Homework 1. . .

Due date: Tuesday, October 14, beginning of the class.

Submission Instructions

The assignment has three parts: (i) the programming part, extending on your Lab 3 work, (ii) the lab exam practice part, which comes with special instructions, and (iii) the paper-and-pencil part.

- The paper-and-pencil part of the assignment should be detached from this handout, completed and submitted before the end of the Tuesday, October 14 class.
- The programming part and the lab exam practice part should be submitted using the following `handin` commands:
  - Section 09:

    `> handin dekhtyar hw01-09 <file1>,...,<fileN>`
  - Section 11:

    `> handin dekhtyar hw01-09 <file1>,...,<fileN>`

The list of files to submit is supplied at the end of each assignment section.

Assignment Preparation

Collaboration. This is an **individual assignment**, no collaboration is permitted on any part of it. **It is especially important that you follow the lab exam practice instructions!** Failure to follow them increases the chances of failing the lab exam itself.

Programming Style. All submitted C programs for the programming part must adhere to the programming style described in detail at

http://users.csc.calpoly.edu/~cstaley/General/CStyle.htm

When graded, the programs will be checked for style. Any stylistic violations are subject to a 10% penalty.

Programming style will not be enforced for the lab exam practice assignment.

Testing and Submissions. Any submission that does not compile using the

```
gcc -ansi -Wall -Werror -lm
```

compiler settings will receive an automatic score of 0.

Programming Assignment.

You will continue working with if-statement zones. Please re-read the instructions from Lab 3 regarding the basic requirements for if-statement zones programs. **These requirements remain in force for the programming assignment**.

As before, we specify the number of `if` statements, `&&` and || operators that you may use in your program.

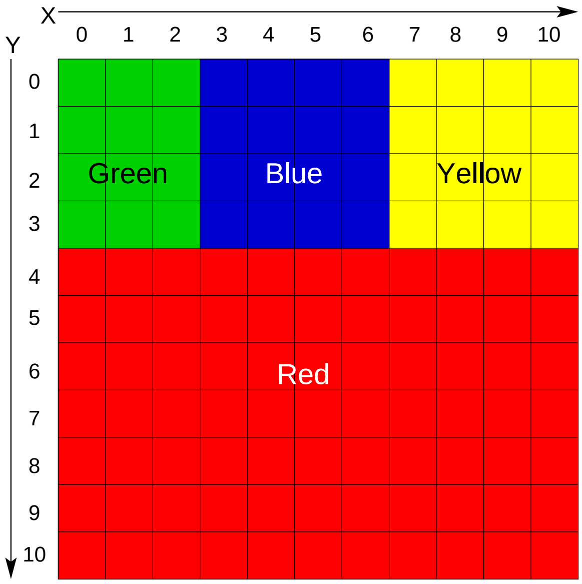
Files to submit. For the programming part of the assignment you shall submit **five files:** files:

`zones06.c`,`zones07.c`, `zones08.c`, `zones09.c`, `zones09.c`.

Program 1: zones06.c

Restrictions:

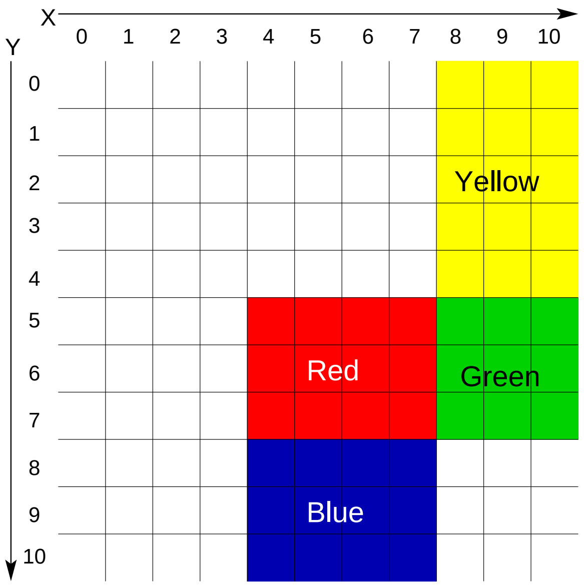
Number of `if`s:	3
Number of `&&` operators:	0
Number of `||` operators:	0



Program 2: zones07.c

Restrictions:

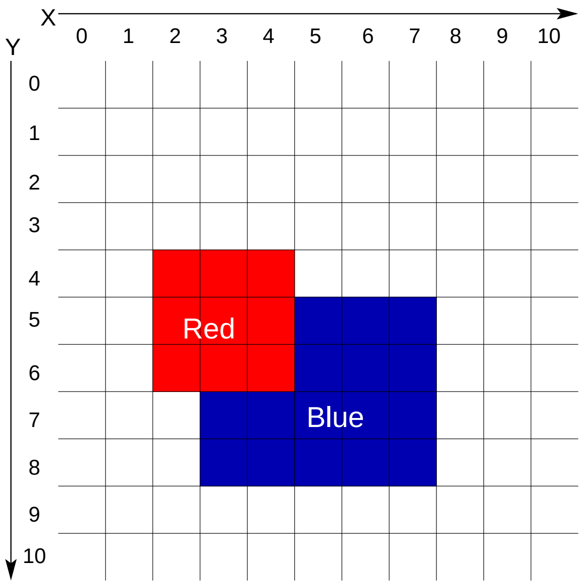
Number of `ifs`:	6
Number of `&&` operators:	0
Number of `||` operators:	0



Program 3: zones08.c

Restrictions:

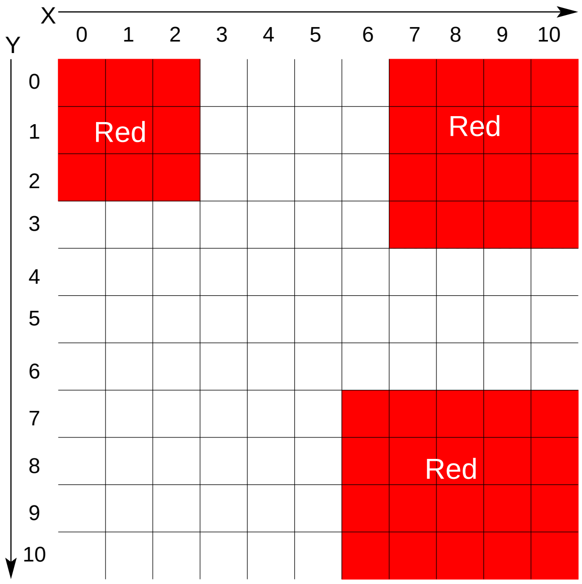
Number of `ifs`:	8
Number of `&&` operators:	0
Number of `||` operators:	0



Program 4: zones09.c

Restrictions:

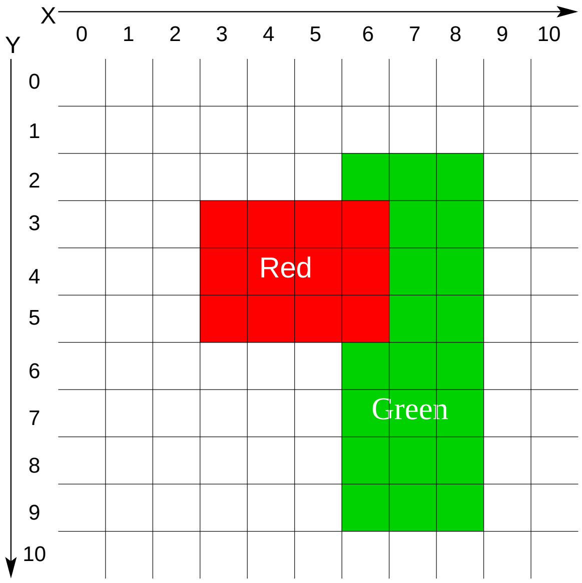
Number of `ifs`:	1
Number of `&&` operators:	*any*
Number of `||` operators:	*any*



Program 5: zones10.c

Restrictions:

Number of `ifs`:	2
Number of `&&` operators:	*any*
Number of `||` operators:	0



Lab Exam Preparation

The following problem gives you an opportunity to practice for the lab exam. You should work on the problem following the instructions provided to you below. Read the instructions carefully before starting the work.

Instructions

- Time to complete the exam: 50 minutes.
- This exam is individual
- This exam is open book
- This exam is closed everything else, including lecture notes, electronic devices, etc
- The only programs you are allowed to have open on the computer during this exam are:
  - The text editor of your choice
  - Terminal window(s)
  - ssh connection to vogon
- Do not use any code you may have access to from earlier in this course or from other courses
- Before starting your work, open a terminal session, and create a directory for the practice exam. It will be referred to as `practice-labexam` below.
- The problem description includes all assumptions necessary to answer the problem.
- GOOD LUCK!

Assignment

Investors like tracking the contents and value of their stock portfolios. You must write a program that assists investors by computing the total value of the portfolio and determining the most valuable stock in the portfolio. The input to your program will be a series of three stocks, each on their own line. The first number in each line is the number of shares held in the portfolio, in whole numbers. The second number in the line is the current market value of one share, in dollars.

Your program must output a line containing the total value of the portfolio, followed by a line indicating the most valuable holding in the portfolio. Stocks are numbered according to their order in the input. Stock 1 is the first stock in the input, 2 the second, and 3 the third. Your program must match the wording in the example output exactly, including capitalization and whitespace.

You must name your source code `portfolio.c` and your compiled executable `portfolio`. Your source code must compile with the following compiler flags: `-ansi -pedantic -Wall -Werror`. Failure to compile with these flags results in failure on the lab exam. Style will **not** be enforced on this exam.

Assume that all input is valid. All input is to be read from standard in and all output is to be printed to standard out.

Example Input:

```
10 12.89
1 108.12
5 38.50
```

Output for example Input:

```
The portfolio is worth $429.52.
Stock 3 is the most valuable holding in the portfolio.
```

Testing your solution

You have access to my full set of test cases for this program, my executable of it and scripts to test my executable and your executable. Copy all the files to your `practice-labexam` directory with the following command:

```
cp ~dekhtyar/www/practice-labexam/* .
```

Instructor's executable is called `alex-protfolio`. Do `ls -al` to ensure that it is executable (if not, run `chmod u+x alex-portfolio`. Perform the same check for `portfolio-test.csh` and `alex-portfolio-test.csh` files that are also copied. Test file names are `portfolio-test01` through `portfolio-test10`.

To test your program first compile it into an executable named `portfolio`. Then run the following command:

```
> portfolio-test.csh
```

The script will run your program on all test cases and produce output. To check it against instructor's output, open a second terminal session, change to `practice-labexam` directory and run the command:

```
> alex-portfolio-test.csh
```

Compare the two outputs.

Submitting your solution

Once you are satisfied your program meets all the requirements you may turn it in and leave the exam. Run the following command on vogon to submit your program:

```
handin dekhtyar  hw01-09 portfolio.c
```

or

```
handin dekhtyar  hw01-11 portfolio.c
```

depending on which section you are in.

Paper-and-Pencil Part

Instructions. Detach the paper-and-pencil part, do all the work in the spaces alotted for it, and submit on the due date.

Problem 1. Identifiers

In the list below, circle all C identifiers (i.e., valid variable names). (note, some variable names are *valid*, but prohibited by our style guide. They should be circled.)

`a1`	`ThisIsAnInt`	`Myspace.com`	`12Months`	`_X_`
`Flash_Drive`	`Num_Vars`	`INWARDS`	`HOW_MUCH?`	`if20`
`true`	`if`	`iff`	`a1a2a3a4`	`float`
`me@gmail`	`don't_like_it`	`not_my_fav_o_rit_e`	`__Robots`	`stone-cold`

Problem 2. Constants

For each constant below, specify its type. If the constant is invalid, say `"invalid"`.

(a) `-357` ________	(b) `929,567` ________	(c) `-2.001` __________
(d) `'c'` ________	(e) `800e-3` ________	(f) `4.2.2` __________
(g) `0` ________	(h) `'Alex'` ________	(i) `true` __________
(j) `23,01.14` ______	(k) `5.2e12` ________	(l) `'\n'` __________
(m) `3.4z3` ________	(n) `"a"` ________	(c) `4.2e2.4` __________

Problem 3. Expressions

Rewrite each C expression using parenthesis to show the order of operations.
(e.g. `a+b-c` is `(a+b)-c`.)

(a) `45 - x * 4 / 7 + 3 + 4 * x` ____________________________________

(b) `b + - 23 - -14 * 2 -1` ____________________________________

(c) `c*f-2 == 3 + 2/d-1` ____________________________________

(d) `a == 4 && b == 3 * -34 * - x -2` ____________________________________

(e) `a + b*--c - b / - a == -a - -b - -c/- d + a || b + 14- -2*y <= 12 % r + k`

____

Problem 4. Assignment

Consider the following code fragment:

```
x = x + -y/3 + z%7;
y = x + y;
z = z + y;
```

All variables are declared as `int`s above the fragment. For each set of initial (i.e., before the code fragment is executed) variable assignments below, specify the values of `x, y` and `z` after the code fragment executes.

```
(a) Initial:   x: 4      y: 3     z: 8

    Final:     x: ___    y: ___   z: ___

(b) Initial:   x: 100    y: 2     z: 17

    Final:     x: ___    y: ___   z: ___

(b) Initial:   x: -12    y: -5     z: 100

    Final:     x: ___    y: ___    z: ___
```

Problem 5. Comparisons and logical expressions

Consider the following assignments (all variables are `int` and declared):

```
x = 2;
y = x*2;
z = x*y;
```

For each expression below, specify what it evaluates to.

```
(a) x >= y%x                              evaluates to _________

(a) z + x*x <= y*y                        evaluates to _________

(b) y >= x && y <= z                      evaluates to _________

(c) !(y+x >= z/2)                         evaluates to _________

(d) !(y-x == z/y) || (y > x)              evaluates to _________

(e) (x*x*x == z) && !(y == x)             evaluates to _________

(f) (z-x <= y + z) || ! (z-x <= y+z)      evaluates to _________

(g) !(x>z) && !(z<y) && !(y<x)            evaluates to _________

(h) (z+z > x*x*y) || !(x==y)              evaluates to _________

(i) (x - x) && (y*y*y*y*y >= x*z*x*z)     evaluates to _________

(j) (z*x - z) || (y - x*2)                evaluates to _________
```