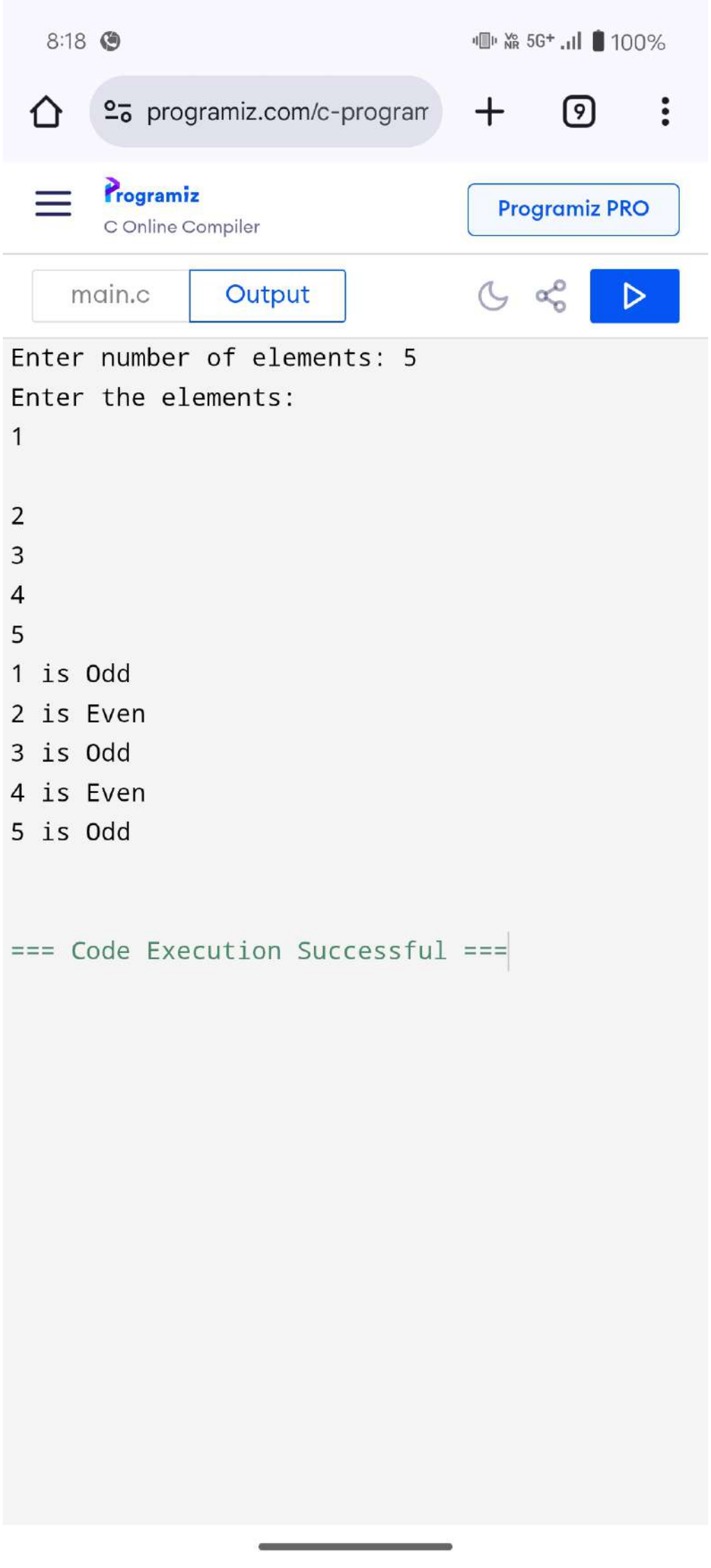
9:37 90%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
Enter rows and columns for first matrix: 2
3
Enter rows and columns for second matrix: 3
2
Enter elements of first matrix:
1
2
3
4
5
6
Enter elements of second matrix:
7
8
9
10
11
12
Resultant matrix:
58 64
139 154

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


8:18
100%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter number of elements: 5
Enter the elements:
1
2
3
4
5
1 is Odd
2 is Even
3 is Odd
4 is Even
5 is Odd
=== Code Execution Successful ===


8:19
100%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter a number: 5
Factorial of 5 is 120
=== Code Execution Successful ===

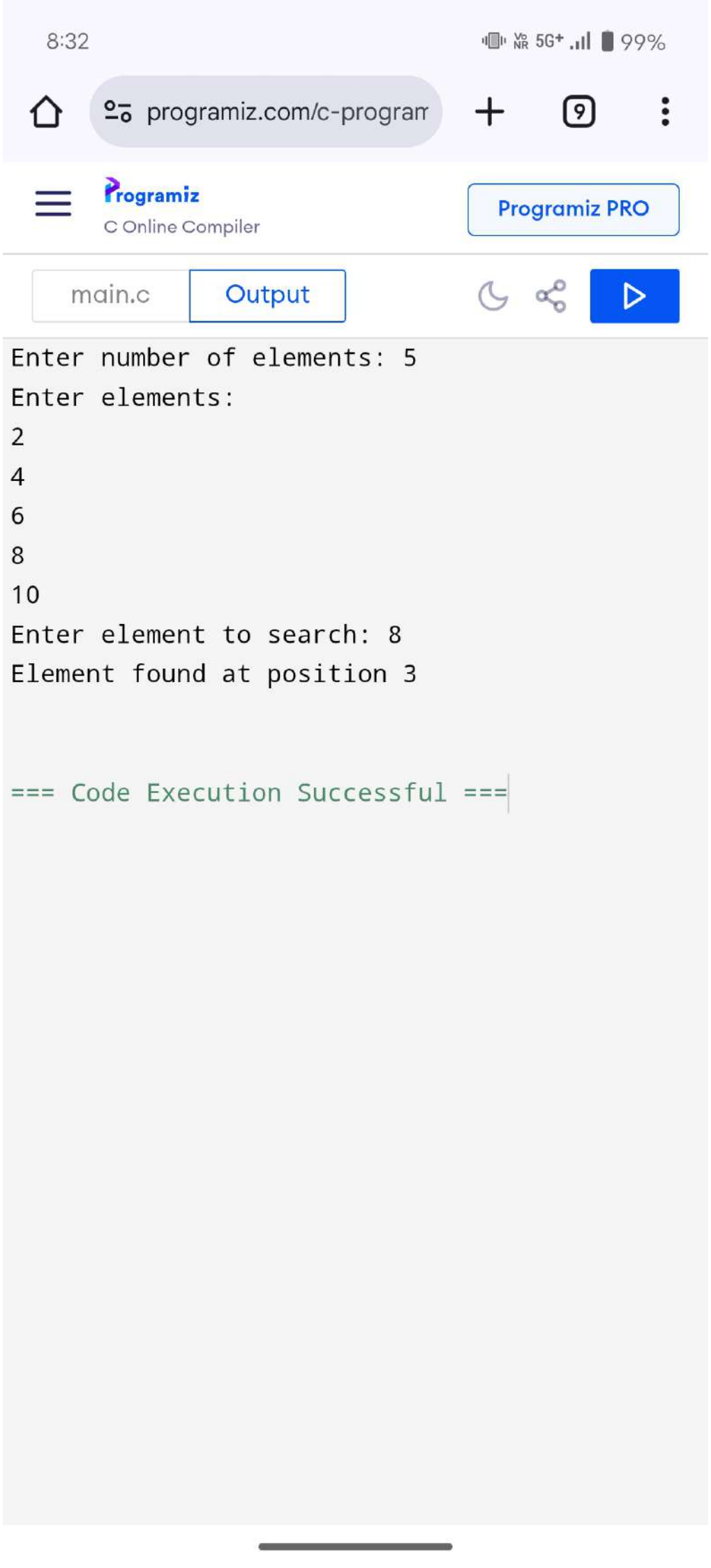

8:20
100%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter number of terms: 5
Fibonacci Series: 0 1 1 2 3
=== Code Execution Successful ===

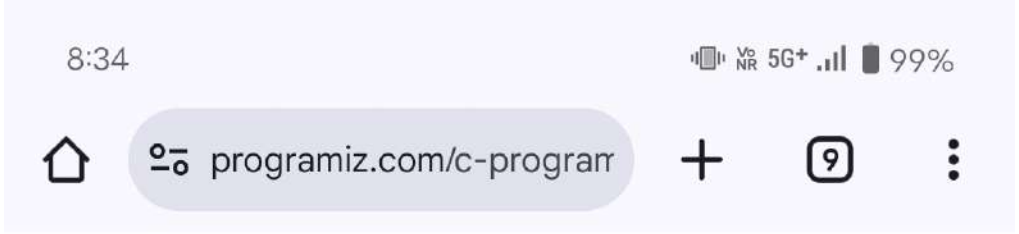

8:21
100%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter a number: 5
Factorial of 5 is 120
=== Code Execution Successful ===


8:22
100%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter number of terms: 5
Fibonacci Series: 0 1 1 2 3
=== Code Execution Successful ===

8:27 100%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
Enter number of elements: 3
Enter elements:
10
20
30

1.Insert
2.Delete
3.Display
4.Exit
Enter choice: 1
Enter position and value to insert: 1
15

1.Insert
2.Delete
3.Display
4.Exit
Enter choice: 3
Array: 10 15 20 30

1.Insert
2.Delete
3.Display
4.Exit
Enter choice: 4


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


8:32
99%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter number of elements: 5
Enter elements:
2
4
6
8
10
Enter element to search: 8
Element found at position 3
=== Code Execution Successful ===




```
Enter number of elements: 5
Enter sorted elements:
1
3
5
7
9
Enter element to search: 7
Element found at position 3


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

8:37 98%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
1.Insert
2.Delete
3.Display
4.Exit
Enter choice: 1
Enter value to insert: 10

1.Insert
2.Delete
3.Display
4.Exit
Enter choice: 1
Enter value to insert: 20

1.Insert
2.Delete
3.Display
4.Exit
Enter choice: 3
List: 20 10

1.Insert
2.Delete
3.Display
4.Exit
Enter choice: 4


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

8:39 98%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
3.Peek
4.Display
5.Exit
Enter choice: 4
Stack: 5

1.Push
2.Pop
3.Peek
4.Display
5.Exit
Enter choice: 2
Popped: 5

1.Push
2.Pop
3.Peek
4.Display
5.Exit
Enter choice: 3
Stack is empty

1.Push
2.Pop
3.Peek
4.Display
5.Exit
Enter choice: 5


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

8:43

programiz.com/c-programming/online-compiler/

Programiz
C Online
Compiler

One tidy tax file?
Too easy.

Start free trial

Adobe
Acrobat Pro

Programiz
PRO ❯

main.c

Run

Output

Clear

```
5  int top = -1;
6  void push(char c) { stack[++top] = c; }
7  char pop() { return stack[top--]; }
8  int precedence(char c) {
9  if (c == '^') return 3;
10 if (c == '*' || c == '/') return 2;
11 if (c == '+' || c == '-') return 1;
```

```
Enter infix expression: 5+8*9
Postfix expression: 589*+

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

8:45

97%

programiz.com/c-programming/online-compiler/

Programiz
C Online
Compiler

Click Here for More Information

Ad by Sponsor

See More

Prog
PRO

main.c

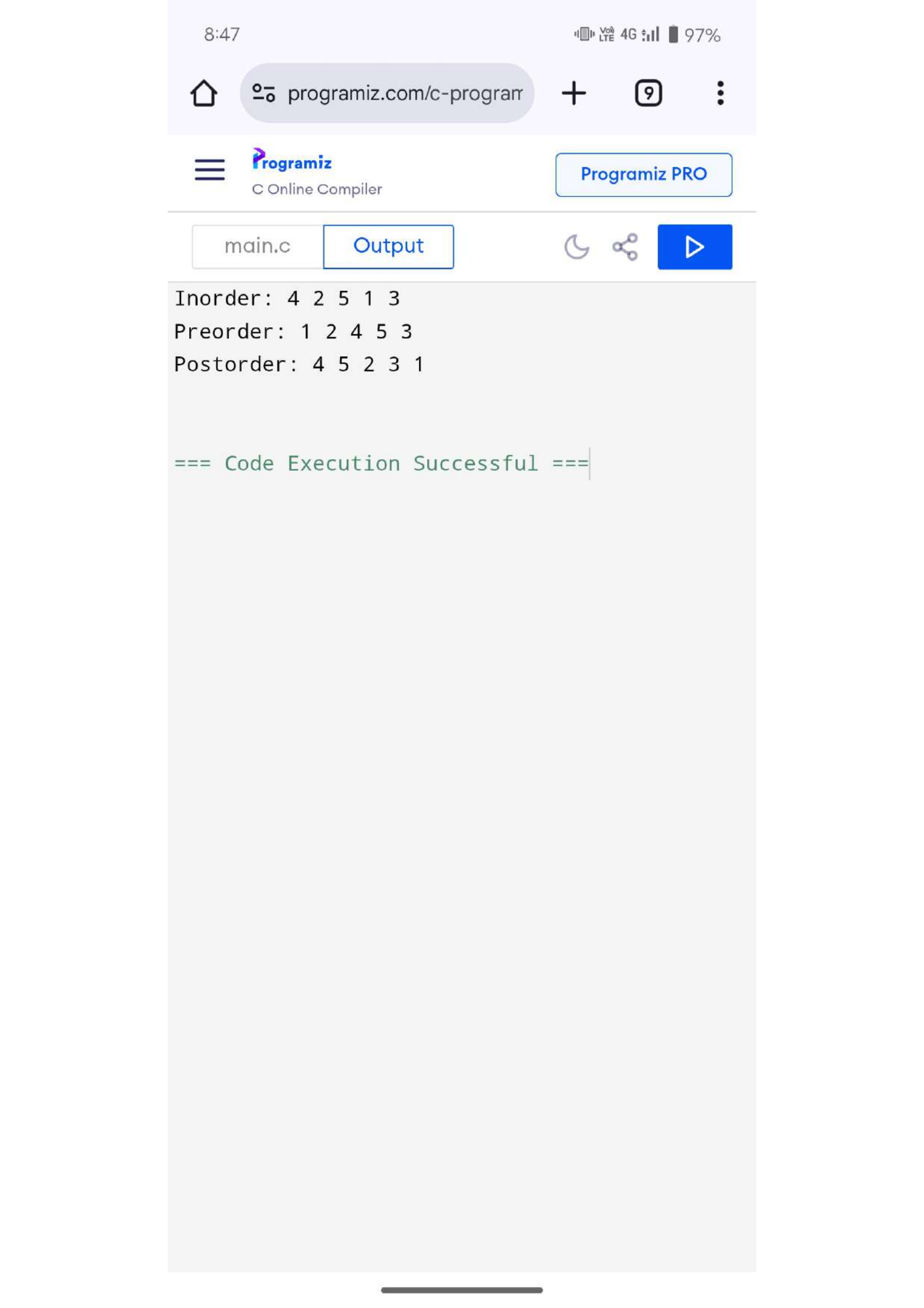
Run

Output

Clea

```
#include <stdio.h>

#define MAX 100

int queue[MAX], front = -1, rear = -1;
void enqueue(int val) {
if (rear == MAX - 1)
printf("Queue Overflow\n");
```

```
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter choice: 1
Enter value to enqueue: 7
```

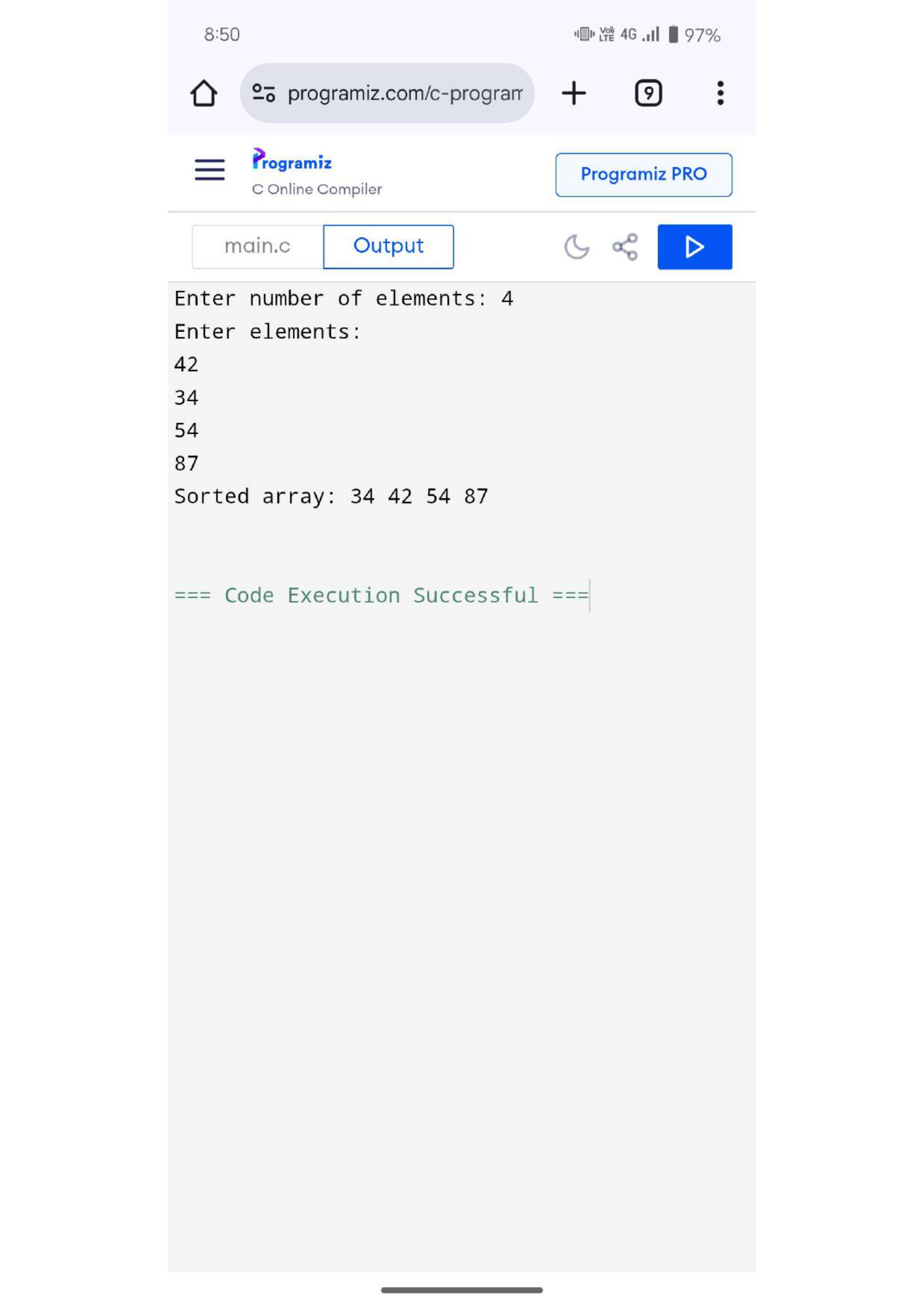

8:47
97%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Inorder: 4 2 5 1 3
Preorder: 1 2 4 5 3
Postorder: 4 5 2 3 1
=== Code Execution Successful ===

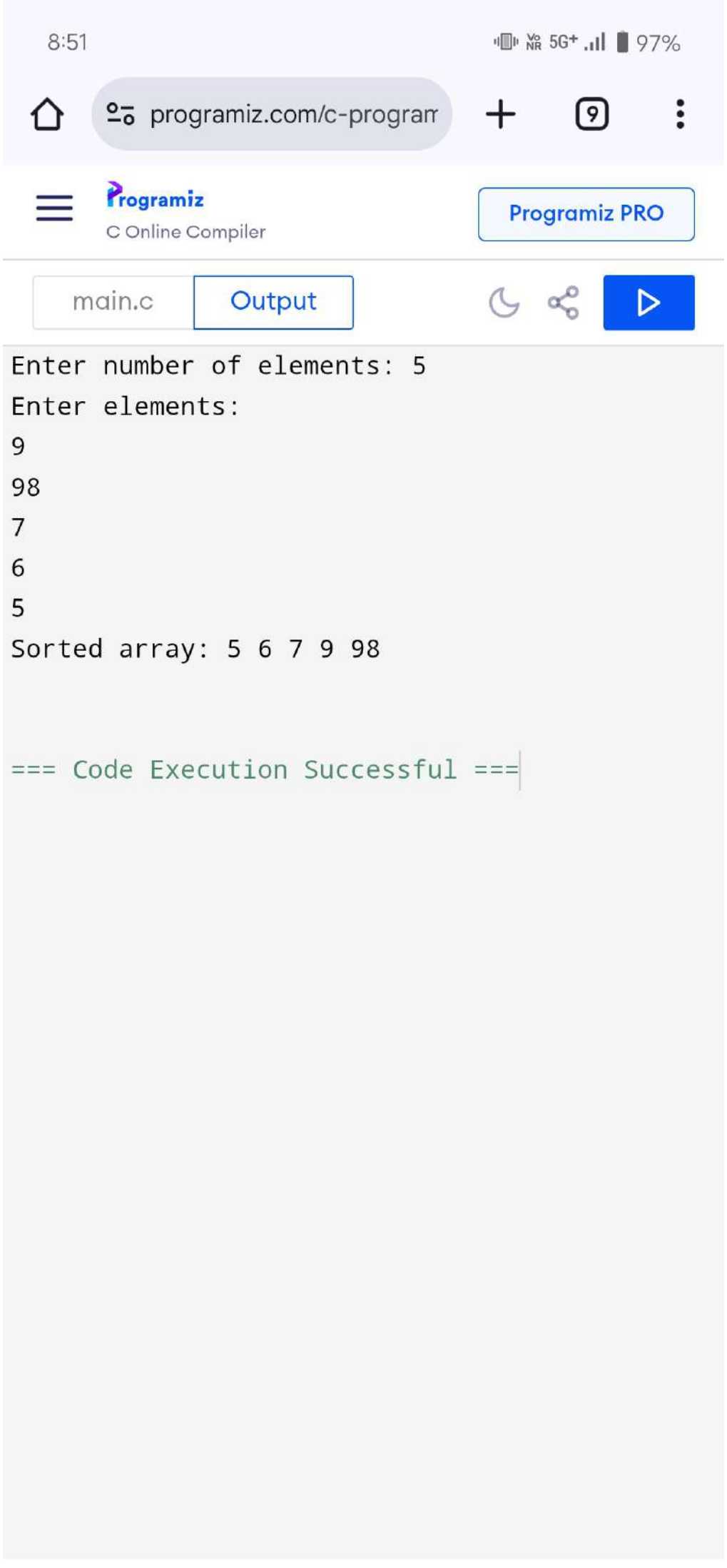
8:48 4G 97%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

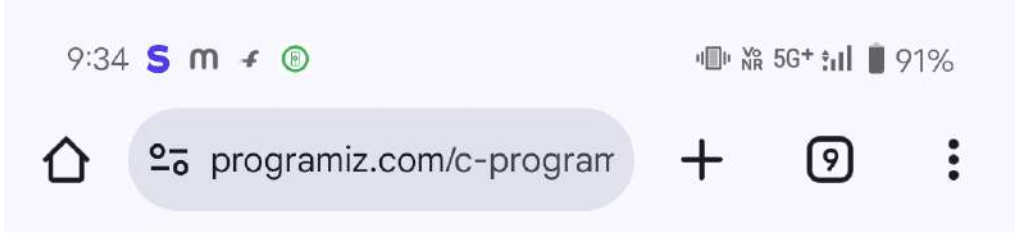
```
Enter number of elements: 5
Enter elements:
12

22
32
42
52
Hash Table:
0: -1
1: -1
2: 12
3: 22
4: 32
5: 42
6: 52
7: -1
8: -1
9: -1


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

8:49 4G 97%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
Enter number of elements: 5
Enter elements:
5
3

4
2
1
Sorted array: 1 2 3 4 5


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


8:50
97%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter number of elements: 4
Enter elements:
42
34
54
87
Sorted array: 34 42 54 87
=== Code Execution Successful ===

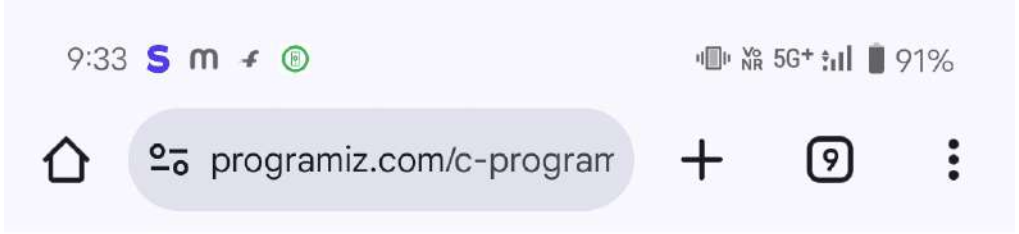

8:51
97%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Enter number of elements: 5
Enter elements:
9
98
7
6
5
Sorted array: 5 6 7 9 98
=== Code Execution Successful ===



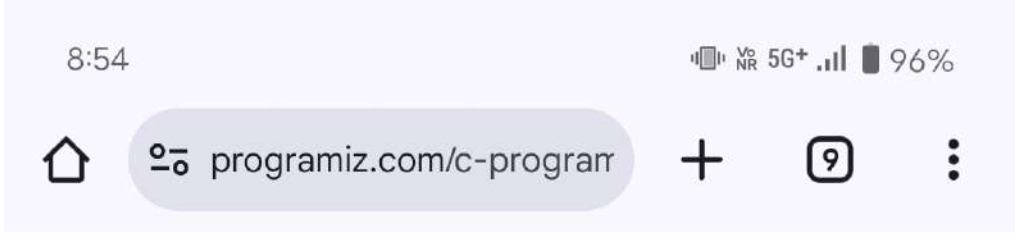

```
Enter number of elements: 5
Enter elements:
5
3
4
2
1
Sorted array: 1 2 3 4 5


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

9:01 95%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
Inorder traversal of the constructed AVL tree
    is
10 20 25 30 40 50
Inorder traversal after deletion of 20
10 25 30 40 50

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




```
Enter number of vertices: 4
Enter number of edges: 4
Enter edge (v1 v2): 0
1
Enter edge (v1 v2): 0
2
Enter edge (v1 v2): 1
2
Enter edge (v1 v2): 2

1
Enter starting vertex: 0
BFS traversal: 0 1 2


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




```
Enter number of vertices: 4
Enter number of edges: 4
Enter edge (v1 v2): 1
2
Enter edge (v1 v2): 1
3
Enter edge (v1 v2): 2
3
Enter edge (v1 v2): 3
2
Enter starting vertex: 0
DFS traversal: 0


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

8:56 96%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
Enter number of vertices: 2
Enter adjacency matrix:
1
3
5
7
Enter source vertex: 0
Shortest distances from source 0:
0: 0
1: 3


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

8:57 96%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output

```
Enter number of vertices:
3
Enter adjacency matrix:
0
1
2

3

3
4
5
6
7
Edges in MST:
0 - 1 : 1
0 - 2 : 2
Total cost: 3


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


8:59
96%
programiz.com/c-program
Programiz
C Online Compiler
Programiz PRO
main.c
Output
Edges in the MST:
2 -- 3 == 4
0 -- 3 == 5
0 -- 1 == 10
=== Code Execution Successful ===