

Lab Exercise 1: Sum of Array Elements

Objective: To write a C program that calculates the sum of all elements in an array.

Problem Statement: Write a C program to find and display the sum of all elements in a given integer array.

Algorithm:

1. Start
2. Read the number of elements n
3. Read n elements into an array arr[]
4. Initialize sum = 0
5. Repeat for i = 0 to n-1:
 - a. sum = sum + arr[i]
7. Print sum
8. Stop

Program Code:

```
#include <stdio.h>

int main() {
    int n, i, sum = 0;
    int arr[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        sum += arr[i];
    }

    printf("Sum of array elements = %d\n", sum);
    return 0;
}
```

Sample Output:

Enter number of elements: 5

Enter 5 elements: 2 4 6 8 10

Sum of array elements = 30

Lab Exercise 2: Maximum Element in an Array

Objective: To find and display the maximum element in an array.

Problem Statement: Write a C program to find the maximum value in a given integer array.

Algorithm:

1. Start
9. Read the number of elements n
10. Read n elements into an array arr[]
11. Initialize max = arr[0]
12. Repeat for i = 1 to n-1:
 13. a. If arr[i] > max, then max = arr[i]
14. Print max
15. Stop

Program Code:

```
#include <stdio.h>
int main() {
    int n, i, max;
    int arr[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    max = arr[0];
    for (i = 1; i < n; i++) {
        if (arr[i] > max)
            max = arr[i];
    }

    printf("Maximum element = %d\n", max);
    return 0;
}
```

Sample Output:

Enter number of elements: 5

Enter 5 elements: 12 45 23 67 34

Maximum element = 67

Lab Exercise 3: Second Maximum Element in an Array

Objective: To find and display the second largest element in an array.

Problem Statement: Write a C program to find the second maximum element in a given integer array.

Algorithm:

1. Start
16. Read the number of elements n
17. Read n elements into arr[]
18. Initialize first = second = smallest value
19. Repeat for i = 0 to n-1:
 20. a. If arr[i] > first, then second = first, first = arr[i]
 21. b. Else if arr[i] > second and arr[i] != first, then second = arr[i]
22. Print second
23. Stop

Program Code:

```
#include <stdio.h>
int main() {
    int n, i, first, second;
    int arr[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    first = second = -99999;
    for (i = 0; i < n; i++) {
        if (arr[i] > first) {
            second = first;
            first = arr[i];
        } else if (arr[i] > second && arr[i] != first) {
            second = arr[i];
        }
    }
    printf("Second maximum element = %d\n", second);
    return 0;
}
```

Sample Output:

```
Enter number of elements: 6
Enter 6 elements: 12 45 23 67 34 56
Second maximum element = 56
```

Lab Exercise 4: Reverse an Array

Objective: To reverse the order of elements in an array.

Problem Statement: Write a C program to reverse and display the elements of an array.

Algorithm:

1. Start
24. Read the number of elements n
25. Read n elements into arr[]
26. Initialize start = 0, end = n - 1
27. Repeat while start < end:
 28. a. Swap arr[start] and arr[end]
 29. b. Increment start, decrement end
30. Print reversed array
31. Stop

Program Code:

```
#include <stdio.h>
int main() {
    int n, i, temp;
    int arr[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < n / 2; i++) {
        temp = arr[i];
        arr[i] = arr[n - i - 1];
        arr[n - i - 1] = temp;
    }
    printf("Reversed array: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
}
```

Sample Output:

Enter number of elements: 5

Enter 5 elements: 1 2 3 4 5

Reversed array: 5 4 3 2 1

Lab Exercise 5: Find Number of Duplicates in an Array

Objective: To count and display the total number of duplicate elements in an array.

Problem Statement: Write a C program to find how many duplicate elements exist in a given integer array.

Algorithm:

1. Start
2. Read the number of elements n
3. Read n elements into arr[]
4. Initialize count = 0
5. Repeat for i = 0 to n-1:
 - a. For j = i+1 to n-1:
 - i. If arr[i] == arr[j], mark arr[j] and increment count
6. Print count
7. Stop

Program Code:

```
#include <stdio.h>
int main() {
    int n, i, j, count = 0;
    int arr[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < n; i++) {
        for (j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                count++;
                break;
            }
        }
    }
    printf("Number of duplicate elements = %d\n", count);
    return 0;
}
```

Sample Output:

Enter number of elements: 6

Enter 6 elements: 2 3 4 3 2 5

Number of duplicate elements = 2

Lab Exercise 6: Frequency of Elements in an Array

1. Start
40. Read the number of elements n
41. Read n elements into arr[]
42. Initialize freq[] = -1
43. Repeat for i = 0 to n-1:
44. a. If freq[i] != 0, set count = 1
45. b. For j = i+1 to n-1:
46. i. If arr[i] == arr[j], increment count and set freq[j] = 0
47. c. If freq[i] != 0, set freq[i] = count
48. Print each element and its frequency
49. Stop

```
#include <stdio.h>
int main() {
    int n, i, j, count;
    int arr[100], freq[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        freq[i] = -1;
    }
    for (i = 0; i < n; i++) {
        if (freq[i] != 0) {
            count = 1;
            for (j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    count++;
                    freq[j] = 0;
                }
            }
            freq[i] = count;
        }
    }
    printf("Element | Frequency\n");
    for (i = 0; i < n; i++) {
        if (freq[i] != 0)
            printf("%d      %d\n", arr[i], freq[i]);
    }
    return 0;
}
```

Sample Output:

Enter number of elements: 7

Enter 7 elements: 2 3 4 2 3 5 2

Element | Frequency

2 3

3 2

4	1
5	1