Name: Eland Anthony

CS403/503 Programming Languages
Spring 2021
Assignment #3

1. Problem 19 of Chapter 7 on Page 327:

   Consider the following C program:

```
int fun(int *i) {
*i += 5;
return 4;
}
void main() {
int x = 3;
x = x + fun(&x);
}
```

   What is the value of x after the assignment statement in main, assuming

   1. operands are evaluated left to right.

   2. operands are evaluated right to left.

Answer:

1) When operands are left to right, x is determined before fun() can run, thus the arithmetic output is 3 + 4 = 7.

2) When operands are right to left, fun() still returns 4 and is run first; however, fun updates the result of x to 8. Thus, the arithmetic output is 8 + 4 = 12.

2. Programming Excise 7 of Chapter 7 on Page 328:

   Write a program in either C++, Java, or C# that illustrates the order of

   evaluation of expressions used as actual parameters to a method.

Answer:

Name: Eland Anthony

```
#include <stdio.h>
#include <iostream>


using namespace std;
```

```
void showOrder(int output, int number) {
cout << "The result of expression " << number << " is " <<
output << "\n";
}
```

```
int main() {
int a = 5; int b = 10; int c = 15; int d = 20;
showOrder(a * b + c * d, 1);
showOrder(a + b * c + d, 2);
showOrder((-a) / ++b * c + (-d), 3);
showOrder(a + b++ * (-c) / d, 4);
showOrder(a / b + c++ + (-d), 5);
return 0;
}
```

3. Programming Exercise 1 of Chapter 8 on Page 362:

   Rewrite the following pseudocode segment using a loop structure in the specified languages:

   k = (j + 13) / 27

   loop:

   if k > 10 **then goto** out

Name: Eland Anthony

```
k = k + 1
i = 3 * k - 1
goto loop
out: . . .
```

1. C, C++, Java, or C#
2. Python
3. Ruby

Answer:

C++:

```
int k = 0; int i = 0; int j = 0;
for(k = (j + 13) / 27; k <= 10; k++) {
   i = 3 *(k + 1) - 1;
}
```

Python:

```
k = (j + 13) / 27
while k <= 10:
        k += 1
        i = 3 * k-1
```

Ruby:

```
k = (j + 13) / 27
until k > 10[do]
   k = k + 1
   i = 3 * k - 1
end
```

Name: Eland Anthony

4. Programming Exercise 5 of Chapter 8 on Page 363:

In a letter to the editor of *CACM*, Rubin (1987) uses the following code segment as evidence that the readability of some code with gotos is better than the equivalent code without gotos. This code finds the first row of an *n* by *n* integer matrix named x that has nothing but zero values.

```
for (i = 1; i <= n; i++) {
for (j = 1; j <= n; j++)
if (x[i][j] != 0)
goto reject;
println ('First all-zero row is:', i);
break;
reject:
}
```

Rewrite this code without gotos in one of the following languages: C, C++, Java, or C#. Compare the readability of your code to that of the example code.

Answer:

```
#include <stdio.h>
#include <iostream>

using namespace std;
```

```
int main() {
```

```
int i = 0; int j = 0;
int x[i][j];
int n = 0;
```

Name: Eland Anthony

```
cout << "What is n? \n \n";
cin >> n;
cout << "\n";
for(i = 1; i <= n; i++){
int counter = 0;
for(j = 1; j <= n; j++){
if(x[i][i] == 0)
counter++;
}
if(counter == n){
cout<< "First all-zero row is: \n" << i << "\n";
break;
}
}
}
```

The claim by Rubin that gotos are better than the equivalent code without gotos seems to hold in this example. Utilizing ‘goto: reject’ allows for the program to continue the initial for loop **if** (x[i][j] != 0). When writing the program without using goto, a counter can be used to decipher whether an all zero row is found as the ‘counter’ is increased as each entry in a row is 0, and once the counter reaches the desired ‘n’ the all zero row has been identified. Here, the modified code without goto has more lines of code (ignoring the additional code added to make the program run). Thus, in this example the goto example is more readable; however, if code includes many goto statements, the reader may agree that the code is more complicated due to the need to jump to many different lines of the code repeatably, which likely complicates the nature of the program to a greater extent as opposed to a non goto implementation.