itemOperations

Write a C program for the following functions that work on arrays which can store up to 10 integers:

(1) **`void insert(int max, int *size, int ar[], int num);`**

This function inserts the number `num` into the array `ar` where the pointer parameter `size` stores the number of integers in `ar`. Before and after the function call, `ar` is an array of integers in **ascending order**. `max` is the maximum number of integers which can be stored in `ar`. This means that the function should issue an error message **`"The array is full\n"`** and no insertion should be done if `*size` is equal to `max` before insertion.

(2) **`void iremove(int *size, int ar[], int num);`**

This function removes the **first appearance** of the number `num` from the array `ar` which has `*size` numbers in it. Before and after the function call, `ar` is an array of integers in **ascending order**. Please note:

- After the number is removed, the message **`"The integer is removed\n"`** should be displayed.
- If `*size` is equal to zero, the error message **`"The array is empty\n"`** should be displayed.
- If `num` does not appear in `ar`, the function should issue an error message **`"The number is not in the array\n"`**.

(3) **`void initialize(int *size, int ar[]);`**

This function reads in a specified number of integers and uses **`insert()`** to store them in `ar`. The pointer parameter `size` returns the actual number of integers stored in `ar`, and `ar` will be an array of integers in **ascending order**.

(4) **`void display(int size, int ar[]);`**

This function prints the numbers stored in `ar`. `size` gives the number of integers stored in `ar`.

A sample program template is given below for testing the functions:

```
#include <stdio.h>
#define MAX 10
void initialize(int *size, int ar[]);
void insert(int max, int *size, int ar[], int num);
void iremove(int *size, int ar[], int num);
void display(int size, int ar[]);
int main()
{
   int  option = 0;
   int  num, ar[MAX], size = 0;

   printf("Please select an option: \n");
   printf("(1) Initialize the array \n");
   printf("(2) Insert an integer \n");
```

```
   printf("(3) Remove an integer \n");
   printf("(4) Display the numbers stored in the array \n");
   printf("(5) Quit \n");
   do {
      printf("Enter your choice: \n");
      scanf("%d", &option);
      switch (option) {
         case 1:
            initialize(&size, ar);
            break;
         case 2:
            printf("Enter an integer: \n");
            scanf("%d", &num);
            insert(MAX, &size, ar, num);
            break;
         case 3:
            printf("Enter the integer to be removed: \n");
            scanf("%d", &num);
            iremove(&size, ar, num);
            break;
         case 4:
            display(size, ar);
            break;
         default:
            break;
      }
   } while (option < 5);
   return 0;
}
void display(int size, int ar[])
{
   int i;

   printf("The %d numbers in the array: \n", size);
   for(i = 0; i < size; i++)
      printf("%d ", ar[i]);
   printf("\n");

}
void initialize(int *size, int ar[])
{
   int total, i, num;

   printf("Enter the total number of integers (MAX=%d): \n", MAX);
   scanf("%d", &total);
   (*size) = 0;
   printf("Enter the integers: \n");
   for (i = 0; i < total; i++) {
      scanf("%d", &num);
      insert(MAX, size, ar, num);
   }
}
void insert(int max, int *size, int ar[], int num)
{
   /* Write your code here */
}
```

```
void iremove(int *size, int ar[], int num)
{
   /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1: **/* initialize */**

```
Enter your choice:
1
Enter the total number of integers (MAX=10):
5
Enter the integers:
3 5 2 1 4
Enter your choice:
4
The 5 numbers in the array:
1 2 3 4 5
Enter your choice:
5
```

(2) Test Case 2: **/*insert */**

```
Enter your choice:
1
Enter the total number of integers (MAX=10):
9
Enter the integers:
5 6 7 8 10 1 2 3 4
Enter your choice:
2
Enter an integer:
9
Enter your choice:
4
The 10 numbers in the array:
1 2 3 4 5 6 7 8 9 10
Enter your choice:
2
Enter an integer:
11
The array is full
Enter your choice:
5
```

(3) Test Case 3: **/* iremove */**

```
Enter your choice:
1
Enter the total number of integers (MAX=10):
2
Enter the integers:
5 6
Enter your choice:
3
Enter the integer to be removed:
```

```
5
The integer is removed
Enter your choice:
3
Enter the integer to be removed:
4
The number is not in the array
Enter your choice:
3
Enter the integer to be removed:
6
The integer is removed
Enter your choice:
3
Enter the integer to be removed:
3
The array is empty
Enter your choice:
5
```