EL-GY 6483 Real Time Embedded Systems
Programming HW 1

Name: Yogya Sharma
NetID: ys5250

1)

A) declare and initialize an array of twenty 32-bit signed integers.
B) define a function named "even_sum", that takes two parameters.

The first parameter should be a pointer to a 32-bit integer to pass in the array and the second should be a 32-bit integer that you will use to pass the length of the previously declared array. The function should return the sum of the array elements with even indices.

Answer 1)

A)
```
int arr[20] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20};
```

B)
```
int even_sum( int *arr, int length)
{  int sum = 0;
   for( int i = 0; i< length; i = i + 1)
     {
             if(i%2 == 0)
                     {sum = sum + arr[i];}
      }
   return sum;
}
```

2) Modify the previous function to also set all elements at odd indices to zero.

Answer 2)

As the question says to modify the function, keeping the name of the function same, and changing the definition for the required purpose.

```
void even_sum( int *arr, int length)
{   int sum = 0;
   for( int i = 0;  i< length; i = i + 1)
    {
             if(i%2 != 0)
                     {arr[i] = 0;}
    }
 }
```

3) Define a struct with 4 members, each member should be an unsigned int. Declare an array with 5 of the previously defined structs.

Answer 3)

```
        struct test {
                unsigned int a;
                unsigned int b;
                unsigned int c;
                unsigned int d;
                }

        struct test arr_ptr[5];
```

4) define a function named "zero_structs", that takes 2 parameters. The first parameter should be a pointer to a struct of the previously defined type to pass in the array of structs, and the second should be a 32-bit integer that you will use to pass the length of the array. The function should set all members of the structs in the passed-in array to zero.

Answer 4)

```
        void zero_structs(struct test *p, int length)
        {
                 for(int i = 0; i<length; i = i+1)
                {
                        p->a = 0;
                        p->b = 0;
                        p->c = 0;
                        p->d = 0;
                }
        }
```

5) Define a function named "fill_structs", which takes 3 parameters. The first parameter should be a pointer to a struct of the previously defined type to pass in the array of structs, the second should be a pointer to a 32-bit integer to pass in the array from part one, and the third should be a 32-bit integer that you will use to pass the length of the integer array. The function should set the members of the structs in the passed-in array to the elements of the 32-bit integer array that you pass in (i.e. the first struct's members should have the first 4 values of the integer array, the second struct should have the next 4 values, etc.)

Answer 5)

```
        void fill_structs( struct test *j, int *k, int length)
        {
                for( int i = 0; i<length; i = i + 4)
                {
```

```
        j->a = *k;
        k++;
        j->b = *k;
        k++;
        j->c = *k;
        k++;
        j->d = *k;
        k++;
        j++;

    }
}
```