Homework 4 C Programming - A few functions

Due Tuesday 9/17/2019 11pm

This assignment you will define a few functions that can complete certain tasks.

Download hw4handout.tar

Use instructions similar to homework 2 to download hw4handout.tar file, copy it to w204 machine's 311 folder, use `tar xvf` command to unpack it to hw4handout folder.

Functions to implement

You are to implement three functions, see comments about what each function should do.

```
// rotate the values pointed to by three pointers
// so values move from xp to yp, yp to zp and zp to xp
void rotate(int *xp, int *yp, int *zp) {
  return;
}

// Write a function that returns 0 if x is 0, returns -1
// if x < 0, returns 1 if x > 0
// Your code must follow the Bit-Level Integer Coding Rules
// on the textbook (located between hw 2.60 and 2.61).
// You can assume w = 32.
// The only allowed operations in your code are:
// ! ~ & ^ | + << >>
// This requirement is more restrictive than the coding rules.

int sign(int x) {
  return 0;
}

// Given a source string, and a destination string, write the
// reversed source string into destination. Make sure you terminate
// the reversed string properly with an integer value 0.
// You may assume that the destination array is big enough to hold
// the reversed string.
void reverse(const char source[], char destination[]) {
  return;
}
```

Edit/Compile/Test your C code

1. Compile your code

Assuming you successfully copied and unpacked hw3handout.tar file in ~/311 folder on your W204 account, the following command will help you compile the given C program. The original tar file contains a complete C program that compiles and works.

```
cse-p204inst11.cse.psu.edu 160% cd ~/311/hw4handout
```

You use make command in the hw4handout folder to compile the hw4.c file to create an executable called hw4.

Make sure that your make does not generate any error/warning messages while compiling your C code. Any warning/error messages while compiling your code will result in a zero for this assignment. For example, the following make command shows some warnings. You must address these warnings and remove all compiler warnings/error messages before you submit.

2. Test your code

2.1 Testing with supplied Driver

You start with code that contains functions that don't really do anything. If you test your code by executing hw4, it will fail most of the tests we provided in the driver. Edit your code so that it can pass the tests we provided.

```
cse-p204inst31.cse.psu.edu 134% cd 311
cse-p204inst31.cse.psu.edu 135% tar xvf hw4handout.tar
hw4handout/
hw4handout/hw4.c
hw4handout/Makefile
hw4handout/hw4.h
hw4handout/driver.c
cse-p204inst31.cse.psu.edu 136% cd hw4handout
cse-p204inst31.cse.psu.edu 137% ls
driver.c  hw4.c  hw4.h  Makefile
cse-p204inst31.cse.psu.edu 138% make
gcc -Wall -Og -g -o hw4 hw4.c driver.c
cse-p204inst31.cse.psu.edu 139% ./hw4
Testing rotate:
Before rotate: 1, 2, 3
After rotate: 1, 2, 3
----fail----
Before rotate: -100, 0, 100
After rotate: -100, 0, 100
```

```
----fail----
Before rotate: 1, 1, 1
After rotate: 1, 1, 1
----pass----
====================
Testing reverse:
source: "Welcome to CMPSC 311!"
destination: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
----fail----
source: "abcdef"
destination: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
----fail----
source: "_ ab  cd  ef *"
destination: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
----fail----
source: ""
destination: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
----fail----
====================
Testing sign:
x = 0
sign = 0
----pass----
x = -1
sign = 0
----fail----
x = 1
sign = 0
----fail----
x = -2147483648
sign = 0
----fail----
x = 2147483647
sign = 0
----fail----
x = 65535
sign = 0
----fail----
x = -65536
sign = 0
----fail----
x = -2
sign = 0
----fail----
x = 3
sign = 0
----fail----
cse-p204inst31.cse.psu.edu 140% vim hw4.c
```

```
cse-p204inst31.cse.psu.edu 140% make
gcc -Wall -Og -g -o hw4 hw4.c driver.c
cse-p204inst31.cse.psu.edu 140%./hw4
Testing rotate:
Before rotate: 1, 2, 3
After rotate: 3, 1, 2
----pass----
Before rotate: -100, 0, 100
After rotate: 100, -100, 0
----pass----
Before rotate: 1, 1, 1
After rotate: 1, 1, 1
----pass----
====================
Testing reverse:
source: "Welcome to CMPSC 311!"
destination: "!113 CSPMC ot emocleW"
----pass----
source: "abcdef"
destination: "fedcba"
----pass----
source: "_ ab  cd  ef *"
destination: "* fe  dc  ba _"
----pass----
source: ""
destination: ""
----pass----
====================
Testing sign:
x = 0
sign = 0
----pass----
x = -1
sign = -1
----pass----
x = 1
sign = 1
----pass----
x = -2147483648
sign = -1
----pass----
x = 2147483647
sign = 1
----pass----
x = 65535
sign = 1
----pass----
x = -65536
```

```
sign = -1
----pass----
x = -2
sign = -1
----pass----
x = 3
sign = 1
----pass----
```

2.2 Testing with your own tests examples

If you managed to pass all the tests provided in driver.c file, we recommend you to try to write your own mydriver.c file and write your own tests for functions. You can use the command

make myhw4

to create an executable from mydriver.c and hw4.c.

3. Edit your code

Before you start, please make sure to write your name and email at the beginning of hw4.c code to replace Prof. Wang's name and email.

```
// Author: Yanling Wang
// Email: yuw17@psu.edu
```

Edit your code in `hw4.c` with vim and save it and repeat step 1/2 to compile and test your code.

Always start with small changes before your compile and test your code.

Keep your code properly indented.

Add comment in your function to explain your algorithm for this assignment. Use block comments that are on its own lines. Do not use tail comments at the end of each line of the code.

Submit your C code

You are to submit the **hw4.c** file to gradescope. You should also submit your own **mydriver.c** file to gradescope, where it should contain **extra test cases beyond the ones provided in driver.c**.