COEN 11- Homework 3 Structures
Solution

1. What does function main output?

```
//      Types
struct info
{
        char string[10];
        int array[10];
};

//      Functions
void b (struct info, struct info *);

//      Main
int main ( )
{
        struct info in1, in2;
        struct info *p;
        int i;

        p = &in1;
        strcpy (p->string, "abc");
        for (i = 0; i < 10; i++)
                p->array[i] = i;

        b (in1, &in2);

        printf ("%s, %d, %d\n", p->string, p->array[0], in2.array[0]);
        return 0;
}

//      Function b
void b (struct info x, struct info *y)
{
        x.string[0] = 'Z';
        *y = x;
        y->array[0]++;
        return;
}
```

OUTPUT: abc, 0, 1

2. Write a void function to initialize an array of structs, in which the members are a string (size STR_SIZE) and a positive integer number, with data obtained from the keyboard. The array has ARRAY_SIZE entries, and the array is global. Define the struct and declare the global

array.

```
struct x
{
   char   string[STR_SIZE];
   int    number;
};

struct x array[ARRAY_SIZE];

void
init ( )
{
        int i;

        for (i = 0; i < ARRAY_SIZE; i++)
                scanf ("%s%d", array[i].string, &array[i].number);

        return;
}
```

3. Write an int function to traverse the array from question 2, searching for a string received as an argument. If the string is found in one of the structs, the function returns the corresponding number. If the string is not found, the function returns -1.

```
int
search (char *str)
{
        int i;

        for (i = 0; i < ARRAY_SIZE; i++)
                if (strcmp (array[i].string, str) == 0)
                        return (array[i].number);

        return -1;
}
```

4. Write a void function to output the string and number in each struct of the array defined in question 2.

```
void output ( )
{
        int i;

        for (i = 0; i < ARRAY_SIZE; i++)
                printf ("%s, %d\n", array[i].string, array[i].number);

        return;
}
```

5. Write a void function to output the value of each member of each struct nonsense in array useless, which is global. Use loops when necessary.

```
struct nonsense
{
        float any_number;
        int nonsense_array[10];
};

struct nonsense useless[10];
```

```
void output ( )
{
        int i, j;
        for (i = 0; i < 10; i++)
        {
                printf ("%f\n", useless[i].any_number);
                for (j = 0; j < 10; j++)
                        printf ("%d\n", useless[i].nonsense_array[j]);
        }

        return;
}
```

6. Write a void function to traverse the array declared in Question 5 searching for a float number received as an argument. Use a struct nonsense pointer to traverse the array. If the number is found, the function outputs the corresponding int array. If not, it notifies the user.

```
void search (float f)
{
        int     i, j;           i;
        struct nonsense *p;

        p = useless;
        for (i = 0; i < 10; i++, p++)
        {
                if (p->any_number == f)
                for (j = 0; j < 10; j++)
                        printf ("%d\n", p->nonsense_array[j]);
        }

        printf ("not found\n");
        return;
}
```