Pointers and Strings

What is the Output?

```
void some( int * pNums );
int main()
{
  int nums[] = {2, 4, 6, 8};
  some( nums );

  return 0;
}
void some( int * pNums )
{
  cout << "*pNums: " << *pNums << endl;
  cout << "*(pNums + 1): " << *(pNums + 1) << endl;
  cout << "*pNums++: " << *pNums++ << endl;
  cout << "*(pNums++): " << *(pNums++) << endl;
  cout << "(*pNums)++: " << (*pNums)++ << endl;
}
```
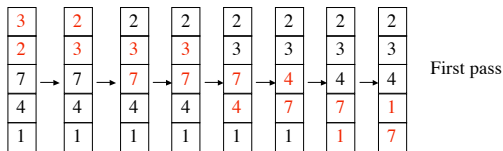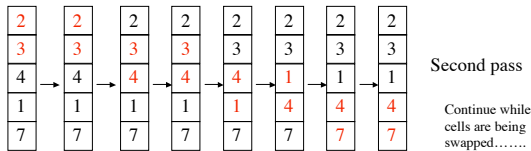

Bubble Sort

First pass

Second pass

Continue while cells are being swapped…….

Bubble Sort

- Let's look through the code that will perform the bubble sort described on the previous slide

- http://zeus.cs.pacificu.edu/shereen/cs250/lectures/04bubble.html

---

Arrays of Pointers

- What do you make of the following declaration?

```
char *cardSuits[4] = {"Clubs", "Diamonds",
                      "Hearts", "Spades"};
```

- What gets output in each of the following cases?

```
cout << cardSuits[1] << endl;

cout << *cardSuits[1] << endl;
```

Arrays of Pointers

- Let's look at an example where arrays of pointers are used

  o http://zeus.cs.pacificu.edu/shereen/cs250/Lectures/05card.cpp

- This example also serves as a reminder of how to generate random numbers

String Functions

- Let's take a look at the string functions on pp. 363

Review of structs

```
struct Person
{
  char nameStr[20];
  char ssNum[9];
  int age;
};
```

- What do each of the following declarations mean?

```
Person personStruct;
Person personArry[5];
Person *pPerson = &personStruct;
```

- Let's examine Fig. 6.1 on pp. 408-409

Summary

- Today I introduced
  - Bubble sort
  - Arrays of pointers
  - Review of structures
- We have covered:
  - All of chapter 5