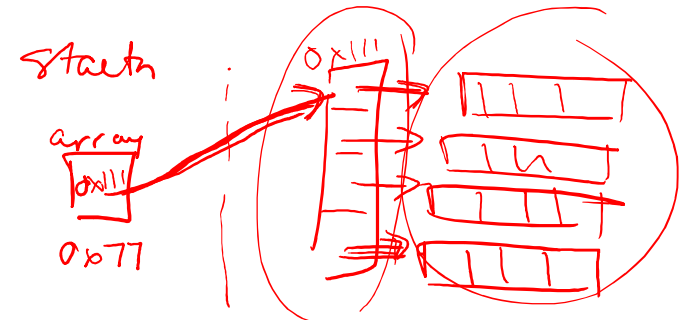


CS 161, Lecture 20: Practice with Arrays and Command Line – 28 February 2018

*int ** array = new int * [rows];
for each row
array[i] = new int [cols];*

Exercise: Multiplication Table



- Design a small program which will take the number of rows and columns from the user and generate a multiplication table.

Ex:

	0	1	2	3
0	0	1	2	3
1	0	1	2	3
2	0	2	4	6
3	0	3	6	9

*int ** array*

array[2][2]

*i * j*

- What functions do we need?
- What kind of memory should we use?
- Write the prototypes for each function.

→ mult_array

pop_array
init_array
delete_array

*ints
get rows
get cols*

print_array

Demo

```
access.engr.orst.edu - PuTTY
1 #include <iostream>
2
3 using namespace std;
4 //create memory on the heap and return the
5 //first address
6 int** init_array(int rows, int cols) {
7     int** ar = new int*[rows];
8     for(int i=0; i<rows; i++) {
9         ar[i] = new int[cols];
10    }
11    return ar;
12 }
13 //print the array contents
14 void print_array(int** array, int rows, int cols) {
15     for(int i=0; i<rows; i++) {
16         for(int j=0; j<cols; j++) {
17             cout << array[i][j] << " ";
18         }
19         cout << endl;
20     }
21 }
22 //set all values in the array to zero
23 void pop_array(int** array, int rows, int cols) {
24     for(int i=0; i<rows; i++) {
```

1,1 Top

Type here to search 7:48 AM 3/1/2018

```
22 //set all values in the array to zero
23 void pop_array(int** array, int rows, int cols) {
24     for(int i=0; i<rows; i++) {
25         for(int j=0; j<cols; j++) {
26             array[i][j] = 0;
27         }
28     }
29 }
30 //set all values in the array to the product of
31 //the indices
32 void mult_array(int** array, int rows, int cols) {
33     for(int i=0; i<rows; i++) {
34         for(int j=0; j<cols; j++) {
35             array[i][j] = i*j;
36         }
37     }
38 }
39 //delete memory on the heap, columns then
40 //rows, set the array pointer to NULL
41 void delete_array(int** array, int rows) {
42     for(int i=0; i<rows; i++) {
43         delete [] array[i];
44     }
45     delete [] array;
```

22,1

50%

```
37     }
38 }
39 //delete memory on the heap, columns then
40 //rows, set the array pointer to NULL
41 void delete_array(int** array, int rows) {
42     for(int i=0; i<rows; i++) {
43         delete [] array[i];
44     }
45     delete [] array;
46     array = NULL;
47 }
48
49
50 int main() {
51     int rows=0, cols=0;
52     cout << "Rows: ";
53     cin >> rows;
54     cout << "Cols: ";
55     cin >> cols;
56
57     int** array = init_array(rows, cols);
58     print_array(array, rows, cols);
59     pop_array(array, rows, cols);
60     print_array(array, rows, cols);
```

40,1

85%

```
46     array = NULL;
47 }
48
49
50 int main() {
51     int rows=0, cols=0;
52     cout << "Rows: ";
53     cin >> rows;
54     cout << "Cols: ";
55     cin >> cols;
56
57     int** array = init_array(rows, cols);
58     print_array(array, rows, cols);
59     pop_array(array, rows, cols);
60     print_array(array, rows, cols);
61     mult_array(array, rows, cols);
62     print_array(array, rows, cols);
63     delete_array(array, rows);
64
65     return 0;
66 }
```

```
~
~
~
```

46,1-8

Bot



Type here to search

7:48 AM
3/1/2018