Name : Bankar Krushna Lahanubhau
MIS : 612303031
SY COMP Division – 1
Batch – S2

Array assignment:

code

main.c

```
#include <stdio.h>
#include "array.h"
#include <stdlib.h>

enum {INIT, APPEND, DISPLAY, REMOVE, MAX, MIN, MERGE,
INSERT, QUIT};

int main() {

    array a;
    array b;
    int choice, index, num, size;
    while(1) {
        display_menu();
        scanf("%d", &choice);
        switch(choice) {
            case INIT:
                printf("Enter size of array a: \n");
                scanf("%d", &size);
                init(&a, size);
                printf("Enter size of array b: \n");
                scanf("%d", &size);
                init(&b, size);
                break;
            case APPEND:
                append(&a, rand());
                append(&b, rand());
                break;
            case DISPLAY:
                printf("Array a elements are: \n");
                display(a);
                printf("Array b elements are: \n");
                display(b);
                break;
            case REMOVE:
                printf("Enter the index to remove element: \n");
                scanf("%d", &index);
                remove_at_index(&a, index);
                break;
            case MAX:
                printf("%d is the maximum element in the array a: \n", max(a));
                printf("%d is the maximum element in the array b: \n", max(b));
                break;
            case MIN:
                printf("%d is the minimum element in the array a: \n", min(a));
                printf("%d is the minimum element in the array b: \n", min(b));
                break;
            case MERGE:
                printf("Array after merge of a and b: \n");
                merge(&a, &b);
                break;
            case INSERT:
                printf("Enter element value and index to insert in array(element
index): \n");
                scanf("%d%d", &num, &index);
                insert_at_index(&a, index, num);
                printf("Array after insert operation: \n");
                display(a);
                break;
            case QUIT:
                return 0;
            default:
                printf("Enter valid choice: \n");
                break;

        }
        }

                    return 0;
}
```

array.c

```
#include <stdio.h>
#include "array.h"
#include <stdlib.h>

void init(array *a, int size) {

                a->A = (int *)malloc(sizeof(int)*size);
                a->size = size;
   /* Modified to intialize array with randome element between 0 to 100 as instructed
by lab faculty */
    for(int i = 0; i < size; i++) {
        a->A[i] = rand()%100;
    }

    a->len = size;
                return;
}

void append(array *a, int ele) {
    int *newArray = (int *)realloc(a->A, sizeof(int) * (a->size + 1));
    a->A = newArray;         /* Updating the pointer to the newly allocated memory */
    a->A[a->len] = ele;      /* Append the new element */
    a->len++;                /* Incrementing the length */
    a->size++;               /* Incrementing the size since added a new element */

    return;
}

void display(array a) {

                for(int i = 0; i < a.len; i++) {
                        printf("%d |", a.A[i]);
                }

                printf("\n");
```

```
                              return;
}

void remove_at_index(array *a, int index) {

   if (index >= a->len || index < 0) {
      return;  /* Index is out of bounds */
   }

   int i = index;

   /* Shifting elements left to fill the gap */
   while (i < a->len - 1) {
      a->A[i] = a->A[i + 1];
      i++;
   }

   /*Decreasing the length of the array */
   a->len--;

   /* resizing the array */
   int *newArray = (int *)realloc(a->A, sizeof(int) * a->len);
   a->A = newArray;
   a->size--;
   return;
}

void insert_at_index(array *a, int index, int ele) {

   if (index > a->len || index < 0) {
      return;
   }

   int *newArray = (int *)realloc(a->A, sizeof(int) * (a->size + 1));
   a->A = newArray;
   a->size++;

   int i = a->len - 1;

   while (i >= index) {
      a->A[i + 1] = a->A[i];
      i--;
   }
```

```
    a->A[index] = ele;
    a->len++;

    return;
}

int max(array a) {

                        int maximum = a.A[0];

                        for(int i = 1; i < a.len; i++) {
                                if(maximum < a.A[i]) {
                                        maximum = a.A[i];
                                }
                        }

                        return maximum;
}

int min(array a) {

                        int minimum = a.A[0];

                        for(int i = 1; i < a.len; i++) {
                                if(minimum > a.A[i]) {
                                        minimum = a.A[i];
                                }
                        }

                        return minimum;
}

void swap(int *a, int *b) {

                        int temp = *a;
                        *a = *b;
                        *b = temp;

                        return;
}

void reverse(array *a) {

                        int i, j;
                        i = 0;
                        j = a->len - 1;

                        while(i < j) {
                                swap(&a->A[i], &a->A[j]);
                                i++;
                                j--;
                        }

                        return;
}


void merge(array *a, array *b) {
   /*Reallocating memory for 'a' to accommodate elements from both arrays*/
   int *newArray = (int *)realloc(a->A, sizeof(int) * (a->len + b->len));

   a->A = newArray;
   a->size += b->size;

   for (int i = 0; i < b->len; i++) {
      a->A[a->len] = b->A[i];
      a->len++;
   }

   return;
}


void display_menu() {

                        printf("Please select apropriate choice: \n");
   printf("1 -> To initialize arrays\n");
                        printf("2 -> To append element to the array\n");
                        printf("3 -> Display contents of array\n");
                        printf("4 -> Remove element at specified index\n");
                        printf("5 -> Print maximum element of the array\n");
                        printf("6 -> Print minimum element of the array\n");
                        printf("7 -> Merge two arrays\n");
                        printf("8 -> Insert element at given index\n");
   printf("9 -> Quit\n");
   printf("\n");
                        return;
}
```

array.h

```
typedef struct array {
                        int *A;
                        int len;
                        int size;
}array;

void init(array *a, int size);
void append(array *a, int ele);
void display(array a);
void remove_at_index(array *a, int index);
void insert_at_index(array *a, int index, int ele);
int max(array a);
int min(array a);
void reverse(array *a);
void merge(array *a, array *b);
void swap(int *a, int *b);
void display_menu();
```

OUTPUT:

```
 [~/dsa/dsa/array]
 krushna Ξ ∡ foss-lab - Ξ gcc -Wall -c main.c array.c array.h

 [~/dsa/dsa/array]
 krushna Ξ ∡ foss-lab - Ξ cc main.o array.o -o array

 [~/dsa/dsa/array]
 krushna Ξ ∡ foss-lab - Ξ ./array
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

1
```

```
Enter size of array a:
4
Enter size of array b:
3
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

3
Array a elements are:
83 |86 |77 |15 |
Array b elements are:
93 |35 |86 |
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

2
Enter array name (a or b):
a
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit
```

```
3
Array a elements are:
83 |86 |77 |15 |92 |
Array b elements are:
93 |35 |86 |
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

4
Enter array name (a or b):
a
Enter the index to remove element:
2
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

3
Array a elements are:
83 |86 |15 |92 |
Array b elements are:
93 |35 |86 |
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
```

```
8 -> Insert element at given index
9 -> Quit

5
92 is the maximum element in the array a:
93 is the maximum element in the array b:
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

6
15 is the minimum element in the array a:
35 is the minimum element in the array b:
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

7
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

3
Array a elements are:
```

```
83 |86 |15 |92 |93 |35 |86 |
Array b elements are:
93 |35 |86 |
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

8
Enter array name (a or b):
b
Enter element value and index to insert in array(element index):
4 2
Array after insert operation:
93 |35 |2 | 86 |
Please select apropriate choice:
1 -> To initialize arrays
2 -> To append element to the array
3 -> Display contents of array
4 -> Remove element at specified index
5 -> Print maximum element of the array
6 -> Print minimum element of the array
7 -> Merge two arrays
8 -> Insert element at given index
9 -> Quit

9

[~/dsa/dsa/array]
 krushna Ξ ∡ foss-lab - Ξ
```