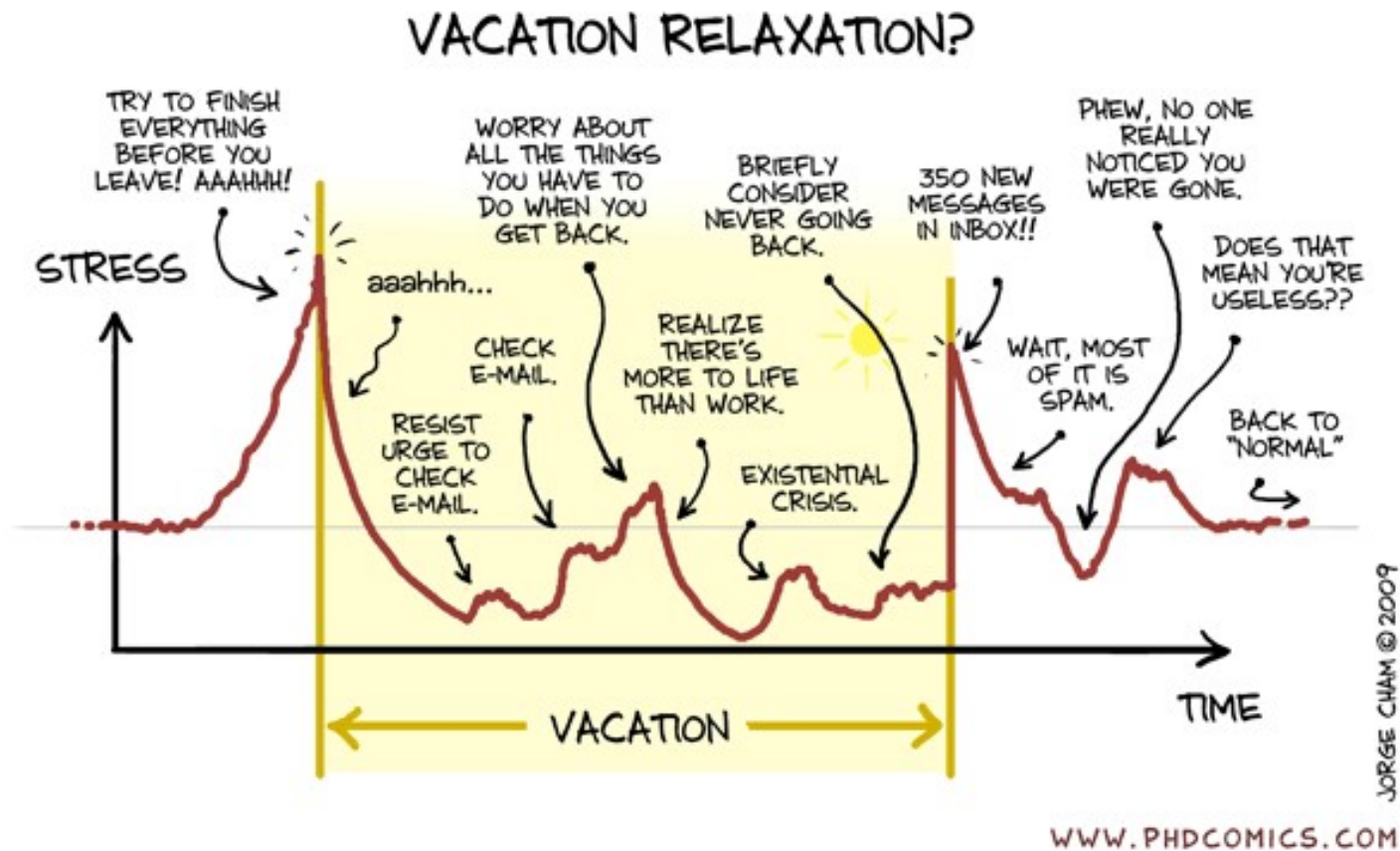


CS 161, Lecture 19: Multidimensional Arrays – 26 February 2018



Minor Correction

- Char arrays which are null terminated are C-style strings
 - That means you can use functions from `<cstring>` (`string.h`) on them
- Char arrays do not need to be null terminated
 - Just means you can't use `<cstring>` (`string.h`) on them
- We mostly default to C-style strings and expect null terminator on character arrays because most problems we solve involving characters are strings
- Two types of strings: counted (C++) and null terminated (C)

Multidimensional Arrays

- data_type array_name[rows][cols];
 - int mult_table[5][5];
 - char cross_word[15][15]
 - float grades[num_students][num_grades]
- ~~Example of multidimensional problem:~~
 - 2D Chess, Othello, Minesweeper
 - 3D Rubrics cube (x, y, z)
 - 4D x, y, z, t

Initializing 2D Arrays

- Declaration: `int arr[2][3] = {{0,0,0},{0,0,0}}`
cols
- Individual Elements:
 - `arr[0][0] = 0;`
 - `arr[0][1] = 0;`
 - `arr[0][2] = 0;`
 - `arr[1][0] = 0;`
 - `arr[1][1] = 0;`
 - `arr[1][2] = 0;`*rows*

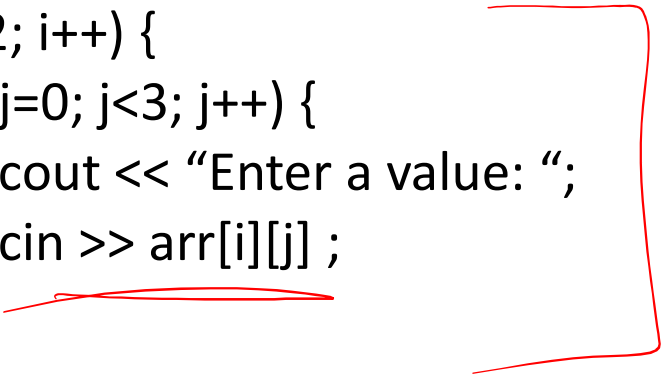
- Loop

```
for(int i=0; i<2; i++) {  
    for(int j=0; j<3; j++) {  
        arr[i][j] = 0;  
    }  
}
```

Reading and Printing 2D Arrays

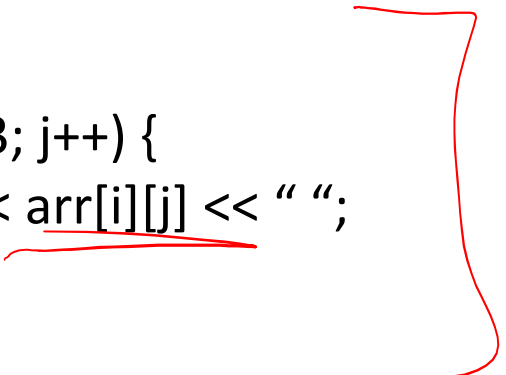
- Reading

```
for(int i=0; i<2; i++) {  
    for(int j=0; j<3; j++) {  
        cout << "Enter a value: ";  
        cin >> arr[i][j] ;  
    }  
}
```



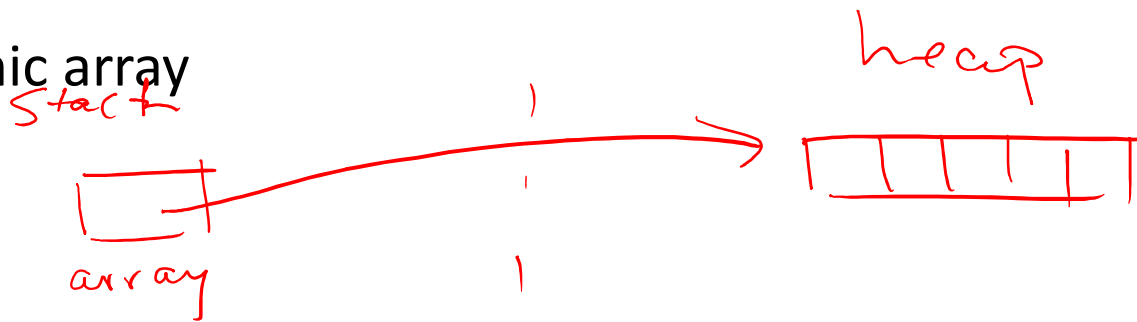
Printing:

```
for(int i=0; i<2; i++) {  
    for(int j=0; j<3; j++) {  
        cout << arr[i][j] << " ";  
    }  
    cout << endl;  
}
```

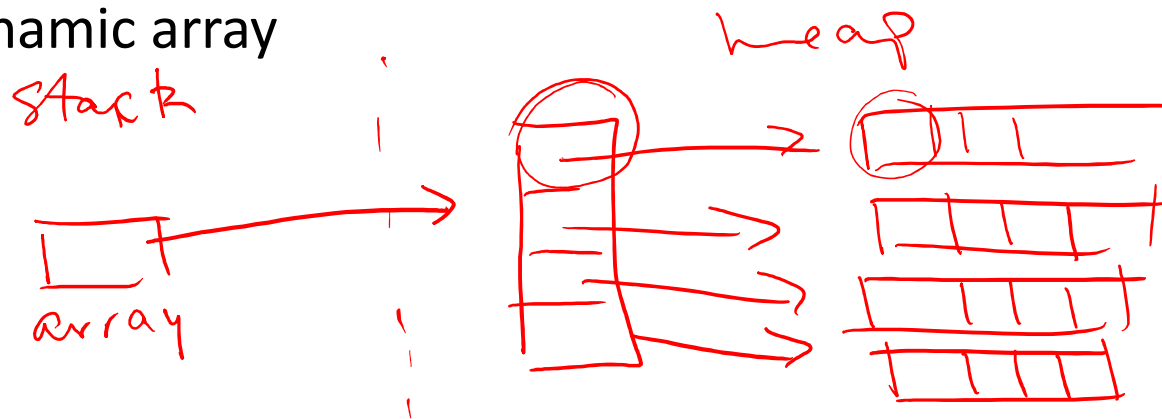


Dynamic 2D Arrays

- 1D dynamic array

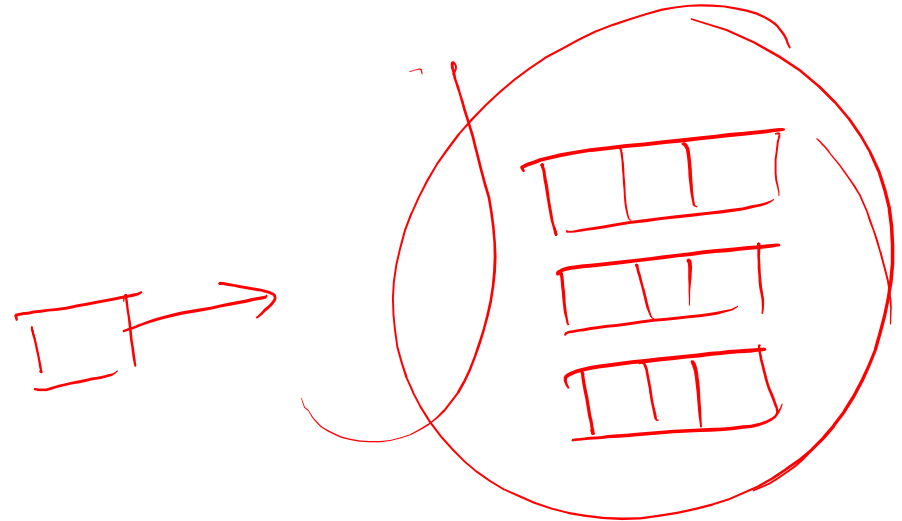


- 2D dynamic array



Declaration, Initialization and Deletion

```
int ** ar;  
ar = new int*[rows];  
for(int i=0; i<rows; i++) {  
    ar[i] = new int[cols];  
}
```



```
for(int i=0; i<rows; i++) {  
    delete [] ar[i];  
}  
delete [] ar;
```

Demo

```
access.engr.orst.edu - PuTTY
1 #include <iostream>
2
3 using namespace std;
4
5 void print_static(int ar[][5], int rows, int cols) {
6     for(int i=0; i<rows; i++) {
7         for(int j=0; j<cols; j++) {
8             cout << ar[i][j] << " ";
9         }
10        cout << endl;
11    }
12 }
13
14 void print_dyn(int** ar, int rows, int cols) {
15     for(int i=0; i<rows; i++) {
16         for(int j=0; j<cols; j++) {
17             cout << ar[i][j] << " ";
18         }
19        cout << endl;
20    }
21 }
22
23 void pop_static(int ar[][5], int rows, int cols) {
24     for(int i=0; i<rows; i++) {

```

1,1 Top

Type here to search 4:27 PM 2/26/2018


```
22 █
23 void pop_static(int ar[][5], int rows, int cols) {
24     for(int i=0; i<rows; i++) {
25         for(int j=0; j<cols; j++) {
26             ar[i][j] = 0;
27         }
28     }
29
30 }
31
32 void pop_dyn(int** ar, int rows, int cols) {
33     for(int i=0; i<rows; i++) {
34         for(int j=0; j<cols; j++) {
35             ar[i][j] = 0;
36         }
37     }
38
39 }
40
41 void init_dyn_array_void(int** ar, int rows, int cols) {
42     *ar = new int*[rows];
43     for(int i=0; i<rows; i++) {
44         (*ar)[i] = new int[cols];
45     }
```

22,0-1

29%

```
40 █
41 void init_dyn_array_void(int** ar, int rows, int cols) {
42     *ar = new int*[rows];
43     for(int i=0; i<rows; i++) {
44         (*ar)[i] = new int[cols];
45     }
46 }
47
48 int** init_dyn_array_return(int rows, int cols) {
49     int **ar = new int*[rows];
50     for(int i=0; i<rows; i++) {
51         ar[i] = new int[cols];
52     }
53     return ar;
54 }
55
56 void delete_dyn_array(int** array, int rows, int cols) {
57     for(int i=0; i<rows; i++) {
58         delete [] array[i];
59     }
60     delete [] array;
61     array = NULL;
62 }
63
```

40,0-1

54%

```
55
56 void delete_dyn_array(int** array, int rows, int cols) {
57     for(int i=0; i<rows; i++) {
58         delete [] array[i];
59     }
60     delete [] array;
61     array = NULL;
62 }
63
64 int main() {
65     /*int ar[5][5];
66     cout << "STATIC ARRAY" << endl;
67     cout << "Initial" << endl;
68     print_static(ar, 5, 5);
69     cout << "Populate" << endl;
70     pop_static(ar, 5, 5);
71     print_static(ar, 5, 5);*/
72
73     int** array;
74     cout << "DYNAMIC DECLARED IN MAIN, PASSED TO FUNCTION ALLOCATED I
75     N FUNCTION" << endl;
76     cout << "Initial" << endl;
77     init_dyn_array_void(&array, 5, 5);
78     print_dyn(array, 5, 5);
```

64,1

75%

```
73     int** array;
74     cout << "DYNAMIC DECLARED IN MAIN, PASSED TO FUNCTION ALLOCATED I
      N FUNCTION" << endl;
75     cout << "Initial" << endl;
76     init_dyn_array_void(&array, 5, 5);
77     print_dyn(array, 5, 5);
78     cout << "Populate" << endl;
79     pop_dyn(array, 5, 5);
80     print_dyn(array, 5, 5);
81     delete_dyn_array(array, 5, 5);
82
83     /*cout << "DYANMIC DECLARED IN MAIN, ALLOCATED IN FUNCTION, MEMOR
      Y RETURNED TO MAIN" << endl;
84     cout << "Initial" << endl;
85     array = init_dyn_array_return(5, 5);
86     print_dyn(array, 5, 5);
87     cout << "Populate" << endl;
88     pop_dyn(array, 5, 5);
89     print_dyn(array, 5, 5);
90     delete_dyn_array(array, 5, 5);*/
91
92
93
94     return 0;
```

73,1-8

98%