TPS activity 2:

1. To compile my punishment.c so I can debug it using GDB I type in the command line

gcc punishment.c -o punishment -g

1. To load it in GDB you simply enter gdb (file name)punishment
2. Type in run after gdb pops up on the side
3. Break points are where you want the program to run up to. To set a break point at a certain line in gdb you enter b and the number of the line of name of the function. Example in punishment.c I can set a break point by entering b 7 which sets a break point at line 7. To set one at the for loop I enter b 24 or break 24
4. To run the program line by line you can enter the command “next” in the gdb
5. To see the value of a variable you can type in info locals or print and the variable name.
6. To finish running line by line you can enter continue function so that it runs the whole program.
7. To exit gdb you enter the letter ‘q’

TPS Activity 3:

1. There are 4 variables
2. x = 0 ; y= 32764 ; px = 0x4….. py = 0x4….
3. To prevent x, y and the content of arr from having unexpected values you simply assign default values
4. -6. Are codes in the program so look into the program

7.Array names are not the same as pointers but their addresses are the same

8. The addresses are the same

Assignment 1 Segmentation Faults:

1. Line 15 caused the segmentation fault because it didn’t have a reference (&) it assumed the input as an address
2. To Fix the line you simply add the & in front of the variable input which is inside the scanf
3. Sum needs to be of type pointer
4. To fix the bug in the read\_values function you add \* to sum to make it double \*sum

Assignment 2, Individual Fix appendTest.c:

1. The output is not expected instead we get HELLO!hello!llo!
2. The output is HI!hi!hellhi! and it should be HI!hi!. To fix this we insert an array of s1 of the sum of s1len and s2len
3. Th output is not expected. I think it happens because the program has char str1 and char str2 with not enough space. The size is set to 10.
4. Code in program
5. Code in program
6. Code in program
7. The addresses are the same
8. Both of them point to the same address