Nisarg Wath
642403014

Assignment 1

Header.h

```
typedef struct Array{
    int * data;
    int size;
    int len;
}Array;

void init(Array * a, int size);
void  append(Array * a, int data);
void  insert_at_index(Array * a, int ind,int value);
void remove_at_index(Array *a, int ind);
void  display(Array * a);
int max(Array * a);
int  min (Array *a);
void  reverse(Array * a) ;
void  merge(Array *a, Array *b);
```

Logic / array.c

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


void init(Array * a, int size){
    a->data = (int *)malloc(sizeof(int)*size);
    a->size = size;
    a->len = 0;
}

void  append(Array * a, int data){
    if(a->size-1 < a->len) return;
        a->data[a->len] = data;
        a->len++;

    return;
}


void  insert_at_index(Array * a, int ind,int value){
    if(ind > a->size || ind < 0) return;
```

```
        for(int i = 0;i<a->size;i++){
            if(i == ind){
                a->data[i]= value;
                a->len++;
            }
        }
        return;

}
void  remove_at_index(Array *a, int ind){
    for(int i = 0;i<a->size;i++){
            if(i == ind){
                a->data[i] = a->data[i+1];
                break;
            }
        }
        a->len--;
return;

}
void  display(Array * a){
    for(int i = 0;i<a->len;i++){
        printf("%d ", a->data[i]);
    }
    printf("\n");
}
int min(Array * a){
    int minmum = a->data[0];
    for(int  i = 0;i<a->len;i++){
        if(a->data[i]  <= minmum ){
            minmum  =a->data[i];
        }
    }
    return minmum ;
}
int max(Array *a){
    int maxmium  = a->data[0];
    for(int  i = 0;i<a->len;i++){
        if(a->data[i]  > maxmium ){
            maxmium  =a->data[i];
        }
    }
    return maxmium ;
}

void  reverse(Array * a){
   int l = 0, h = a->len-1;

   while(l<h){
        int temp = a->data[l];
        a->data[l] = a->data[h];
        a->data[h] =temp;
```

```
        l++;
        h--;
    }
    printf("\n");
}
//
void  merge(Array *a, Array *b){
    a->size = a->size+b->size;
    for(int i = 0;i<b->len;i++){
        append(a,b->data[i]);
    }
    return;
}
```

Main.c

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


int main(){
    Array a;
    int size;
    printf("\nEnter the Size of the Array:\t");
    scanf("%d", &size);
    init(&a, size);
    int choice , r;
    do
    {
       printf("Enter 1)Append 2)Insert At Index \n3)Remove At
Index 4)Display \n5)Max 5)Min \n7)Reverse the list 8)Merge the
list");
       scanf("%d", &choice);
       switch (choice)
       {
       case 1:
           append(&a, rand());
        break;
       case 2:
       printf("\nEnter index and value to append:-\t");
       int ind, val;
       scanf("%d %d", &ind, &val);
            insert_at_index(&a, ind, val);
        break;

        case 3:
            printf("\nEnter index to delete:-\t");
            int in;
       scanf("%d", &in);
```

```
            remove_at_index(&a, in);
        break;

        case 4:
            printf("List :-\t");
            display(&a);
        break;
        case 5:
            printf("max element is :-\t");
          printf("\n%d \n", max(&a));
        break;
         case 6:
            printf("minimum element is :-\t");
            printf("\n%d \n", min(&a));

        break;
         case 7:
            printf("Before Reverse list is :-\t");
            display(&a);
            printf("After Reverse list is :-\t");
            reverse(&a);
            display(&a);

        break;
        break;
         case 8:
            printf("before Merging list is :-\t");
            display(&a);
            printf("After Merging list is :-\t");
            merge(&a,&a);
            display(&a);

        break;


       default:
       printf("Invalid!!");
        break;
       }
        printf("Enter 0  to exit !!");
        scanf("%d",&r);

    } while (r != 0);



    return 0;
}
```

Output photo

```
appleApple@Nisargs-MacBook-Air build % cd Assigment/Assigment1
appleApple@Nisargs-MacBook-Air Assigment1 % ./out

Enter the Size of the Array:    4
Enter 1)Append 2)Insert At Index
3)Remove At Index 4)Display
5)Max 5)Min
7)Reverse the list 8)Merge the list1
Enter 0  to exit !!1
Enter 1)Append 2)Insert At Index
3)Remove At Index 4)Display
5)Max 5)Min
7)Reverse the list 8)Merge the list2

Enter index and value to append:-       1
23
Enter 0  to exit !!1
Enter 1)Append 2)Insert At Index
3)Remove At Index 4)Display
5)Max 5)Min
7)Reverse the list 8)Merge the list4
List :- 16807 23
Enter 0  to exit !!
```