

main.c



Share

Run

Output

Clear

```
1 #include <stdio.h>
2 int binarySearch(int array[], int size, int target) {
3     int left = 0;
4     int right = size - 1;
5     while (left <= right) {
6         int mid = left + (right - left) / 2;
7         if (array[mid] == target) return mid;
8         if (array[mid] < target) left = mid + 1;
9         else right = mid - 1;
10    }
11    return -1;
12 }
13 int main() {
14     int array[] = {2, 3, 4, 10, 40};
15     int target = 10;
16     int size = sizeof(array) / sizeof(array[0]);
17     int result = binarySearch(array, size, target);
18     if (result == -1) {
19         printf("Element is not present in array\n");
20     } else {
21         printf("Element is present at index %d\n", result);
22     }
23     return 0;
24 }
```

/tmp/kIDFyFI5Yy.o
Element is present at index 3

=== Code Execution Successful ===

main.c



Share

Run

Output

Clear

```
1 #include <stdio.h>
2 int main()
3 {
4     int arr[] = {12, 34, 54, 1, 3, 69, 14, 11};
5     int n = sizeof(arr) / sizeof(arr[0]);
6     int max, min;
7     max = min = arr[0];
8     for (int i = 1; i < n; i++)
9     {
10         if (arr[i] > max)
11         {
12             max = arr[i];
13         }
14         if (arr[i] < min)
15         {
16             min = arr[i];
17         }
18     }
19     printf("Maximum element: %d\n", max);
20     printf("Minimum element: %d\n", min);
21     return 0;
22 }
```

/tmp/mRS2XdAj2Y.o

Maximum element: 69

Minimum element: 1

=== Code Execution Successful ===

main.c



Share

Run

Output

Clear

```
1 #include <stdio.h>
2 int main()
3 {
4     int arr[] = {1, 2, 3, 4, 2, 5, 6, 7, 8};
5     int n = sizeof(arr) / sizeof(arr[0]);
6     for (int i = 0; i < n; i++)
7     {
8         for (int j = i + 1; j < n; j++)
9         {
10             if (arr[i] == arr[j])
11             {
12                 printf("Duplicate element found: %d\n", arr[i]);
13             }
14         }
15     }
16     return 0;
17 }
```

/tmp/5RmeA56TiI.o
Duplicate element found: 2

=== Code Execution Successful ===

main.c

Share Run

Output

Clear

```
1 #include <stdio.h>
2 void printArray(int arr[], int n) {
3     for (int i = 0; i < n; i++) {
4         printf("%d ", arr[i]);
5     }
6     printf("\n");
7 }
8 int main() {
9     int arr[5] = {1, 2, 3, 4, 5};
10    int n = 5;
11    printf("Original array: ");
12    printArray(arr, n);
13    int element = 6;
14    arr[n++] = element;
15    printf("Array after insertion: ");
16    printArray(arr, n);
17    int index = 2;
18    for (int i = index; i < n - 1; i++) {
19        arr[i] = arr[i + 1];
20    }
21    n--;
22    printf("Array after deletion: ");
23    printArray(arr, n);
24    return 0;
25 }
```

```
/tmp/g7SDQI8xop.o
Original array: 1 2 3 4 5
Array after insertion: 1 2 3 4 5 6
Array after deletion: 1 2 4 5 6

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



main.c



Share

Run

```
1 #include <stdio.h>
2 int fibonacci(int n)
3 {
4     if (n == 0 || n == 1) return n;
5     return fibonacci(n-1) + fibonacci(n-2);
6 }
7 int main()
8 {
9     int n;
10    printf("Enter the number of terms: ");
11    scanf("%d", &n);
12    int sum = 0;
13    printf("Fibonacci series: ");
14    for (int i = 0; i < n; i++)
15    {
16        int fib = fibonacci(i);
17        printf("%d ", fib);
18        sum += fib;
19    }
20    printf("\nSum of the series: %d\n", sum);
21    return 0;
22 }
```

Output

Clear

```
/tmp/ZbFzLXSh90.o
Enter the number of terms: 10
Fibonacci series: 0 1 1 2 3 5 8 13 21 34
Sum of the series: 88
```

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

Activate Windows
Go to Settings to activate Windows.



main.c



Share

Run

Output

Clear

```
1 #include <stdio.h>
2 void linearSearch(int arr[], int n, int target) {
3     int i;
4     for (i = 0; i < n; i++) {
5         if (arr[i] == target) {
6             printf("Element %d found at index %d\n", target, i);
7             return;
8         }
9     }
10    printf("Element %d not found in the array\n", target);
11 }
12 int main() {
13     int arr[] = {1, 2, 3, 4, 5};
14     int n = sizeof(arr) / sizeof(arr[0]);
15     int target = 3;
16     linearSearch(arr, n, target);
17     return 0;
18 }
```

/tmp/fqmAJ6RM1V.o
Element 3 found at index 2

=== Code Execution Successful ===

main.c



Share

Run

Output

Clear

```
1 #include <stdio.h>
2 int factorial(int n)
3 {
4     if (n == 0)
5     {
6         return 1;
7     } else
8     {
9         return n * factorial(n - 1);
10    }
11 }
12 int main()
13 {
14     int num;
15     printf("Enter a number: ");
16     scanf("%d", &num);
17     int result = factorial(num);
18     printf("Factorial of %d is %d\n", num, result);
19     return 0;
20 }
```

/tmp/UKXbAUXtr3.o
Enter a number: 5
Factorial of 5 is 120

=== Code Execution Successful ===