

# CSE 031 - Lab 1

Luis Andrade

January 31, 2020

# 1 punishments.c source code

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void printLine() {
5     char s [36] = "C programming language is the best!";
6     int i;
7     for (i = 0; i < 36; ++i ){
8         putchar(*(s+i));
9     }
10 }
11
12 int main(int argc, char const *argv[]) {
13     printf("\n");
14
15
16     printf("Enter # of lines to print: ");
17     int amount, i;
18     scanf("%d", &amount);
19
20     if (amount < 0) {
21         printf("Invalid input! \n");
22         return 0;
23     }
24     printf("\n");
25
26     int line;
27     printf("Enter the line of the typo: ");
28     scanf("%d", &line);
29     printf("\n");
30
31     if (line < 0 || line > amount) {
32         printf("Invalid input! \n");
33         return 0;
34     }
```

```
35
36     int j;
37     for (j = 0; j < amount; ++j) {
38         if (line-1 == j) {
39             printf("C++ programming language is the best!\n");
40         } else {printLine(); printf("\n");}
41     }
42
43     printf("\n");
44     return 0;
45 }
```

## 2 averages.c source code

```
1 #include <stdio.h>
2
3
4 int main(int argc, char const *argv[]) {
5
6     int input = 1;
7     int posSum, negSum, nc, pc = 0;
8
9     printf("\n");
10    while (input != 0) {
11        printf("Enter an integer: ");
12        scanf("%d", &input);
13        if (input < 0) {negSum += input; nc++;}
14        else if (input > 0) {posSum += input; pc++;}
15    }
16
17    if (pc != 0){printf("Positive average: %d\n",
18        posSum/pc);}
19    if (nc != 0){printf("Negative average: %d\n",
20        negSum/nc);}
21    printf("\n");
22    return 0;
23 }
```

### 3 TPS activity 2

1. On Linux, Linux OS in school computer labs, you can right click and select "open terminal here".
  - Yes
  - So you can multitask, edit/debug source code in one terminal, compile in another.
2. You can use the 'ls' bash command.
3. `$ cd; cd Desktop; mkdir CSE31; cd CSE31; mkdir Lab_1; cd Lab_1`

### 4 TPS activity 3

3. `gcc csourcefile.c -o executablename`
4. When you compile successfully with gcc there is no message indicating you did so, it simply returns execution back to the shell. When there is errors or warnings they will be printed out to std (bash) and compilation will not succeed.
5. Only run preprocess, compile, and assemble steps
6. Generate source-level debug information
7. You pass in the -o flag followed by a space and the name you want the executable to be saved as.
8. It creates an executable with the default name "a.out".
9. `./PATH/TO/executable`

## 5 Averages psudo Code

**\*\* this is a start, probably has errors but, I wrote this after writing the actual C source code.**

```
psudo:
positive_input, pc = 0
negative_input, pc = 0
user_input = 1
    while (user_input != 0):
        getUserInput()
        if (user_input > 0 && user_input IS_AN_INT):
            positive_input += user_input
            pc++

        elif (user_input < 0 && user_input IS_AN_INT)
            negative_input += user_input
            nc++

        else: print_to_stdout("Invalid")

output_to_stdout_averages()
```