 2D array to represents the seat assignments of the airplane.
3. Display the initial assignments (when no seat has been assigned, and all are available). Below is an example of small airplane with 8 rows and 4 seats per row.

Eagle126: 32 seats remaining

```
1 A B C D
2 A B C D
3 A B C D
4 A B C D
5 A B C D
6 A B C D
7 A B C D
8 A B C D
```

Alternatively, you can display the assignments as follows:

Eagle126: 32 seats remaining

```
  A B C D
1 O O O O
2 O O O O
3 O O O O
4 O O O O
5 O O O O
6 O O O O
7 O O O O
8 O O O O
```

O: available, X: unavailable

4. Repeated
 - a) ask the user to choose a seat by entering a string such as "2D".
 - b) If the seat is already taken, display an error message and ask the user to try again. if the seat is available, update the 2D array to record the seat is now taken and display the updated assignment:
 - c) display all seats that have been taken in the order they were assigned. (you can use a vector to store them)

Eagle126: 31 seats remaining

```

1 A B C D
2 A B C X
3 A B C D
4 X B C D
5 A B C D
6 A B C D
7 A B C D
8 A B C D

```

seats assigned: 2D 4A

until the user does not want to continue or there is no more available seat.

5. Delete the 2D array, and go back to 1 (choose a different model ...)

Program Organization

```
typedef bool* BoolPtr;
```

1. Write a function that allocates, initializes and returns the address of the array (each array element points to an array of bool). The function should take the parameters: row_num, and col_num.

```
/* dynamically allocate @row_num number of arrays (each of size @col_num)
   return the address of array (of pointers
*/
```

```
*/
```

```
BoolPtr * CreateArrayOfArrays (int row_num, int col_num)
```

```
{
```

```
    BoolPtr * p = NULL; //a pointer variable to store the array of addresses
```

```
    assert (row_num>0 && col_num>0);
```

```
    p = new BoolPtr[row_num];
```

```
    for (int i=0; i<row_num;i++){
```

```
        p[i] = new bool[col_num]; //allocate array for i-th row
```

```
        for (int j=0;j<col_num;j++)
```

```
            p[i][j] = false; //init every col of i-th row to false
```

```
    }
```

```
    return p;
```

```
}

```

- Write a function that display seat assignments given by a 2D array, with a specified row number and col number

```
/* b is an array of pointers (pointing to each row-array)
   row_num is the length of array b,
   col_num is the length of row-arrays
*/
void DisplaySeats (BoolPtr b[ ], int row_num, int col_num)
```

- Write a function that calculates how many remaining seats are available for a given seat assignments (an array of arrays, and the row and col number)

```
int RemainingSeats (BoolPtr b[ ], int row_num, int col_num)
```

Hints:

- You can start simple by using a 2D static array to implement the whole functionalities.
- Try to first implement all functionalities in main():

Please refer to the notes from lectures about how to allocate an array of arrays:

- allocate an array of pointer types of certain size (here, r, the number of row)
- write a for loop to allocate r arrays, each array of size given by the number of column (seats per row). The type of array element can be bool, or char.
- Refer your note about how to initialize elements in the above array of arrays (also, this will show you a way to access each elements in the array of arrays in row-by-row pattern).

- After finish the above task, try to modularize your code and create the three functions as required above.

Helper functions:

```
// if seat is 'A', return 0
// 'B', return 1
// ...
```

```
int CharToInt (char seat)
{
    return (seat-'A'); //As the ascii code for letters are consecutive integers
}

// if col is 0, return 'A'
//1, return 'B'...
char ColToLetter (int col)
{
    return ('A'+col);
}

// row=0, col=0: return string "1A"
// row=3, col=1: return string "4B",....
string SeatString (int row, int col)
{
    string result;

    row++;
    result = to_string(row); //a C++ library that convert int to string, for details:
        //http://www.cplusplus.com/reference/string/to_string/

    result = result+ColToLetter(col); //concatenate the Column letter to result

    return result;
}
```

Extra credits:

Define a class to encapsulate all variables/data used to store the seats assignments (the pointer to array of arrays, row number, col number) into a class, and implement the three functions as a member function of the class.

To submit:

submit2000 LAB6 lab6.cpp