ASSIGNMENT 2

Q1:

```c
#include <stdio.h>

int linearSearch(int array[], int side, int t) {
    for (int i = 0; i < side; i++) {
        if (array[i] == t) {
            return 1;
    }
    return 0;
}

int main() {
    int array[100];
    int side, t;

    printf("Enter the size of the array: ");
    scanf("%d", &side);

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < side; i++) {
        scanf("%d", &array[i]);
    }

    printf("Enter the number to search: ");
    scanf("%d", &t);

    if (linearSearch(array, side, t)) {
        printf("Number %d is present in the array.\n", t);
    } else {
        printf("Number %d is not present in the array.\n", t);
    }
    return 0;
}
```

Q2:

```c
#include <stdio.h>
#include <limits.h>

void findSecondMaxAndMin(int array[], int size, int *secmax, int *secmin) {
```

ASSIGNMENT 2

```c
    int max = INT_MIN;
    int secondMaxValue = INT_MIN;
    int min = INT_MAX;
    int secondMinValue = INT_MAX;

    for (int i = 0; i < size; i++) {
        if (array[i] > max) {
            secondMaxValue = max;
            max = array[i];
        } else if (array[i] > secondMaxValue && array[i] != max) {
            secondMaxValue = array[i];
        }

        if (array[i] < min) {
            secondMinValue = min;
            min = array[i];
        } else if (array[i] < secondMinValue && array[i] != min) {
            secondMinValue = array[i];
        }
    }

    *secmax = secondMaxValue;
    *secmin = secondMinValue;
}

int main() {
    int array[100];
    int size;

    printf("Enter the number of elements in the array: ");
    scanf("%d", &size);

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &array[i]);
    }

    int secmax, secmin;
    findSecondMaxAndMin(array, size, &secmax, &secmin);

    printf("Second maximum element: %d\n", secmax);
    printf("Second minimum element: %d\n", secmin);

    return 0;
}
```

ASSIGNMENT 2

Q3:

```
#include <stdio.h>

int main() {
    int arr[100];
    int size = 0;

    while (1) {

 printf("\nArray Operations Menu:\n");
        printf("1. Insert element\n");
        printf("2. Delete element\n");
        printf("3. Traverse array\n");
        printf("4. Exit\n");

        int choice;
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {

            case 1: {
                int position, element;

  printf("Enter the position for insertion: ");
                scanf("%d", &position);

    printf("Enter the element to insert: ");
                scanf("%d", &element);
                if (position >= 0 && position <= size) {
                    for (int i = size; i > position; i--) {
                        arr[i] = arr[i - 1];
                    }
                    arr[position] = element;
                    size++;
                    printf("Element inserted.\n");
                } else {
                    printf("Invalid position for insertion.\n");
                }
                break;
            }

            case 2: {
                int position;
                printf("Enter the position for deletion: ");
                scanf("%d", &position);
                if (position >= 0 && position < size) {
                    for (int i = position; i < size - 1; i++) {
                        array[i] = array[i + 1];
                    }
                    size--;
                    printf("Element deleted.\n");
                } else {
                    printf("Invalid position for deletion.\n");
                }
                break;
            }

            case 3: {
                if (size == 0) {
                    printf("Array is empty.\n");
                } else {
                    printf("Array elements:");
                    for (int i = 0; i < size; i++) {
                        printf(" %d", arr[i]);
                    }
                    printf("\n");
                }
                break;
            }

            case 4:
                printf("Exiting the program.\n");
                return 0;

            default:
                printf("Invalid choice. Please select a valid option.\n");
                break;
        }
    }

    return 0;
}
```

Q4:

```
#include <stdio.h>
```

ASSIGNMENT 2

```c
void operateArrays(int arr1[], int arr2[], int result[], int size, char
openheimer) {

     for (int i = 0; i < size; i++) {

       switch (openheimer) {
            case '+': result[i] = arr1[i] + arr2[i]; break;
            case '-': result[i] = arr1[i] - arr2[i]; break;
            case '*': result[i] = arr1[i] * arr2[i]; break;
        }
    }
    printf("Arrays %c performed.\n", openheimer);
}

void displayArray(int arr[], int size) {
    printf("Array elements:");

   for (int i = 0; i < size; i++) printf(" %d", arr[i]);
    printf("\n");
}

int main() {
    int arr1[100], arr2[100], result[100], size;

    printf("Enter the size of the arrays: ");
    scanf("%d", &size);

    printf("Enter the elements of the first array:\n");
    for (int i = 0; i < size; i++) scanf("%d", &arr1[i]);

    printf("Enter the elements of the second array:\n");
    for (int i = 0; i < size; i++) scanf("%d", &arr2[i]);



    while (1) {
        printf("\nArray Operations Menu:\n1. Addition\n2. Subtraction\n3.
Multiplication\n4. Display Result Array\n5. Exit\n");
        printf("Enter your choice: ");

        int choice;
        scanf("%d", &choice);

        switch (choice) {
            case 1: case 2: case 3:
                operateArrays(arr1, arr2, result, size, choice == 1 ? '+' :
(choice == 2 ? '-' : '*'));
                break;

         case 4:
                displayArray(result, size);
                break;

            case 5:
                printf("Exiting the program.\n");
                return 0;
            default:
                printf("Invalid choice. Please select a valid option.\n");
                break;
        }
    }
}
```

Q5:

```
#include <stdio.h>

void mergeSortedArrays(int arr1[], int size1, int arr2[], int size2, int
result[]) {

 int i = 0, j = 0, k = 0;

    while (i < size1 && j < size2) {
        result[k++] = (arr1[i] <= arr2[j]) ? arr1[i++] : arr2[j++];
    }

    while (i < size1) result[k++] = arr1[i++];

    while (j < size2) result[k++] = arr2[j++];
}

int main() {
    int arr1[100], arr2[100], result[200];

 int size1, size2;

    printf("Enter the size of the first sorted array: ");
    scanf("%d", &size1);
    printf("Enter the elements of the first sorted array:\n");
```

ASSIGNMENT 2

```
    for (int i = 0; i < size1; i++) scanf("%d", &arr1[i]);

    printf("Enter the size of the second sorted array: ");
    scanf("%d", &size2);

   printf("Enter the elements of the second sorted array:\n");
    for (int i = 0; i < size2; i++) scanf("%d", &arr2[i]);

    mergeSortedArrays(arr1, size1, arr2, size2, result);

    printf("Merged sorted array:");
    for (int i = 0; i < size1 + size2; i++) printf(" %d", result[i]);

    printf("\n");

    return 0;
}
```

Q6:

```
1) #include <stdio.h>


int linearSearch(int *array, int size, int target) {
    for (int i = 0; i < size; i++) {

   if (array[i] == target) {
            return 1;
        }
    }

return 0;
}

int main() {
    int array[100];
    int size, target;

    printf("Enter the size of the array: ");

scanf("%d", &size);

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &array[i]);
```

ASSIGNMENT 2

```c
    }

    printf("Enter the number to search: ");
    scanf("%d", &target);

    if (linearSearch(array, size, target)) {
        printf("Number %d is present in the array.\n", target);
    }
else {
        printf("Number %d is not present in the array.\n", target);
    }

    return 0;
}
```

2)

```c
#include <stdio.h>

void findSecondMaxMin(int *array, int size, int *secondMax, int *secondMin) {
    *secondMax = *secondMin = array[0];

    for (int i = 0; i < size; i++) {

        if (array[i] > *secondMax) {
            *secondMin = *secondMax;
            *secondMax = array[i];
        } else if (array[i] > *secondMin && array[i] != *secondMax) {
            *secondMin = array[i];
        }
    }
}

int main() {
    int array[100];
    int size, secondMax, secondMin;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &array[i]);
    }
```

ASSIGNMENT 2

```
    findSecondMaxMin(array, size, &secondMax, &secondMin);

    printf("Second maximum element: %d\n", secondMax);
    printf("Second minimum element: %d\n", secondMin);

    return 0;
}
```

```
3) #include <stdio.h>

void insertElement(int *array, int *size, int position, int element) {
    if (*size >= 100) {
        printf("Array is full, cannot insert.\n");
        return;}
    if (position < 0 || position > *size) {
        printf("Invalid position for insertion.\n");
        return;}
    for (int i = *size; i > position; i--) {
        array[i] = array[i - 1];}
    array[position] = element;
    (*size)++;
    printf("Element inserted.\n");}
void deleteElement(int *array, int *size, int position) {
    if (*size == 0) {
        printf("Array is empty, cannot delete.\n");
        return;}
    if (position < 0 || position >= *size) {
        printf("Invalid position for deletion.\n");
        return;}
    for (int i = position; i < *size - 1; i++) {
        array[i] = array[i + 1];}
    (*size)--;
    printf("Element deleted.\n");}
void traverseArray(int *array, int size) {
    if (size == 0) {
        printf("Array is empty.\n");
        return;}
    printf("Array elements:");
    for (int i = 0; i < size; i++) {
        printf(" %d", array[i]);}
    printf("\n");}
int main() {
    int array[100];
    int size = 0;
    while (1) {
```

ASSIGNMENT 2

```c
        printf("\nArray Operations Menu:\n");
        printf("1. Insert element\n");
        printf("2. Delete element\n");
        printf("3. Traverse array\n");
        printf("4. Exit\n");
        int choice;
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {case 1:
                int position, element;
                printf("Enter the position for insertion: ");
                scanf("%d", &position);
                printf("Enter the element to insert: ");
                scanf("%d", &element);
                insertElement(array, &size, position, element);
                break;
            case 2:
                printf("Enter the position for deletion: ");
                scanf("%d", &position);
                deleteElement(array, &size, position);
                break;
            case 3:
                traverseArray(array, size);
                break;
            case 4:
                printf("Exiting the program.\n");
                return 0;
            default:
                printf("Invalid choice. Please select a valid option.\n");
                break;
        }}return 0;}
```