2020/08/26 - The Warmup Lab

2020年08月24日 月曜日　午後 02:54

SYNOPSIS

- Welcome to CS140!

- TAs will introduce themselves as well as discuss office hours, lab submission procedure, take questions, etc.

- Go over LAB 1.

LAB 1

- On Canvas, go to the "Lab 1.1" assignment and read the "lab1.html" file. All lab details will be there.

- There are 4 parts. There are 2 due dates:
  - Sept 1: Prog1b, Prog1c　　(STATISTICS ON NUMBERS)
  - Sept 8: Prog1d.　　(BASIC CHARACTER MANIPULATION)

- Prog1a is NOT submitted.

SUBMISSION COMMANDS

Lab 1.1
```
tar -cvf lab1_1.tar Prog1b.cpp Prog1c.cpp
```

Lab 1.2
```
tar -cvf lab1_2.tar Prog1d.cpp
```

Prog1B + Prog1C

- Nothing complex.

- Read integers via cin.

- Do not store in an array or vector. Instead, adjust variables upon reading each in.

- Prog1B is in main.

- Prog1C has the following class prototype:

```
class stats {
    public:
        stats();
        void push(int);
        void print() const;

    private:
        int N, sum, min, max;
}
```

Implement stats::stats, stats::push, and stats::print.

Prog1D

- Remember passing arguments into your program.
  - int main(int argc, char ** argv)

    ARGUMENT   ARGUMENT
    COUNT   VECTOR

  - ./Prog1d -encode 4

    argc = 3
    argv[0]: "./Prog1d"
    argv[1]: "-encode"
    argv[2]: "4"

  - Check if argc is correct. Print an error to stderr via cerr if not.

- A char is often a letter or number. But really, they are bytes of numerical value. As such, they can be manipulated like ints.

    (Ex. char c = 'a';
           c++; //haha
           cout << c << endl;

              ⇓

        Prints 'b'.

- In fact, ALL letters have an integer association:

LETTER	A	B	C	D	E	...	V	W	X	Y	Z
HEX	41	42	43	44	45	...	56	57	58	59	5A

- Two functions:
    1. `void encode ( string &input, int shift );`
    2. `void decode ( string &input, int shift );`

- ENCODE
    - Shift all letters in "input" to the RIGHT by "shift" amount.
        (Ex. `encode("abc", 1);`
            "abc" ⟹ "bcd"
            Each letter shifted once to the right.

        `encode("abc", 2);`
        "abc" ⟹ "cde"
        Each letter shifted twice to the right.

        `encode("xyz", 1);`
        "xyz" ⟹ "yza"
        Each letter shifted once to the right.
        "z" wraps around and becomes "a".

- DECODE
    - Same as ENCODE, except shift LEFT instead.

- Use `isupper` and `islower` to check if you can shift the characters or not.
    - `#include <ctype.h>` will give you these functions.