CS 161
Intro to CS I

Finish 1-D arrays/C-style string
Odds and Ends

• Last week to demo Assignment 4
• I’ll be out of town Wed./Fri.
• Exam II next Wed.
Class Exercise

• How do I initialize an array in a function?
• How can I print the contents of the array in a function?
• How about printing the address of the pointer to the array and the address of where the array begins in a function?

• How would I create an array in a function?
```cpp
using namespace std;

void init_arrays(int *array, int array_heap[]) {
    // initialize the elements in the arrays
    for(int i=0; i<10; i++) {
        array[i]=i;
        array_heap[i]=i+1;
    }
}

void print_pointer_addr(int (*array)[10], int **array_heap){
    // print where the pointer to the array lives in memory
    cout << "stack array pointer lives: " << array << endl;
    cout << "heap array pointer lives: " << array_heap << endl;
}

int main() {
    // create an integer array with 10 elements on stack and one on the heap
    int array[10]={3}; // does not initialize all elements to 3, only 1st
    int array1[]={1,2,3,4,5,6,7,8,9,10};
    int *array_heap;
    array_heap=new int[10];
    init_arrays(array, array_heap);
    print_pointer_addr(&array, &array_heap);
```
```c++
10 }
11 void print_addr(int **array, int **array_heap) {
12     //print where the pointer to the array lives in memory
13     cout << "where pointer to stack array lives: " << array << endl;
14     cout << "where pointer to heap array lives: " << array_heap << endl;
15 }
16 int main() {
17     //create an integer array with 10 elements on stack and one on the heap
18     int array[10]={3};  //this doesn't initialize all elements
19     int array1[]={1,2,3,4,5,6,7,8,9,10};
20     int *array_heap=NULL;
21     array_heap=new int[10];
22     array_init(array, array_heap);
23     print_addr(&array, &array_heap);
24     //print where the array begins in memory
```
void print_addr(int **array, int **array_heap) {
    // print where the pointer to the array lives in memory
    cout << "where pointer to stack array lives: " << array << endl;
    cout << "where pointer to heap array lives: " << array_heap << endl;
}

int main() {

    // create an integer array with 10 elements on stack and one on the heap
    int array[10]={3};  // this doesn't initialize all elements
    int array1[]={1,2,3,4,5,6,7,8,9,10};
    int *array_heap=NULL;
    array_heap=new int[10];
    int **a=&array;

    array_init(array, array_heap);
    print_addr(&array, &array_heap);
    // print where the array begins in memory
```cpp
#include <iostream>

using namespace std;

void array_init(int *array, int array_heap[]) {
    // initialize the elements in the arrays
    for(int i=0; i<10; i++){
        array[i]=i;
        array_heap[i]=i+1;
    }
}

void print_addr(int (*array)[10], int **array_heap) {
    // print where the pointer to the array lives in memory
    cout << "where pointer to stack array lives: " << array << endl;
    cout << "where pointer to heap array lives: " << array_heap << endl;
}

void array_begins(int *array, int *array_heap) {
    // print where the array begins in memory
    cout << "where stack array begins: " << &array[0] << endl;
    cout << "where stack array begins: " << array << endl;
    cout << "where heap array begins: " << &array_heap[0] << endl;
    cout << "where heap array begins: " << array_heap << endl;
    cout << "3rd element stack: " << array[2] << endl;
    cout << "3rd element stack: " << *(array+2) << endl;
    cout << "3rd element heap: " << array_heap[2] << endl;
    cout << "3rd element heap: " << *(array_heap+2) << endl;
}
```
void fun(int **a) {
    //int array1[]={1,2,3,4,5,6,7,8,9,10};
    *a=new int[10];
}

int main() {
    //create an integer array with 10 elements on stack and one on the heap
    int array[10]={3}; //this doesn't initialize all elements
    int *array_heap=NULL;
    //array_heap=new int[10]; //we are going to create array in a function

    //can't do this because array is a constant pointer to a static array
    //of 10 elements
    //int **a=&array;

    fun(&array_heap);
    cout << array_heap[2] << endl;

    array_init(array, array_heap);
    print_addr(&array, &array_heap);
    array_begins(array, array_heap);

    //print the contents of the 3rd element
    delete [] array_heap;
    return 0;
}
What are C-style strings?

- Ended by `\0` character
- Need to include `<cstring>`
```c++
#include <iostream>
#include <cstring>
using namespace std;

int main() {
    char sentence[10]; // make it big

    cout << "Enter a string: ";
    cin.getline(sentence,100); // don't ever make this bigger than the array

    cout << sentence << endl;

    cout << "Enter a string: ";
    cin.getline(sentence,10); // read until seeing a newline or 9 characters to leave room for the null character '\0'

    cout << sentence << endl;

    return 0;
}
```