Assignment 2: sed

CS 3424 - Systems Programming — Sam Silvestro

For this assignment you will use **sed** and **bash** to create a program for formatting C code. Bash will *only* be used to the extent that it will run your sed script using the sed command, supplying it with the names of the input files passed on the command line. Your program should take a source code file as input and apply the following formatting/content rules:

- No more than one space between tokens.
- No trailing whitespace after a line.
- Binary operators should always surrounded by a single space on either side (including assignment and Boolean). Only the following operators must be accounted for: `+`, `-`, `*`, `/`, `=`, `==`, `<=`, `>=`, `<`, `>`.
- Conditions should not have whitespace immediately inside of the parentheses.
- The program should *not* modify spaces which are leading, expanded tabs.
- Comments should be left alone. You may assume comments (single- and multi-line) will not appear on lines with source code.

Hint: All of the above will NOT be performed in a single pass.

This assignment requires the use of *only* sed and bash. **Do not** use awk, Python, or any other languages/utilities/commands.

Example

In the code below, underscores (_) represent spaces. Note that there are no changes to comments or #include lines.

Input (inputProgram.c):

```
1  /**
2  author:_____some_student
3  **/
4  #include_<stdio.h>
5
6  int_main()_{
7  ____int_numberIn;
8
9  ____printf("Enter_a_number:_");
10
11 ____scanf("%d",_&numberIn);__
12
13 ____if_(_numberIn_>_10_)_{
14 ____//____add__two
15 ________return_numberIn_+__2;
16 ____}_else___if____(numberIn<5){
17 ____//____subtract_two___
18 ________return__numberIn_-_2;
```

```

____return__numberIn*2;
}
```

Output (outputProgram.c):

```
/**
author:_____some_student
**/
#include_<stdio.h>

int_main()_{
____int_numberIn;

____printf("Enter_a_number:_");

____scanf("%d",_&numberIn);

____if_(numberIn_>_10)_{
____//____add__two
________return_numberIn_+_2;
____}_else_if_(numberIn_<_5){
____//____subtract_two___
________return_numberIn_-_2;

____return_numberIn_*_2;
}
```

Script Execution

Your program should be invoked through a single bash file (see below) with the path to the input program as the argument. The resulting output file should be printed directly to stdout.

```
$ assign2.bash ./path/to/input.txt
```

Assignment Data

A sample input file can be found in:
`/usr/local/courses/ssilvestro/cs3424/Spring23/assign2/inputProgram.c`.

Script Files

Your program should consist of *at least* two files:

- **`assign2.bash`** - the main file which is initially invoked
- **`assign2.sed`** - used by the sed invocation present in **`assign2.bash`**. This is your sed script which will contain multiple sed commands.

Verifying Your Program

Your program must work for *arbitrary* programs by applying the rules above. You can test your program with the input provided in **`inputProgram.c`** and compare the output with **`outputProgram.c`** using **`diff`** (check the man-pages on how to use it).

Submission

Turn your assignment in via Blackboard. Your submission should be a single ZIP archive named **`a2-`*`abc123`*`.zip`**, where *abc123* represents your myUTSA ID. This archive should contain only two files: your bash and sed scripts.