ENSF 337 – Fall 2018 - Tutorial 1 – September 19

Exercise a: What is the output of the following program?
Note: the ASCII value of 'A' is 65 and the ASCII value of '

```
#include <stdio.h>
int main(){
 char ch1= 'F';
 char ch2 = 66;
 int i = 'A';
 int j = '9';
 printf(" Values are: %c  %d   %c   %c  %d", ch1 + 1, ch1 + 1,  ch2, i, i);
 return 0;
}
```

Exercise b: (1) What is the output of the following program if user enters: 33 44

```
int main(void){
    int a = 20, b = 30 , n;
    printf("Please enter two integer numbers: ");
    n = scanf("%d%d", &a, &b);
    printf("%d   %d    %d", a, b, n);
    return 0;
}
```

(2) What is the output if user enters: 44 aaa
(3) What is the output if user enters: aaa 44

Exercise c:

Draw memory diagram for points 1, 2, and 3 in the following program. This exercise has a couple tricky points that need understating of the details of basic constructs and memory allocation in C.

```
#include <stdio.h>
int func1(int x);
int func2(int a);
int func3 (int n);

int main(void){
    int  r1 = 65;
    r1 = func1(++r1);
    return 0;
}

int func1 (int bar){
    int y = func2(bar) + func3(bar -2);
    /* point 3 */
    return y;
}

int func2 (int c){
    int d = c % 100;
    /* Point 1 */
    return ++c;
}

int func3 (int a){
    int x = 10;
    /* Point 2 */
    return a+x;
}
```

Exercise d:
Draw a memory diagram for point one.

```
#include <stdio.h>
int main(void)
{
    int x = 99, y = 77, z = 55, w = 33, *p1 = &x;
    int *p2 = &y, *p3, *p4;

    p3 = &z;
    p4 = &w;

    p1 = p2;
    *p3 = *p4;

    *p1 -= 90;
    *p2 *= 100;
    *p3 %= 10;
    *p4 = 1;

    // point one
    return 0;
}
```

Exercise e:

Consider the following program.

- Read the program and try to predict the program output.
- Then, draw a memory diagram for points one and two.

```
#include <stdio.h>

int fun(int* x, int y){
    *x = 22;
    y = 34;
    // Point one
    return y;
}

int main(void){
    int a = 55;
    int b = 67;
    int c = fun(&a, b);
    printf("a = %d, b = %d, c = %d", a, b, c);
    // Point two
    return 0;
}
```