Uriel Montes

TPS activity 2

1. To compile punishment.c so that i can run in debug it using gdb i need to type “cc -g punishment.c”. This will create an a.out file.
2. To load into gdb i need to type “gdb a.out”.
3. To run the program in gdb i need to type “run [args]”
4. Breakpoints are where the program will stop so that i can debug. To set a breakpoint at a certain line i need to type “break” and the line that i want to stop at.
5. To run the program line by line i need to type either “step” or “s”
6. I can see the value of a variable by typing “print” and the variable that i wish to see the value of.
7. To let the program finish its run you type either “continue” or “c”.
8. To exit gdb simply type “quit” or “q”.

TPS activity 3

1. There were 4 variables declared in line 1. 2 of them are pointers. They are pointers to the address of x and y.
2. The value of x is 1. The value of y is 0. The value of arr[0] is a random large number.
3. I think it happens that way because x is the first number that is printed and then y. Arr[0] is a random number because without inserting a specific number in the arr then a random number will occupy it.
4. To prevent unexpected values i need to make the variables equal to something.

Segmentation Faults

1. The line that caused the segmentation fault was line 23 where values = read\_values(sum)
2. To make it work you have to make it the address of sum
3. The bug is that sum has to be a pointer in the read\_values function
4. I made sum a pointer wherever it needed to be

Fix appendTest.c

1. The output is expected when i type HELLO! and hello!
2. The output is not expected when i type HI! and hi! The bug is that s1 is being written over. What was in s1 is used twice.
3. It isn't expected. This happens because the string can’t contain as many characters.