TPS Activity 1:

1. I expect the program to print out the integer 2 because printf(“%x\n”, four\_ints[0]); because four\_ints[i] =4 and there is an array of size 4 and in the for loop i is initialize with 0 with the following line four\_ints[i] =2; meaning that the first index will have an element of 2 furthermore the print statememtn has four\_ints[0] which is the first index of the declared array so the print out should be 2.
2. If we were to insert the same printf statement at the very bottom then we would get 4 total pairs of 41 since the array is declared with a size of 4. So output would be 41414141
3. It prints out 41414141
4. It printed out:

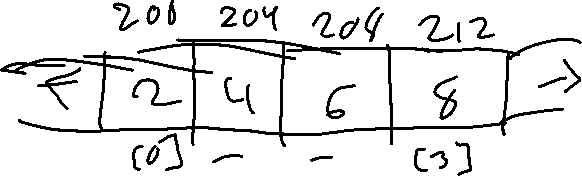
2

2

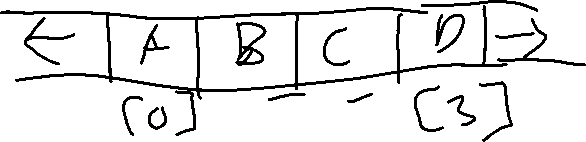
41414141

Im surprised, I thought it would print out 2 4 41414141

1. 1. There was an array of 4 elements allocated in this program.
   2. No they are not pointing to the same direction
   3. Fout\_ints[i] points to -67463872 and four\_c[i] points to 2
2. The difference between two consecutive elements can be found by example diff[i] = abs(arr[i]-arr[i+1]);



1. It points towards the array four\_ints which gives ascii code return values.



1. When the index of an array is icrementd the

**Exercise 2D arrays with malloc:**

1. Line 8 is declared as a pointer pointed to a dynamically allocated memory location with n being the size of the array. It is constructing a 2D array. Each row is an array of int so we use int\*\* pointing to each row.
2. To access the ith value of the array we dereference it so we insert &four\_c[i] into a print statement.
3. To access[i][j]
4. The size of the array needs to be passed as an argument because of its use throughout the program. Its used when making arr a diagonal matrix and when printing an array.