CS 271 - Sept 12, 2017

makefile

The name of the file is makefile (no extension). When you submit a makefile on Canvas, be absolutely certain that there is no extention.

Insertion Sort

loop from element 1 to the end of the array

for each element, loop to move the element to the left until it is in the correct position. Don't run off the left edge of the array.

Example array

```
    3     8      10      6     4     9

    3     8      10      6     4     9

    3     8      10      6     4     9

    3     8      10      6     4     9
       exchange
    3     8       6      10    4     9
       exchange
    3     6       8      10    4     9

    3     6       8      10    4     9
      move to the left until it's in the correct position
    3     4       6       8    10     9

     3     4       6       8    10     9

     move to the left
     3     4       6       8     9    10
```

Selection Sort

Start with element 0:

Find the smallest value from array[0] to array[size-1]

Swap the smallest with array[0].

Move to element 1:

Find the smallest from array[1] to array[size-1]
Swap smallest with array[1].

Continue up to element size - 2

```
3     8      10      6     4     9

3     8      10      6     4     9
   swap array[4] with array[1]
```

```
3     4      10      6     8     9
   swap array[2] with array[3]

3     4        6     10    8    9

swap array[3] with array[4]

3      4      6      8     10    9

swap array[4] with array[5]

finished
3     4       6     8      9     10
```

Selection Sort and Insertion Sort can work with arrays of characters or other data types. As long as you are able to compare one element to another and decide which is less than or greater than.

Random numbers in C.

rand function
- no parameters
- returns a number between 0 and MAXINT
- this is the number of seconds since Jan 1, 1970

Take the value of rand and use % to scale it down to the range you need.

```
range = max - min + 1
```

Then add the minimum value to shift the numbers over.

Examples: generate a random number between 80 and 90 and store it in z.

```
int z = rand( ) % 11 + 80;  // 11 is the range, 80 is the minimum
```

Generate a random number between -35 and -22.

```
int x = rand( ) % (-22 - -35 + 1) + -35;
```

The rand function always starts at the same place.

A way to set the starting point is to use srand. The starting point is called the "seed".

```
srand(36);  // sets the seed to 36

….
int x = rand() % 11 + 80;
```

To set a different seed every time you run the program:

```
srand ( time(NULL) );
```

Dynamic memory allocation - allocating space during runtime

malloc function

```
void *malloc(size_t size)
```

size_t is an unsigned integer

The return type is a void pointer.

Void pointer is a pointer (memory address) with no specified data type that it is pointing to.

```
int * arrayPtr = malloc ( 10 * sizeof( int ) );  // works

int * arrayPtr = (int *) malloc ( 10 * sizeof( int ));  // better style
```

Work with the pointer to access array elements.

NOTE: There was a very important handout given out in class. It contains two example programs illustrating how to work with array elements by using pointer arithmetic.

Extra copies on the bulletin board at SH 157.