



Guru Gobind Singh Indraprastha University

University School of Automation & Robotics

Data Structures Lab (ARI254)

Practical File

| | |
|--------------------------|-------------|
| Name | Sujal Singh |
| Enrollment Number | 04119051723 |
| Batch | ILOT-B1-23 |

Index

| No. | Question | Remarks |
|-----|--|---------|
| 1. | <p>Write C programs by using Array data structure for the following problem domains:</p> <ul style="list-style-type: none">(a) Create an array of integer with size n. Return the difference between the largest and the smallest value inside that array.(b) Initializes an array with ten random integers and then prints four lines of output, containing: Every element at an even index, Every odd element, All elements in reverse order, Only the first and last element.(c) Consider an integer array of size 5 and display the following: Sum of all the elements, Sum of alternate elements in the array, and second-highest element in the array. | |
| 2. | <p>Write a program to create a singly linked list of n nodes and perform:</p> <ul style="list-style-type: none">(a) Insertion at the beginning.(b) Insertion at the end.(c) Insertion at a specific location.(d) Deletion at the beginning.(e) Deletion at the end.(f) Deletion At a specific location. | |
| 3. | <p>Write a program to create a doubly linked list of n nodes and perform:</p> <ul style="list-style-type: none">(a) Insertion at the beginning.(b) Insertion at the end.(c) Insertion at a specific location.(d) Deletion at the beginning.(e) Deletion at the end.(f) Deletion At a specific location. | |

| | | |
|-----|---|--|
| 4. | <p>Write a program to create a singly circular and doubly linked list of n nodes and perform:</p> <ul style="list-style-type: none"> (a) Insertion at the beginning. (b) Insertion at the end. (c) Insertion at a specific location. (d) Deletion at the beginning. (e) Deletion at the end. (f) Deletion At a specific location. | |
| 5. | Write a program to implement stack using arrays and linked lists. | |
| 6. | Write a program to reverse a sentence/string using stack. | |
| 7. | Write a program to check for balanced parenthesis in a given expression. | |
| 8. | Write a program to convert infix expression to prefix and postfix expression. | |
| 9. | Write a program to implement Linear Queue using Array and Linked Lists. | |
| 10. | Write a program to implement Circular Queue using Array and Linked Lists. | |
| 11. | Write a program to implement Doubly Ended Queue using Array and Linked Lists. | |
| 12. | Write a Program to implement Binary Search Tree operations. | |
| 13. | Write a program to implement Bubble Sort, Selection Sort, Heap Sort, Quick Sort, Merge Sort and Insertion Sort algorithm. | |
| 14. | <p>Write C Programs by using Graph data structure for the following problem domains:</p> <ul style="list-style-type: none"> (a) Graph Traversal: BFS (b) Graph Traversal: DFS | |

Experiment–1

Aim:

Write C programs by using Array data structure for the following problem domains:

- Create an array of integer with size n . Return the difference between the largest and the smallest value inside that array.
- Initializes an array with ten random integers and then prints four lines of output, containing: Every element at an even index, Every odd element, All elements in reverse order, Only the first and last element.
- Consider an integer array of size 5 and display the following: Sum of all the elements, Sum of alternate elements in the array, and second-highest element in the array.

Program (a):

```

1  #include <stdio.h>
2
3  int main() {
4      int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
5      int n = (int) (sizeof(arr) / sizeof(int));
6
7      int min = 0, max = 0;
8
9      for (int i = 0; i < n; i++) {
10         if (arr[i] < min)
11             min = arr[i];
12         if (arr[i] > max)
13             max = arr[i];
14     }
15
16     printf("Minimum = %d; Maximum = %d; Difference = %d", min, max,
17           ↪ max-min);
18
19     return 0;
20 }
```

Output (a):

```

1  Minimum = 0; Maximum = 10; Difference = 10
```

Program (b):

```

1  #include <stdio.h>
2
3  int main() {
4      int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
5      int n = (int) (sizeof(arr) / sizeof(int));
6
7      printf("Elements at an even index: ");
```

| | |
|--------------------|--|
| 8 | <code>for (int i = 0; i < n; i += 2) {</code> |
| 9 | <code> printf("%d, ", arr[i]);</code> |
| 10 | <code>}</code> |
| 11 | <code>printf("\n");</code> |
| 12 | |
| 13 | <code>printf("Odd numbers: ");</code> |
| 14 | <code>for (int i = 0; i < n; i++) {</code> |
| 15 | <code> if (arr[i] % 2 != 0)</code> |
| 16 | <code> printf("%d, ", arr[i]);</code> |
| 17 | <code>}</code> |
| 18 | <code>printf("\n");</code> |
| 19 | |
| 20 | <code>printf("Reverse order: ");</code> |
| 21 | <code>for (int i = n - 1; i >= 0; i--) {</code> |
| 22 | <code> printf("%d, ", arr[i]);</code> |
| 23 | <code>}</code> |
| 24 | <code>printf("\n");</code> |
| 25 | |
| 26 | <code>printf("First: %d; Last: %d;\n", arr[0], arr[n - 1]);</code> |
| 27 | |
| 28 | <code>return 0;</code> |
| 29 | <code>}</code> |
| Output (b): | |
| 1 | Elements at an even index: 1, 3, 5, 7, 9, |
| 2 | Odd numbers: 1, 3, 5, 7, 9, |
| 3 | Reverse order: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, |
| 4 | First: 1; Last: 10; |