CS 161
Intro to CS I

2-D arrays & Static vs. Dynamic
Chap. 5 and 10
#include <iostream>
using namespace std;

int main() {
    int *p;
    int j=10, c[3];
    p=&j;  //Make p point to j

    //Nothing stops you from going a little bit
    //outside your array, but you cannot go too far
    cout << &c[3] << endl;
    cout << c[3000] << endl;
    cout << &p << endl;  //Address of p
    cout << p << endl;  //Contents of p, which is j's address
    cout << *p << endl;  //Dereference p, which is j's content
    cout << &j << endl;  //Address of j
    cout << j << endl;  //Contents of j

    return 0;
}
What if we don’t have the $y$?

• We need to create the address space.

• How do we do this?
  
  – `new` type;

• For example:
  
  ```
  int *x;
  x = new int;  //new returns an address
  *x = 10;
  ```

• [http://cslibrary.stanford.edu/104/](http://cslibrary.stanford.edu/104/)
Static vs. Dynamic 2-D arrays...
Jagged Arrays

int *array[2];
array[0] = new int[3];
array[1] = new int[2];

• Can we make the columns jagged?
Make sure you free your memory!

- Free one memory location created
  delete a;
- Free the row pointers
  for(i=0;i<rows; i++)
    delete [] arr2[i];
- Free the pointer to rows
  delete [] arr2;
How does freeing memory work?

```c
int *p, *q;
p = new int;
q = new int[5];
delete p;
delete [] q;
```
Passing a 1-D Array (Static/Dynamic)

```c
int main() {
    int array[5];
    ...
    pass_1darray(array);
    ...
}
void pass_1darray(int *a) {
    cout << "Array at zero: " << a[0] << endl;
}
OR
void pass_1darray(int a[]) {
    cout << "Array at zero: " << a[0] << endl;
}
```
Passing a 2-D Array (Static)

```c
int main() {
    int array[5][5];
    ...
    pass_2darray(array);
    ...
}
void pass_2darray(int a[5][5]) {
    cout << "Array at zero: " << a[0][0] << endl;
}
OR
void pass_2darray(int a[][5]) {
    cout << "Array at zero: " << a[0][0] << endl;
}
```
Passing a 2-D Array (Dynamic)

```c
int main() {
    int *array[2];
    ...
    pass_2darray(array);
    ...
}
void pass_2darray(int *a[]) {
    cout << “Array at zero: ” << a[0][0] << endl;
}
OR
void pass_2darray(int **a) {
    cout << “Array at zero: ” << a[0][0] << endl;
}
```
Passing a 2-D Array (Dynamic)

```c
int main() {
    int **array;
    ...
    pass_2darray(array);
    ...
}
void pass_2darray(int *a[]) {
    cout << "Array at zero: " << a[0][0] << endl;
}
OR
void pass_2darray(int **a) {
    cout << "Array at zero: " << a[0][0] << endl;
}
```
Quiz #5

• Assignment #4:
  – How will you reverse a string?
  – How will you ignore case or special characters in your sentence?
  – How will you determine if a sentence is a palindrome?