



IQRA NATIONAL UNIVERSITY PESHAWAR

DEPTT. B.E. (ELECTRICAL)

8TH SEMESTER

SPRING 2020

SESSIONAL ASSIGNMENT

DATA STRUCTURE AND ALGORITHMS

NAME : ABDUL MATEEN

ID : 13009

INSTRUCTOR:
MUHAMMAD ADIL ASST: PROF

DATE OF SUBMISSION:
16 / June / 2020

Task# 1

Design a Program (in any Computer Language of your choice) to implement the Algorithms to

- Travers an Array using for loop 02
- Traverse an Array using while loop 02
- Insert an element in an Array 02
- Delete an element from an Array 02

Code along with the output screenshot should be submitted.

1. Travers An Array Using For Loop:

Code:

```
#include <stdio.h>
main() {
    int LA[] = {1,3,5,7,8};
    int item = 10, k = 3, n = 5;
    int i = 0, j = n;
    printf("The original array elements are :\n");
    for(i = 0; i<n; i++) {
        printf("LA[%d] = %d \n", i, LA[i]);
    }
}
```

Output:

```
The original array elements are :
LA[0] = 1
LA[1] = 3
LA[2] = 5
LA[3] = 7
LA[4] = 8
```

2. Traverse an Array using while loop:

Code:

```
#include <stdio.h>
main() {
    int LA[] = {1,3,5,7,8};
    int item = 10, k = 3, n = 5;
    int i = 0, j = n;
    printf("The original array elements are :\n");
    n = n+1;
    while( j >= k) {
        LA[j+1] = LA[j];
        j = j - 1;
    }
}
```

Output:

```
The original array elements are :
LA[0] = 1
LA[1] = 3
LA[2] = 5
LA[3] = 7
LA[4] = 8
```

3. Insert an element in an Array:

Code:

```
#include <stdio.h>

main() {
    int LA[] = {1,3,5,7,8};
    int item = 10, k = 3, n = 5;
    int i = 0, j = n;

    printf("The original array elements are :\n");

    for(i = 0; i<n; i++) {
        printf("LA[%d] = %d \n", i, LA[i]);
    }

    n = n + 1;

    while(j >= k) {
        LA[j+1] = LA[j];
        j = j - 1;
    }

    LA[k] = item;

    printf("The array elements after insertion :\n");

    for(i = 0; i<n; i++) {
        printf("LA[%d] = %d \n", i, LA[i]);
    }
}
```

Output:

```
The original array elements are :
LA[0] = 1
LA[1] = 3
LA[2] = 5
LA[3] = 7
LA[4] = 8
The array elements after insertion :
LA[0] = 1
LA[1] = 3
LA[2] = 5
LA[3] = 10
LA[4] = 7
LA[5] = 8
```

4. Delete an element from an Array:

Code:

```
#include <stdio.h>

void main() {
    int LA[] = {1,3,5,7,8};
    int k = 3, n = 5;
    int i, j;

    printf("The original array elements are :\n");

    for(i = 0; i<n; i++) {
        printf("LA[%d] = %d \n", i, LA[i]);
    }

    j = k;

    while(j < n) {
        LA[j-1] = LA[j];
        j = j + 1;
    }

    n = n - 1;

    printf("The array elements after deletion :\n");

    for(i = 0; i<n; i++) {
        printf("LA[%d] = %d \n", i, LA[i]);
    }
}
```

Output:

```
The original array elements are :
LA[0] = 1
LA[1] = 3
LA[2] = 5
LA[3] = 7
LA[4] = 8
The array elements after deletion :
LA[0] = 1
LA[1] = 3
LA[2] = 7
LA[3] = 8
```

Task# 2

Design a Program (in any Computer Language of your choice) to implement the Algorithms to implement Linear Search.

Code along with the output screenshot should be submitted.

06

Code:

```
#include <stdio.h>

#define MAX 20
int intArray[MAX] =
{1,2,3,4,6,7,9,11,12,14,15,16,17,19,33,34,43,45,55,66};
void printline(int count) {
    int i;
    for(i = 0;i <count-1;i++) {
        printf("=");
    }
    printf("\n");
}
int find(int data) {
    int comparisons = 0;
    int index = -1;
    int i;
    for(i = 0;i<MAX;i++) {

        comparisons++;
        if(data == intArray[i]) {
            index = i;
            break;
        }
    }
    printf("Total comparisons made: %d", comparisons);
    return index;
}
void display() {
    int i;
    printf("[");
    for(i = 0;i<MAX;i++) {
        printf("%d ",intArray[i]);
    }
    printf("]\n");
}
void main() {
    printf("Input Array: ");
    display();
    printline(50);
    int location = find(55);
    if(location != -1)
        printf("\nElement found at location: %d" ,(location+1));
    else
        printf("Element not found.");
}
```

Output:

```
Input Array: [1 2 3 4 6 7 9 11 12 14 15 16 17 19 33 34 43 45 55 66 ]
```

```
=====
Total comparisons made: 19
```

```
Element found at location: 19
```

Task# 3

Design a Program (in any Computer Language of your choice) to implement the Algorithms to implement Binary Search.

Code along with the output screenshot should be submitted.

Code:

```
#include <stdio.h>

#define MAX 20

// array of items on which linear search will be conducted.
int intArray[MAX] =
{1,2,3,4,6,7,9,11,12,14,15,16,17,19,33,34,43,45,55,66};

void printline(int count) {
    int i;

    for(i = 0;i <count-1;i++) {
        printf("=");
    }

    printf("\n");
}

int find(int data) {
    int lowerBound = 0;
    int upperBound = MAX -1;
    int midPoint = -1;
    int comparisons = 0;
    int index = -1;

    while(lowerBound <= upperBound) {
        printf("Comparison %d\n" , (comparisons +1) );
        printf("lowerBound : %d, intArray[%d] =
%d\n", lowerBound, lowerBound,
        intArray[lowerBound]);
        printf("upperBound : %d, intArray[%d] =
%d\n", upperBound, upperBound,
        intArray[upperBound]);
        comparisons++;

        // compute the mid point
        // midPoint = (lowerBound + upperBound) / 2;
        midPoint = lowerBound + (upperBound - lowerBound) / 2;

        // data found
        if(intArray[midPoint] == data) {
            index = midPoint;
        }
    }
}
```

```

        break;
    } else {
        // if data is larger
        if(intArray[midPoint] < data) {
            // data is in upper half
            lowerBound = midPoint + 1;
        }
        // data is smaller
        else {
            // data is in lower half
            upperBound = midPoint - 1;
        }
    }
}
printf("Total comparisons made: %d" , comparisons);
return index;
}

void display() {
    int i;
    printf("[");

    // navigate through all items
    for(i = 0;i<MAX;i++) {
        printf("%d ",intArray[i]);
    }

    printf("]\n");
}

void main() {
    printf("Input Array: ");
    display();
    printline(50);

    //find location of 1
    int location = find(55);

    // if element was found
    if(location != -1)
        printf("\nElement found at location: %d" ,(location+1));
    else
        printf("\nElement not found.");
}

```


Code Screenshot:

```
#include <stdio.h>
#define MAX 20
int intArray[MAX] = {1,2,3,4,6,7,9,11,12,14,15,16,17,19,33,34,43,45,55,66};
void printline(int count) {
    int i;
    for(i = 0; i < count-1; i++) {
        printf("--");
    }
    printf("--\n");
}
int find(int data) {
    int lowerBound = 0;
    int upperBound = MAX - 1;
    int midPoint = -1;
    int comparisons = 0;
    int index = -1;
    while(lowerBound <= upperBound) {
        printf("Comparison %d\n", (comparisons + 1));
        printf("LowerBound : %d, intArray[%d] = %d\n", lowerBound, lowerBound,
            intArray[lowerBound]);
        printf("UpperBound : %d, intArray[%d] = %d\n", upperBound, upperBound,
            intArray[upperBound]);
        comparisons++;
        midPoint = lowerBound + (upperBound - lowerBound) / 2;
        if(intArray[midPoint] == data) {
            index = midPoint;
            break;
        } else {
            if(intArray[midPoint] < data) {
                lowerBound = midPoint + 1;
            }
            else {
                upperBound = midPoint - 1;
            }
        }
    }
    printf("Total comparisons made: %d", comparisons);
    return index;
}

void display() {
    int i;
    printf("[");
    for(i = 0; i < MAX; i++) {
        printf("%d ", intArray[i]);
    }
    printf("]\n");
}

void main() {
    printf("Input Array: ");
    display();
    printline(50);
    int location = find(55);
    if(location != -1)
        printf("\nElement found at location: %d", (location+1));
    else
        printf("\nElement not found.");
}
```

Output:

```
Input Array: [1 2 3 4 6 7 9 11 12 14 15 16 17 19 33 34 43 45 55 66 ]
```

```
-----
```

```
Comparison 1
```

```
lowerBound : 0, intArray[0] = 1
```

```
upperBound : 19, intArray[19] = 66
```

```
Comparison 2
```

```
lowerBound : 10, intArray[10] = 15
```

```
upperBound : 19, intArray[19] = 66
```

```
Comparison 3
```

```
lowerBound : 15, intArray[15] = 34
```

```
upperBound : 19, intArray[19] = 66
```

```
Comparison 4
```

```
lowerBound : 18, intArray[18] = 55
```

```
upperBound : 19, intArray[19] = 66
```

```
Total comparisons made: 4
```

```
Element found at location: 19
```