ENSF 337 Fall 2018 – Tutorial 4

Problem I - - Draw Memory Diagrams for points ONE in the following C program.

```
#include <stdio.h>
typedef struct Point_s {
    int x, y;
} Point;
typedef struct Circle_s {
    Point center;
    double radius;
} Circle;
typedef struct Cylinder_s {
    Circle* base;
    double height;
} Cylinder;

Point do_something (const Cylinder *arg1, Point* arg2){
    Point temp = {0, 0};
    temp.x += arg1->base->center.y;
    temp.y += arg2->y;
    // point One
    return temp;
}
```

```
int main(void ) {
    Circle ci = {{4, 5}, 2.0};
    Cylinder cy = {&ci, 60.0};
    cy.base->center.y = 77;
    (*cy.base).radius = 33.0;
    Point p = do_something (&cy,
                          &cy.base->center);
    return 0;
}
```

Problem II – Consider the following C program and draw a memory diagram when program reaches point one for the **second time**:

```
#include <stdio.h>
#define SIZE 6

typedef struct String{
    char storage [SIZE];
    int length;
}String;

String constructor (const char* s)
{
    String str;
    int i;
    str.length =0;
    for (i =0; s[i] != '\0'; i++)
        str.length++;

    for (i =0; s[i] != '\0'; i++)
        str.storage[i] = s[i];
    str.storage[i] = '\0';
     // Point one
    return str;
}
```

```
int main(){
  struct String arr[] = {constructor("ABC"), constructor ("BBC"), constructor ("CBC")};
  return 0;
}
```

Problem III: Consider the definition of `struct String`, defined in the previous exercise, and compete the definition of the following function:

```
const String* serarch (const String string_array, int n, const char target)
/* REQUIRES: string_array[0] ... string_array[n-1] exist, n > 0.
 * PROMISES: return a pointer to a String object that the value of its storage matches the
 *           argument target. Otherwise, if not found returns NULL.
 */
```