CS 2213 Advanced Programming Recitation - Exercise

One-D Arrays and functions: Complete the following program. You will mainly implement the MERGE function, and a PRINT_ARRAY function!

```
main()
{
     /* 1.  Declare three integer arrays as follows */

     int  a[50], b[70], c[120];

     /* 2. implement a function set_array_rand(int x[], int n)
     and  call it to generate the values in array a and b
     randomly. */

     set_array_rand(a, 50);
     set_array_rand(b, 70);

     /* 3. using the selection_sort(double x[], int n) function
     we implemented in class, sort the elements in a and b
     arrays.  */

     selection_sort(a, 50);
     selection_sort(b, 70);

     /* 4. implement a MERGE function and call it as follows to
     merge the values in arrays a and b into array c such that
     the values in c will be sorted after merging */

     MERGE(a, 50, b, 70, c, 120);

     /* 5. print the values in array c */
     PRINT_ARRAY(“Array c”, c, 120);
}


void set_array_rand(int x[], int n)
{
/* 1.  randomly generate elements of x array, e.g, */
for(int i=0; i< n; i++)
     x[i] = rand_int(30, 100);
}

int rand_int(int a,int b)
{
   return rand()%(b-a+1) + a;
}
```

```
void selection_sort(int x[], int n)
{
  int k,j,m;
  double temp;

  for(k=0; k<=n-2; k++)  {
    m = k;
    for(j=k+1; j<=n-1; j++){
        if(x[j] < x[m])
          m = j;
    }
    temp = x[k];
    x[k] = x[m];
    x[m] = temp;
  }
}

void MERGE(int a[], int na, int b[], int nb, int c[], int nc)
{
/*  merge the values in a and b into c while keeping the values
    sorted. For example, suppose we have the following two
    Arrays a = { 3, 7, 9, 12} and b = {4, 5, 10}
    When we merge these two arrays, we will get
    c = {3, 4, 5, 7, 9, 10, 12}
*/


/* YOUR CODE */

}

PRINT_ARRAY(char *name, int x[], int nx)
{

   /* YOUR CODE */

}
```

You must submit your work using Blackboard Learn and respect the following rules:

1) All assignments must be submitted as either a zip or tar archive file unless it is a single pdf file.
2) Assignments must include all source code.
3) Assignments must include an output.txt file which demonstrates the final test output run by the student.
4) If your assignment does not run/compile, the output.txt file should include an explanation of what was accomplished, what the error message was that prevented the student from

finishing the assignment and what the student BELIEVES to be the underlying cause of the error.