No handouts today

Day 9

Coding example (selection sort)

Scope

Finish up structs

Admin

- HW1 is being returned this Tues/Wed in lab.
- Quiz 1 is graded (but not recorded) and should be returned this Thur/Friday in lab.
- P0 is graded and recorded. Scores will be posted or e-mailed by noon tommorow.
  - Some of you got e-mails about problems with handing in P0. Please see your GSI for help.

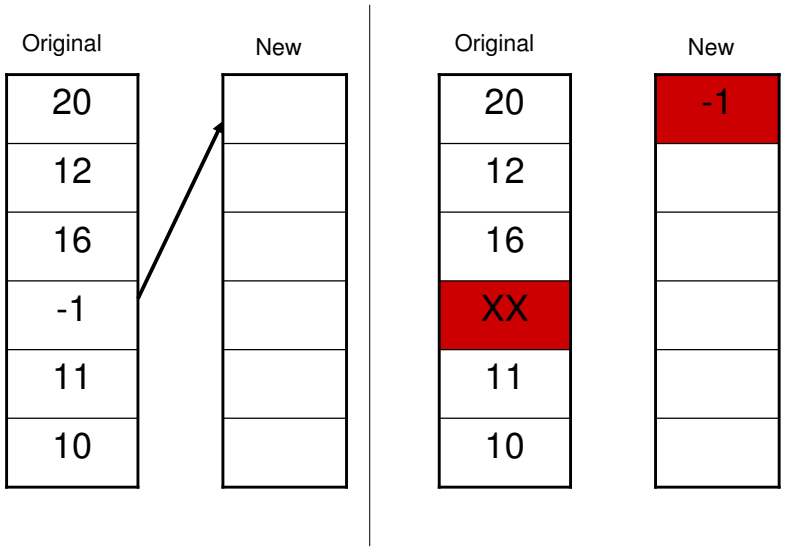
Last time

- "How to code"
  - Chose an algorithm called "selection sort"
  - Discussed steps of coding
    1. Understand algorithm
    2. Try to get a high-level understanding of algorithm to code (what data is needed, what looks hard to code, etc.)
    3. Write pseudo-code
    4. Code in C++
- It is not uncommon to be able to skip some steps. But if you are lost/stuck/confused, this at least gives plan.

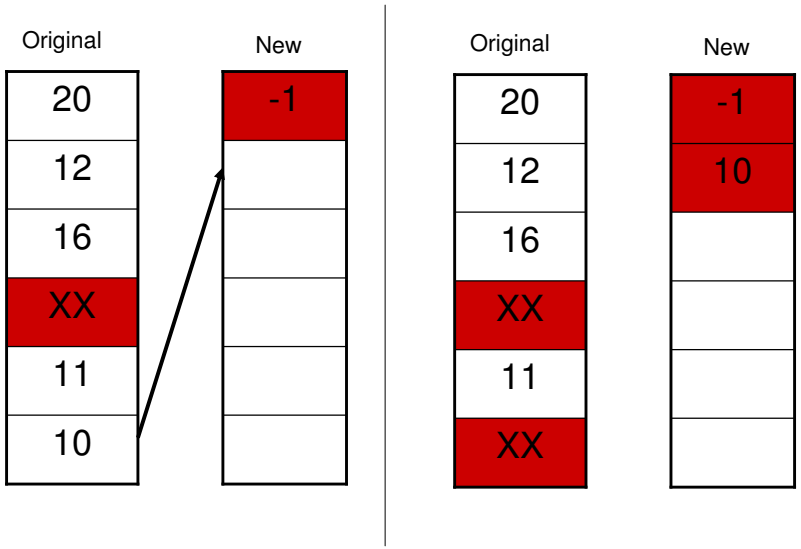
Algorithm

- Go through each element of an array.
  - Find the smallest element
  - Copy it into the top of a new array.
  - Remove it from the original array.
- Now do the same as above over the original array
  - But keep putting things into the next open space.

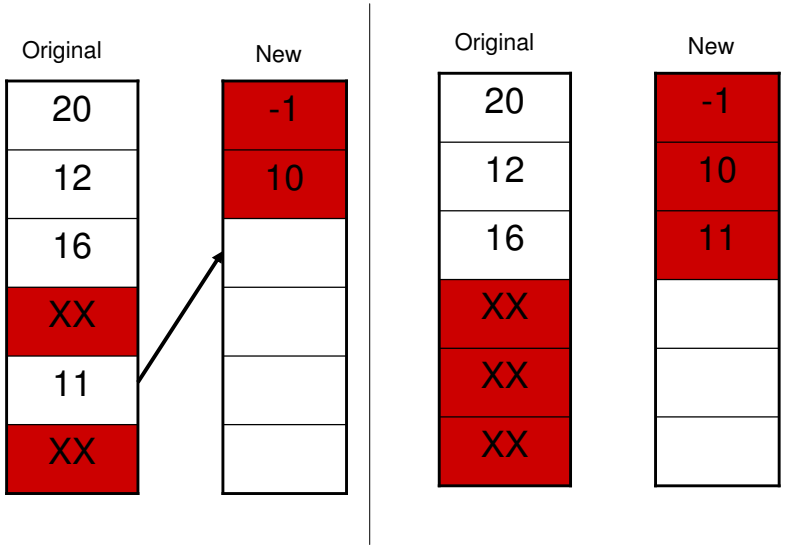
Selection sort



Selection sort



Selection sort



Back to the board for code

Scope

- General theme:
  - Variables are only “visible” in the function (including main) in which they are declared.
- Ramifications
  - I can have two variables in different functions with the same name.
    - **They do not conflict**.
  - If you want to share information between functions, you need to pass it as an argument/parameter or as a return value.
  - In the debugger “out-of-scope” variables are not displayed.

Scope

- Globals
  - You can declare a variable to have global scope.
  - All functions (at least all in the same file) can use it.
- **Do not use globals**
  - One exception: global constants can be acceptable.
- To declare a global, just place it at the top of the file, outside of any function.

Example from “wrong.cc” of P0

```
#include<iostream>
using namespace std;

const int STEPS=8;

double my_intpower(double value, int power)
{

-- more code goes here --
```

Bonus slides

- Time allowing will touch on structs again.

```
#include<iostream>
using namespace std;

struct complex
{
        double real;
        double img;
};
complex c_add(complex a, complex b)
{
        complex result;

        result.real=a.real+b.real;
        result.img=a.img+b.img;
        return(result);
}

complex c_print(complex a)
{
        cout << "(" <<a.real << " + " << a.img << "i) ";
}
```

Page 1 of 2

```
main()
{
        complex x ={1.0,3.0};
        complex y ={-3.0, -1.0};
        complex z;

        z=c_add(x,y);

        c_print(x);
        cout << " + ";
        c_print(y);
        cout << " = ";
        c_print(z);
        cout << endl;
}
```

Page 2 of 2

```
main()
{
        const int SIZE=4;
        complex x[SIZE];
        int i=0;

        while(i<SIZE)
        {
                x[i].real=i;
                x[i].img=i*(-i);
                i=i+1;
        }
        i=0;
        while(i<SIZE)
        {
                c_print(x[i]);
                cout << endl;
                i=i+1;
        }
}
```

Different main

Coming up

- Lecture:
  - Friday:
    - Multi-dimensional arrays
  - Monday
    - Strings
  - Wednesday or Friday
    - File input/output
- Assignments
  - P2 will come out tomorrow, due a week from Friday.