```
/*##############################################################################*/
/*HW05_part2.c                                                                  */
/*_____________________________                                                 */
/*Written by Mustafa Akilli on March 22, 2015                                   */
/*                                                                              */
/*Description                                                                   */
/*___________                                                                   */
/*   Provides the following output values using the input values.               */
/*Inputs:                                                                       */
/*  -Array                                                                      */
/*Outputs:                                                                      */
/*  -Max Number in the Array                                                    */
/*  -Max Second Number in the Array                                             */
/*  -Sum All Array                                                              */
/*  -How many of the value                                                      */
/*  -Where is the value                                                         */
/*##############################################################################*/
/*                                                                              */
/*------------------------------------------------------------------------------*/
/*                           Includes                                           */
#include <stdio.h>
/*------------------------------------------------------------------------------*/

typedef enum{NOPE=-1}bool;
int max_array(int array[], int n);
int second_max_array(int array[], int n);
int sum_all_array (int array[], int n);
int count_array(int array[], int n, int value);
bool search_array (int array[], int n, int value);


int
main(void)
{

    int myarray[9]={6,8,3,3,12,8,3,8,2};
    int max,second_max,sum_all,count,count2,count3,search,search2,search3;

    max = max_array(myarray,9);
    second_max = second_max_array(myarray,9);
    sum_all = sum_all_array(myarray,9);

    printf("++++++++++++++++++++\n");
    printf("Maksimum array is %d\n",max);
    printf("++++++++++++++++++++\n");
    printf("Maksimum second array is %d\n",second_max);
    printf("++++++++++++++++++++\n");
    printf("Sum of all array is %d\n",sum_all);
    printf("++++++++++++++++++++\n");
    count = count_array(myarray,9,6);
    printf("%d\n",count);
    count2 = count_array(myarray,9,8);
    printf("%d\n",count2);
    count3 = count_array(myarray,9,3);
    printf("%d\n",count3);
    printf("++++++++++++++++++++\n");
    search = search_array(myarray,9,2);
    search2 = search_array(myarray,9,8);
    search3 = search_array(myarray,9,12);
    printf("++++++++++++++++++++\n");

    return 0;
}

/*****************************************************************************
 * array[0] assign to the max.                                               *
 * if array[location] bigger than max,                                       *
 * array[location] assign to the max.                                        *
 * when array is end,                                                        *
 * return max.                                                               *
 *****************************************************************************/
int max_array(int array[], int n)
```

```
{

    int max,location;

    max = array[0];

    for(location=1;location<n;++location)
    {
        if(max<array[location])
        {
            max = array[location];
        }
    }


    return max;
}

/**************************************************************************
 * if array[0] not equal to maximum value,                                *
 * array[0] assign to the second_max.                                     *
 * if array[location] bigger than second_max                              *
*  and array[location] not equal to maximum value,                        *
 * array[location] assign to the second_max.                              *
 * when array is end,                                                     *
 * return second_max.                                                     *
 **************************************************************************/

int second_max_array(int array[], int n)
{

    int second_max,location,max,temp;

    max = max_array(array,9);
    temp=array[0];
    if (max != temp)
        {
            second_max = array[location];
        }

    for(location=1;location<n;++location)
    {
        if(second_max<array[location])
        {
            temp=array[location];

            if (max != temp)
            {
                second_max = array[location];
            }
        }
    }

    return second_max;
}

/**************************************************************************
 * array[0] assign to the sum.                                            *
 * when array is not end,                                                 *
 * array[location] add to sum.                                            *
 * when array is end,                                                     *
 * return sum.                                                            *
 **************************************************************************/
int sum_all_array (int array[], int n)
{
    int sum,array_number,location;

    array_number = array[0];
    sum = array_number;

    for(location=1;location<n;++location)
    {
```

```
        array_number = array[location];
        sum = sum+array_number;
    }

    return sum;
}

/**************************************************************************
 * if value is equal to array[location],                                  *
 * count add to 1.                                                        *
 * when array is end,                                                     *
 * return count.                                                          *
 **************************************************************************/
int count_array(int array[], int n, int value)
{
    int location,array_number,count=0;

    for(location=0;location<n;++location)
    {
        array_number = array[location];

        if(array_number==value)
        {
            ++count;
        }
    }

    printf("Count of value %d is ",value);
    return count;
}

/**************************************************************************
 * if array[location] not equal to value,                                 *
 * location add to 1.                                                     *
 * if array[location] equal to value,                                     *
 * Prints location of value                                               *
 **************************************************************************/

bool search_array (int array[], int n, int value)
{
    int location;

    for(location=0;(location<n) && (array[location]!=value);++location)
    {

    }

    if(location==n)
    {
        location= NOPE;
    }

    printf("%d is at [%d]\n",value,location);
}
/*#########################################################################*/
/*                            End of HW05_part2.c                          */
/*#########################################################################*/
```