Experiment 2: Odd or Even in a Set

Aim:

To find whether numbers in a given set are odd or even.

Algorithm:

```
1. Start.
2. Input number of elements n.
3. Read array elements.
4. For each element check:
  - If divisible by 2, print Even.
  - Else, print Odd.
5. Stop.
```

Code:

```
#include <stdio.h>

int main() {
    int n, i, num[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d numbers: ", n);
    for(i = 0; i < n; i++)
        scanf("%d", &num[i]);

    for(i = 0; i < n; i++) {
        if(num[i] % 2 == 0)
            printf("%d is Even\n", num[i]);
        else
            printf("%d is Odd\n", num[i]);
    }
    return 0;
}
```

Sample Output:

```
Enter number of elements: 5
Enter 5 numbers: 10 15 22 33 40
10 is Even
15 is Odd
22 is Even
33 is Odd
40 is Even
```

Result:

The program correctly checks odd/even for all numbers.

Experiment 2: Odd or Even in a Set

Aim:

To find whether numbers in a given set are odd or even.

Algorithm:

```
1. Start.
2. Input number of elements n.
3. Read array elements.
4. For each element check:
   - If divisible by 2, print Even.
   - Else, print Odd.
5. Stop.
```

Code:

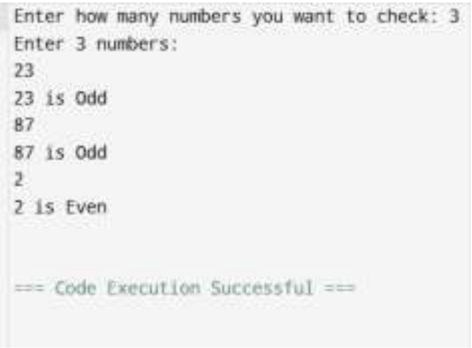
```
#include <stdio.h>

int main() {
    int n, i, num[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d numbers: ", n);
    for(i = 0; i < n; i++)
        scanf("%d", &num[i]);

    for(i = 0; i < n; i++) {
        if(num[i] % 2 == 0)
            printf("%d is Even\n", num[i]);
        else
            printf("%d is Odd\n", num[i]);
    }

    return 0;
}
```

Sample Output:



```
Enter how many numbers you want to check: 3
Enter 3 numbers:
23
23 is Odd
87
87 is Odd
2
2 is Even


=== Code Execution Successful ===
```

Result:

The program correctly checks odd/even for all numbers.