Josh Johnson

CS4121 – Program 1

1. Enter the following program into a Racket tab, save it in the brainfudge folder as prog1.bf, then run it. What does the program do? Explain how it works.

```
#lang reader "reader.rkt"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++.
+++++++.
+.
```

This program prints “MTU”. This is achieved by incrementing the first cell to the ASCII value of “M”, which is 77, and prints it. Next, the cell is incremented to 84, and prints “T”. Finally, it is incremented one more time to get “U”.

2. Enter the following program. Put your name after the exclamation point. Save it in the brainfudge folder as prog2.bf**,** then run it. What does the program do? Explain how it works.

```
#lang reader "reader.rkt"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++.---.+++++++..+++.----------+++-------------------------------
-----------------------------------------
++++++++++++++++++++++++++++++++.--------------------------------
,.[,.]!Josh
```

Similar to prog1, prog2 increments and decrements the first cell to print “`HELLO `”. The comma then reads the first input character into the first cell, and prints it. The loop “`[,.]`” reads each next input character into the first cell, and prints that as well. Once the read receives a 0 (EOF), the loop will exit and the program finishes execution. The “!” operator sends all preceding characters into stdin.

3. Write a program called length.bf that takes as input a string and prints the input string and its length separate by a space and having a newline at the end. Describe in detail how your program works.

length.bf begins by incrementing the first cell and begins a loop to read all characters from stdin and print them out. At the end of each iteration, the pointer is moved to the next cell, increments it, and moves back. This counts the length of the string. After printing the string and obtaining its length, a space is printed. Next, the value of the length cell is printed. This is done by finding how many times one can be subtracted from 10 until the 1’s place cell holds 0. This is also done with the 10’s and 100’s place.

4. Write a program called palindrome.bf that prints its input forwards and backwards (printing the last character of input only once) to produce a palindrome. Thus an input of rot prints rotor. And an input of race prints racecar. Describe in detail how your program works.

palindrome.bf begins by moving to the second cell and incrementing it. A loop begins to move to the next cell, read the next character into the cell, and print it. This will result in each character being read into its own cell. After all values are read, the pointer is moved back one to avoid printing the last character twice. Another loop begins to print each cell’s contents in backwards order.

5. Write a program called sum.bf that takes as input two double-digit numbers and outputs their three-digit sum. Thus an input of 02 and 02 prints 004. And an input of 99 and 99 prints 198. Describe in detail how your program works.

sum.bf begins by reading 5 characters. The first two are read and put into the first two cells. A dummy comma is placed to consume the space between the two numbers. The next two numbers are read into their respective cells. Each time a digit is read, 48 is subtracted to account for the ASCII values of the numbers (0 starts at 48). After each value is read and is in its respective cell, a loop goes through each one and increments cell 4 with either 1 or 10 for each number, depending on if it’s holding the ones or tens place. Once the sum is obtained, a print algorithm similar to that of the length.bf program is used. The only difference is there is no checking for 0-printing in the 100’s and 10’s digit.