In-class Worksheet 1

1. C basics

   (a) What are the final values of x, y and `result` in this code snippet? [3 points]

   ```
   int x = 1;
   int y = 2;
   int result = 0;
   if ((x = x - 1) && (y = y - 2)) {
       result = 3;
   }
   ```

   [handwritten: x=0]

   [handwritten: AND shortcircuit.]

   x = **0**   y = **2**   result = **0**

   (b) What is the output of the following print statements? [4 points]

   ```
   printf("%c\t", mystery('K'));
   printf("%c\t", mystery(97));
   printf("%c\t", mystery(86));
   printf("%c\n", mystery('i'));

   int mystery(int ch) {
       if (ch >= 'a' && ch <= 'z') {
           return ch - 'a' + 'A';
       } else {
           return ch;
       }
   }
   ```

   *[handwritten:
   A — 65
   K — 75
   a — 97
   97 - 97 + 65
   = 65]*

   OUTPUT: **K A V I**

   (c) The function below tries to find the number of digits in a C string. For example, the sting "ABC123XYZ456#$%" has 6 digits. Fill the missing lines of code in this function.
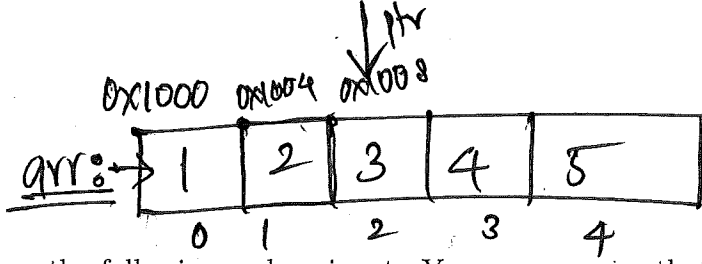
   ```
   int digits_in_string(char s[]) {
       int i;
       int count = ___0___;

       for (i = 0; __s[i] != '\0'__; i++) {
           if (_s[i] >= '0' && s[i] <= '9'_) {
               count++;
           }
       }
       return count;
   }
   ```

   *[handwritten:
   0 → 48
   zero
   1 — 49
   2 — 50]*

2. Pointers, Dreaded Pointers!

arr: (handwritten memory diagram)

0x1000 0x1004 0x1008 ↓ptr

| 1 | 2 | 3 | 4 | 5 |
  0   1   2   3   4

Answer the questions based on the following code snippet. You may assume that the **base location** (i.e., starting memory address) of the array arr is **0x1000**.

```
int arr[] = {1, 2, 3, 4, 5};
int *ptr = arr + 2;
```

(a) What are the values of the following expressions? In other words what is the value that will be printed if these expressions are used in a printf(). e.g., printf("%d", *arr); If the value of any expression is indeterminate, you should write "indeterminate". If any expression is illegal in C, you should write "illegal".

Expression	Value
*arr	1
ptr[0]	3
ptr[2]	5
ptr[4]	indeterminate
arr[-1]	u
ptr[-1]	2
3[arr]	4
3[ptr]	indeterminate

(b) Are the following expressions legal or illegal in C? If an expression is legal, write "LEGAL"' and write the **value** of the expression (when it is used in a print statement like printf("%p", ptr++);). If an expression is illegal, write "ILLEGAL" and explain **why**.

i. ptr++; ⟹ 0x1008

ii. ++arr;

(handwritten annotations surrounding the page:)

0x1000
+ 2
—————
⟹ 0x1000
+ 8
—————
0x·1008
*(&arr[0])
arr[0]

a[3]
*(a+3)
*(3+a)
3[a]

ptr[0]
= *(ptr + 0)
= *ptr

++ptr

arr[-1] ⟹ *(arr + (-1)) ⟹ *(0x1000 - 1) [int]
⟹ 0x̶F̶F̶C̶
0X FFC

2