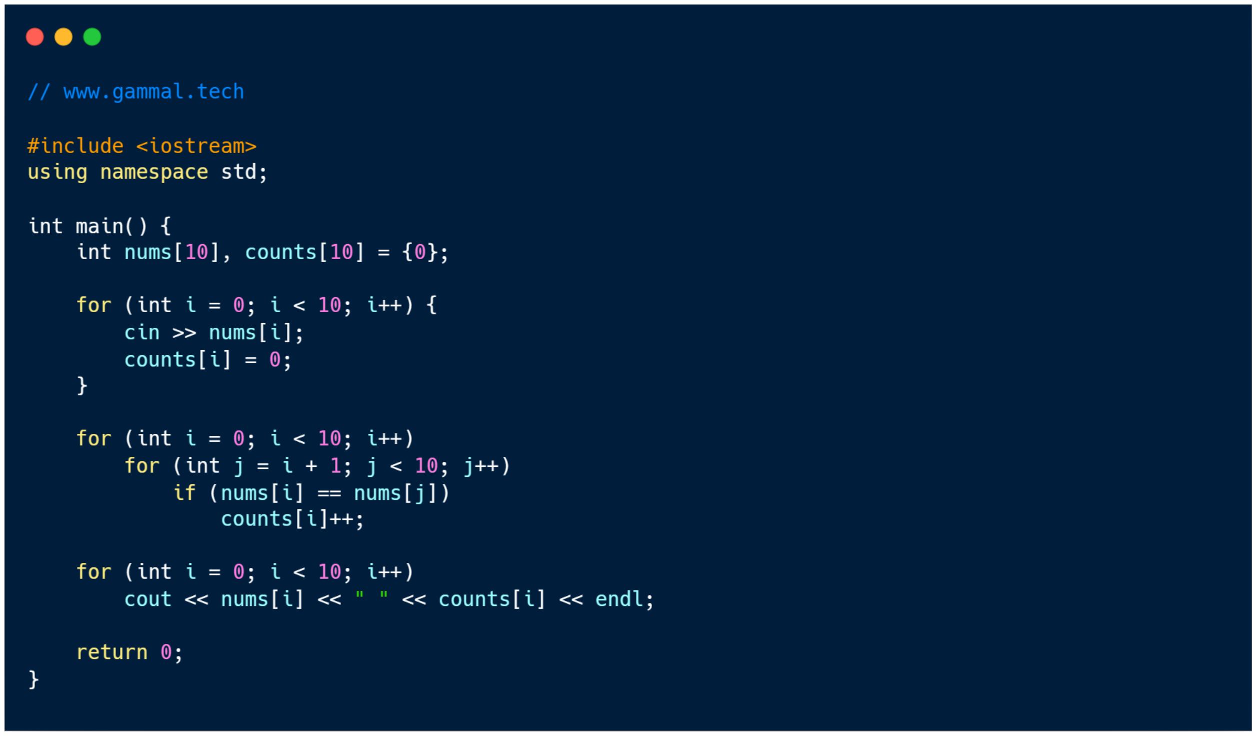
1- Trace the following program and predict the output.

Input

```
1 2 3 1 1 5 4 8 9 1
```



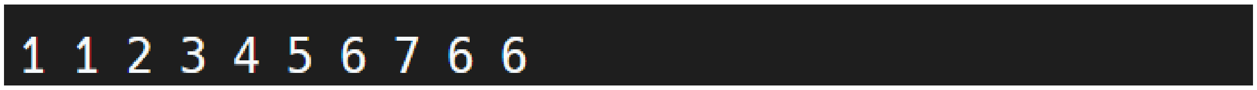
```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[10], counts[10] = {0};

    for (int i = 0; i < 10; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 10; i++)
        for (int j = i + 1; j < 10; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 10; i++)
        cout << nums[i] << " " << counts[i] << endl;

    return 0;
}
```

Solution

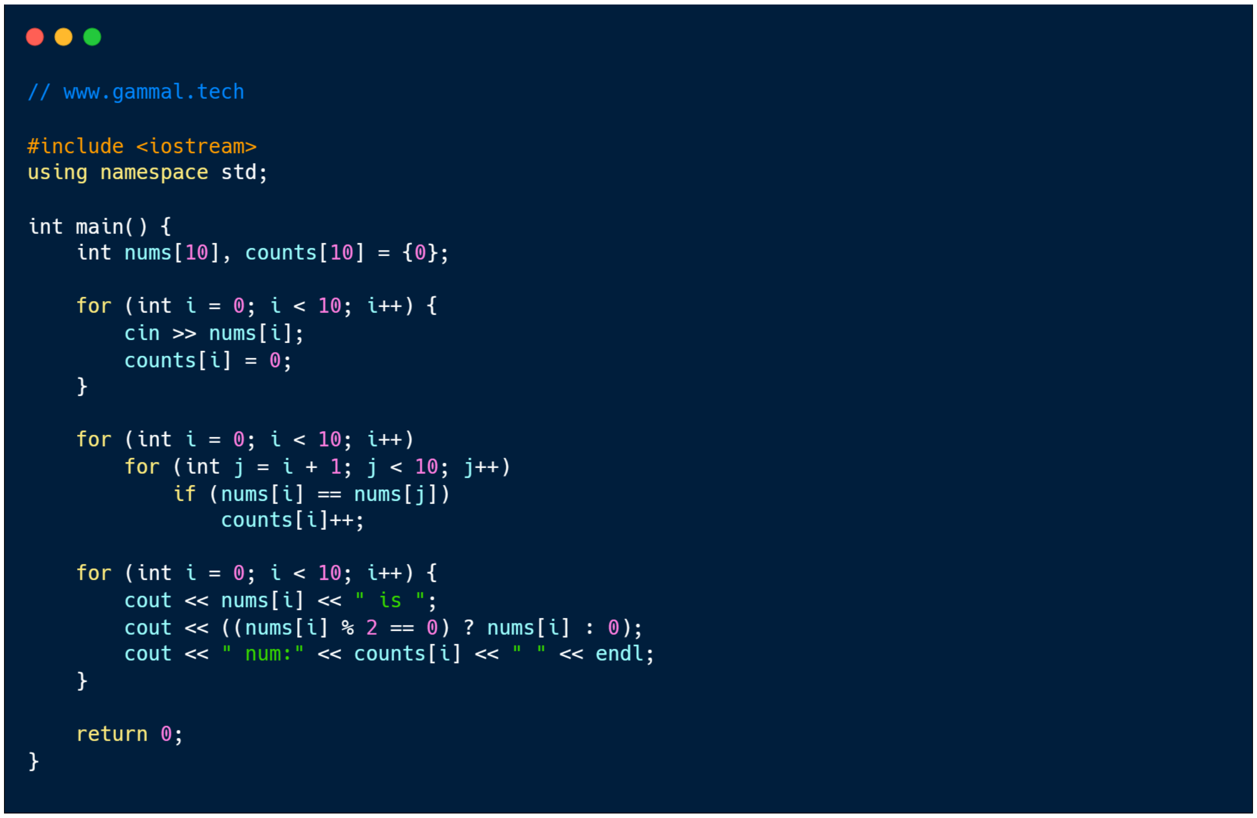
```
1 3
2 0
3 0
1 2
1 1
5 0
4 0
8 0
9 0
1 0
```

2- Trace the following program and predict the output.

Input

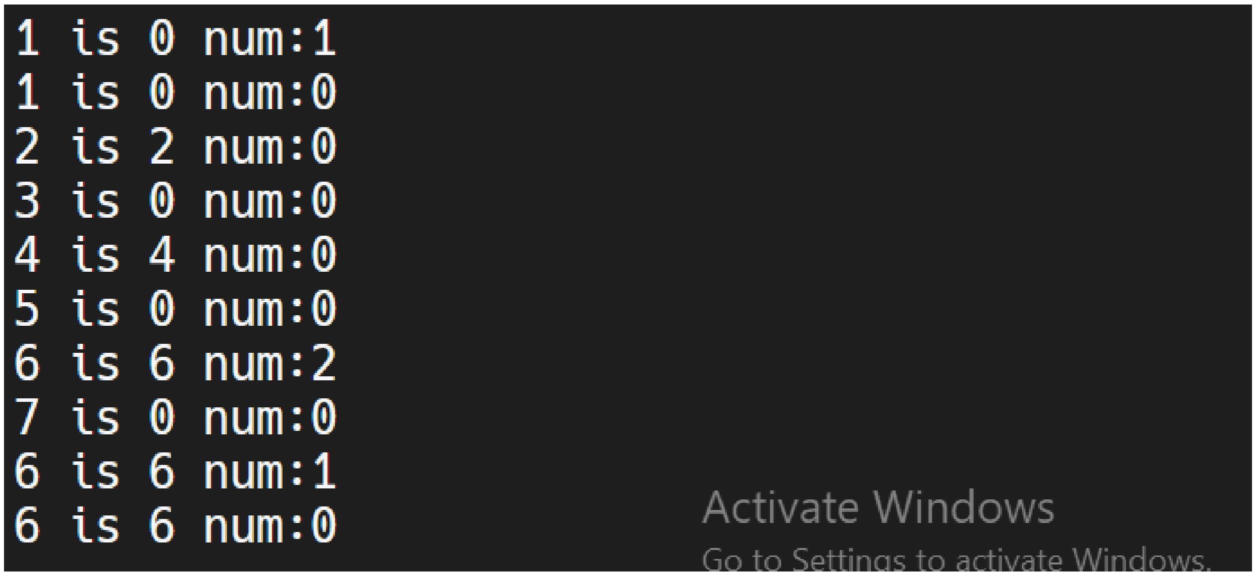


```
1 1 2 3 4 5 6 7 6 6
```



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[10], counts[10] = {0};

    for (int i = 0; i < 10; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 10; i++)
        for (int j = i + 1; j < 10; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 10; i++) {
        cout << nums[i] << " is ";
        cout << ((nums[i] % 2 == 0) ? nums[i] : 0);
        cout << " num:" << counts[i] << " " << endl;
    }

    return 0;
}
```

Solution

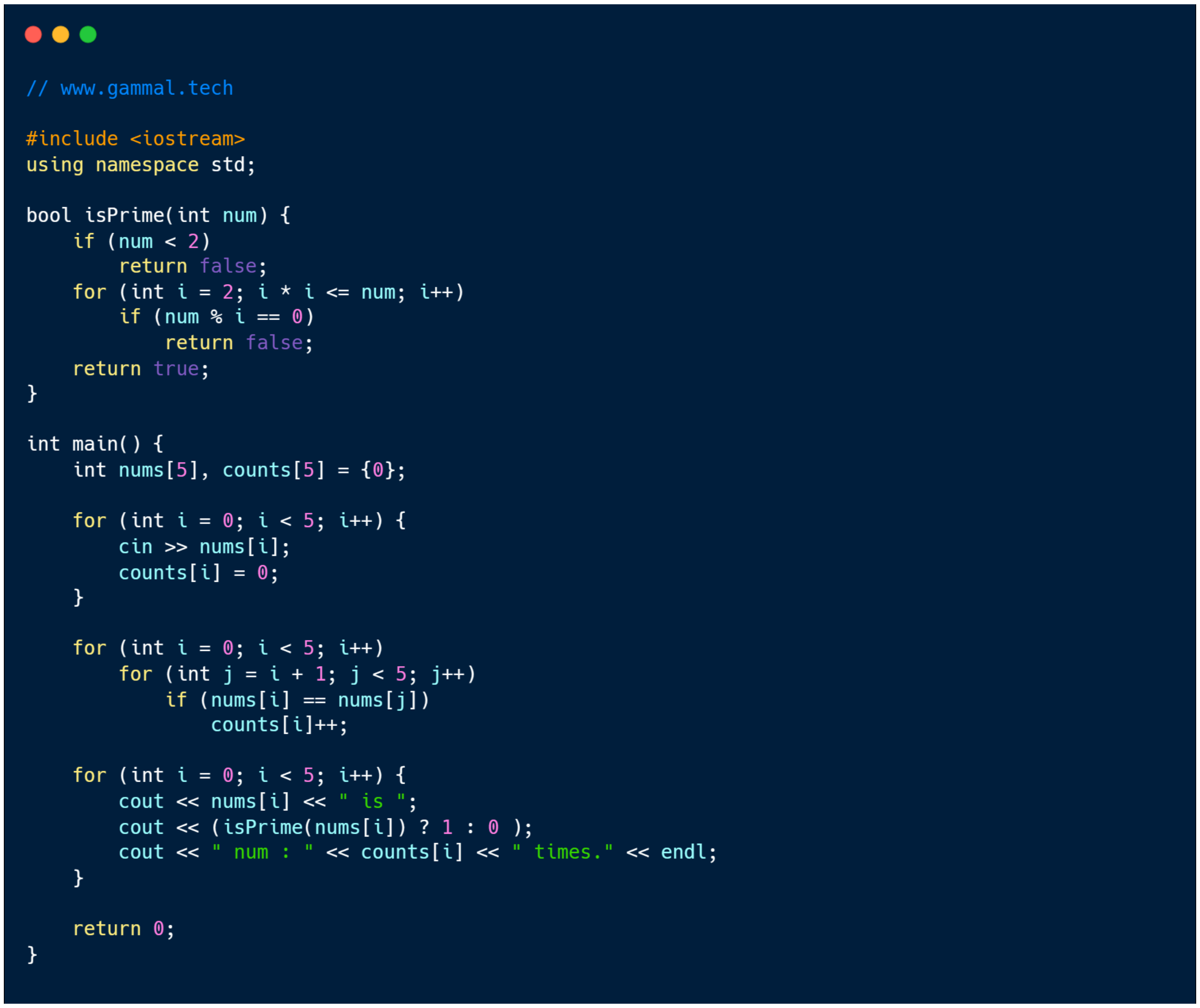


```
1 is 0 num:1
1 is 0 num:0
2 is 2 num:0
3 is 0 num:0
4 is 4 num:0
5 is 0 num:0
6 is 6 num:2
7 is 0 num:0
6 is 6 num:1
6 is 6 num:0
```

3- Trace the following program and predict the output.

Input

```
1 1 2 3 5
```



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

bool isPrime(int num) {
    if (num < 2)
        return false;
    for (int i = 2; i * i <= num; i++)
        if (num % i == 0)
            return false;
    return true;
}

int main() {
    int nums[5], counts[5] = {0};

    for (int i = 0; i < 5; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 5; i++)
        for (int j = i + 1; j < 5; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 5; i++) {
        cout << nums[i] << " is ";
        cout << (isPrime(nums[i]) ? 1 : 0 );
        cout << " num : " << counts[i] << " times." << endl;
    }

    return 0;
}
```

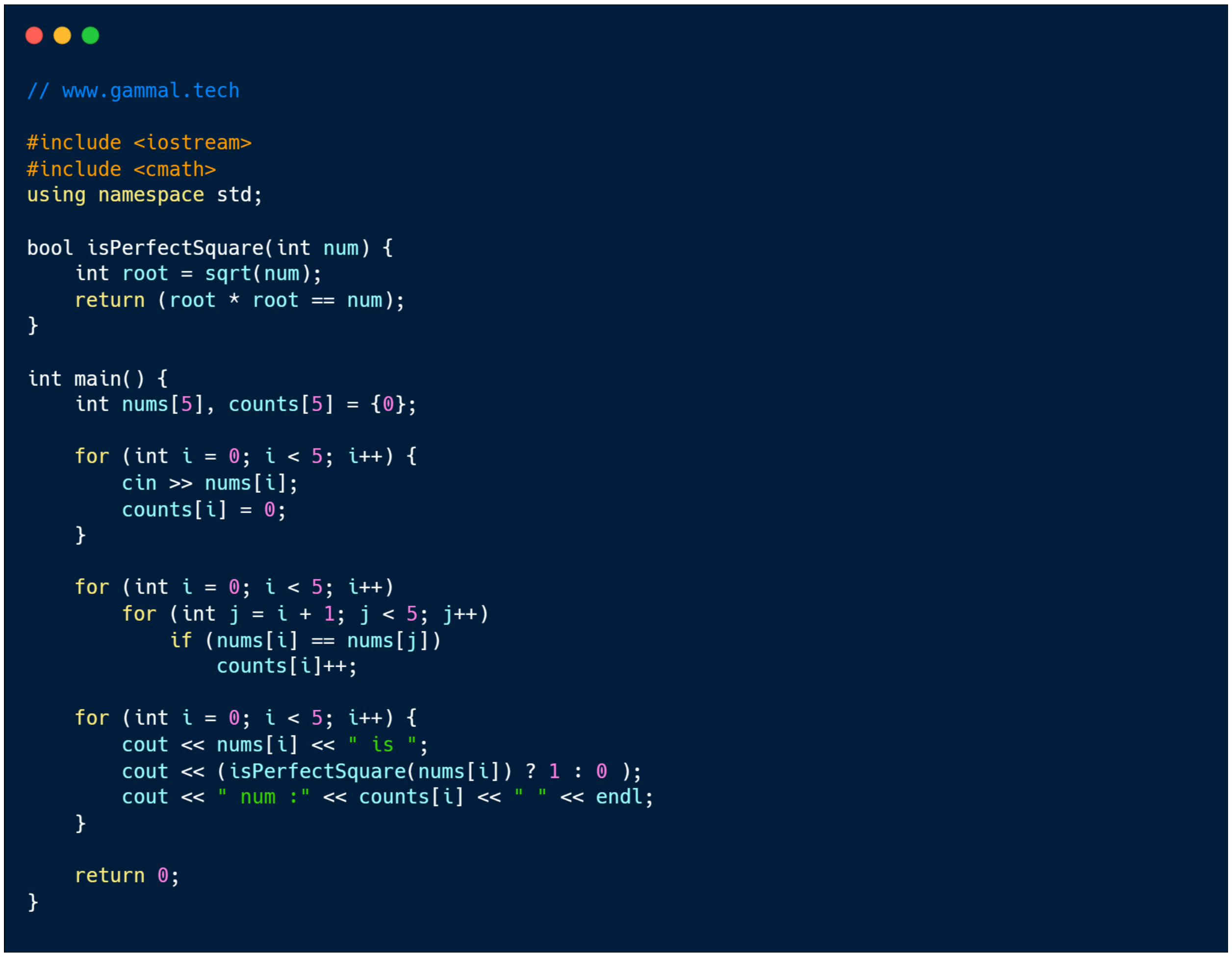
Solution

```
1 is 0 num : 1 times.
1 is 0 num : 0 times.
2 is 1 num : 0 times.
3 is 1 num : 0 times.
5 is 1 num : 0 times.
```

4- Trace the following program and predict the output.

Input

```
4 5 16 16 8
```



```cpp
// www.gammal.tech

#include <iostream>
#include <cmath>
using namespace std;

bool isPerfectSquare(int num) {
    int root = sqrt(num);
    return (root * root == num);
}

int main() {
    int nums[5], counts[5] = {0};

    for (int i = 0; i < 5; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 5; i++)
        for (int j = i + 1; j < 5; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 5; i++) {
        cout << nums[i] << " is ";
        cout << (isPerfectSquare(nums[i]) ? 1 : 0 );
        cout << " num :" << counts[i] << " " << endl;
    }

    return 0;
}
```

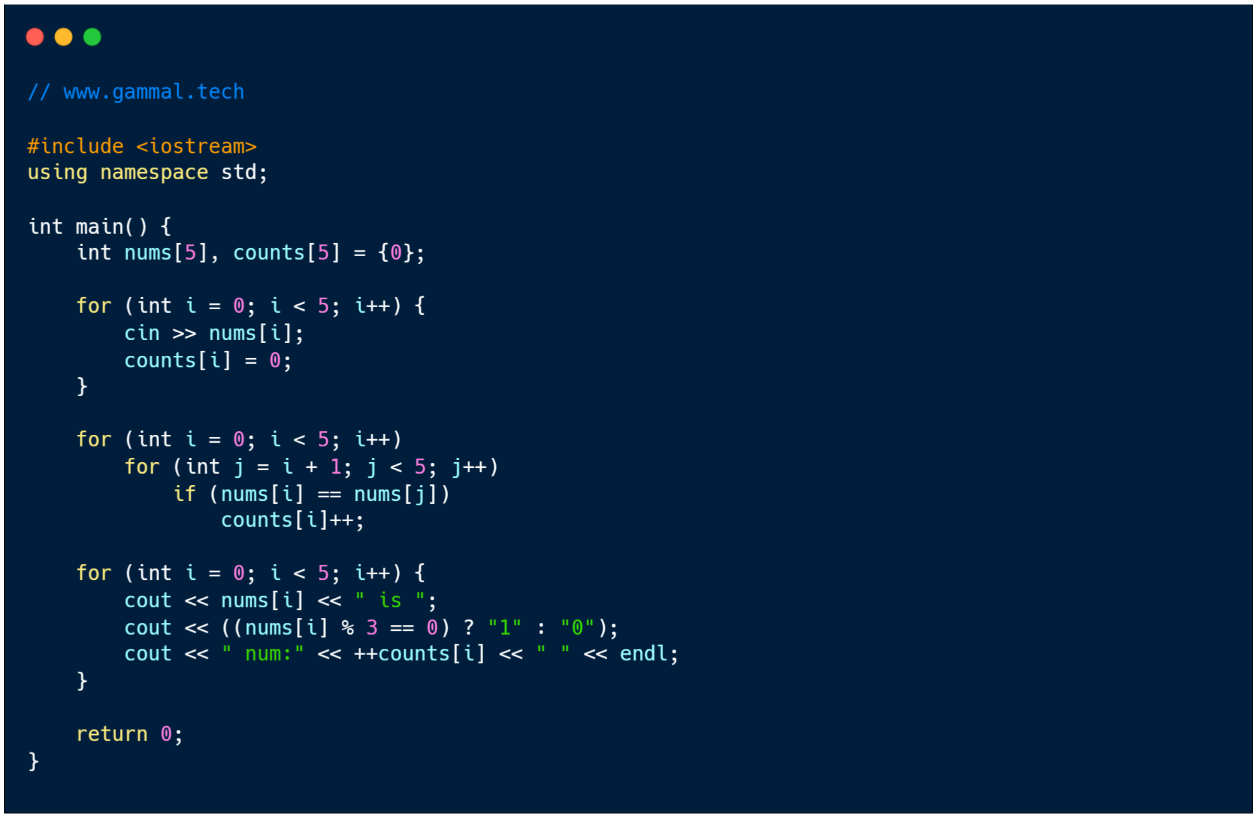
Solution

```
4 is 1 num :0
5 is 0 num :0
16 is 1 num :1
16 is 1 num :0
8 is 0 num :0
```

5- Trace the following program and predict the output.

Input

```
21 15 3 1 3
```



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[5], counts[5] = {0};

    for (int i = 0; i < 5; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 5; i++)
        for (int j = i + 1; j < 5; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 5; i++) {
        cout << nums[i] << " is ";
        cout << ((nums[i] % 3 == 0) ? "1" : "0");
        cout << " num:" << ++counts[i] << " " << endl;
    }

    return 0;
}
```

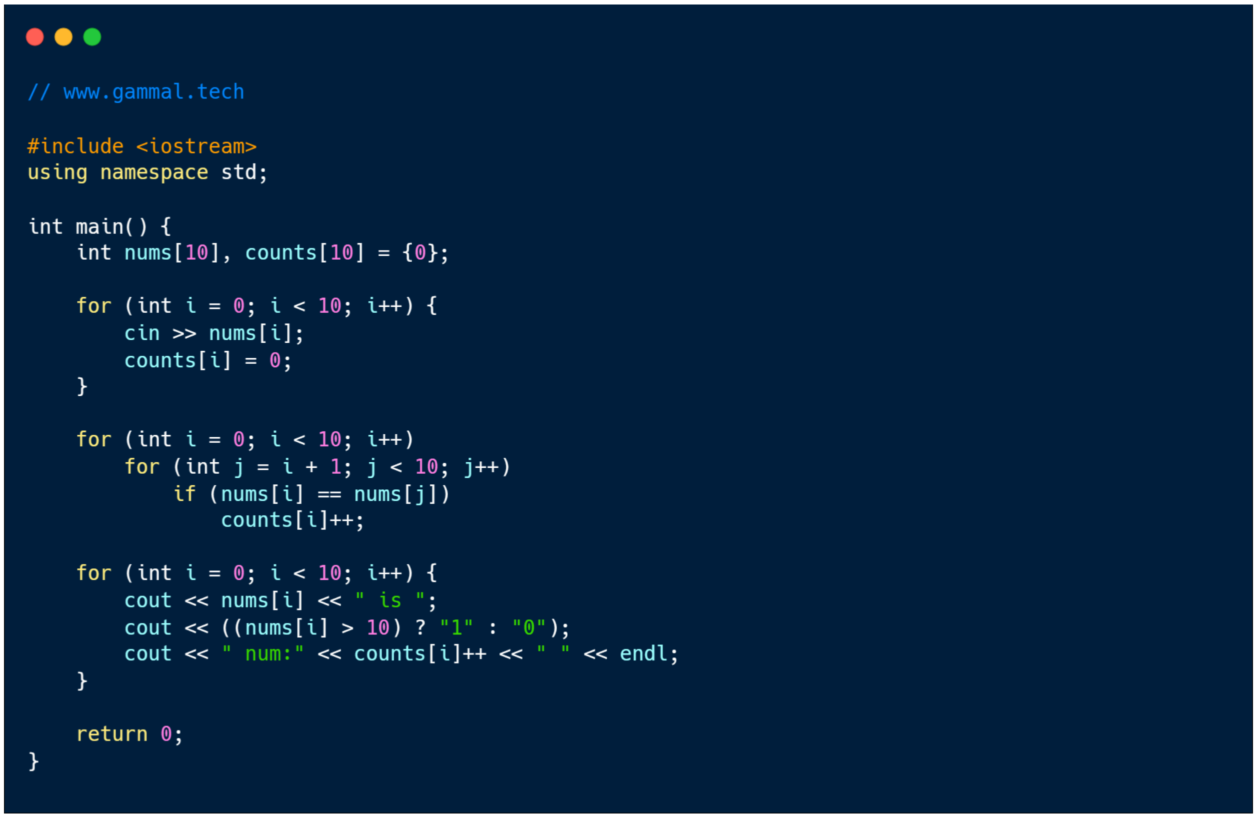
Solution

```
21 is 1 num:1
15 is 1 num:1
3 is 1 num:2
1 is 0 num:1
3 is 1 num:1
```

6- Trace the following program and predict the output.

Input

```
1 2 10 11 12 3 6 8 9 11
```



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[10], counts[10] = {0};

    for (int i = 0; i < 10; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 10; i++)
        for (int j = i + 1; j < 10; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 10; i++) {
        cout << nums[i] << " is ";
        cout << ((nums[i] > 10) ? "1" : "0");
        cout << " num:" << counts[i]++ << " " << endl;
    }

    return 0;
}
```

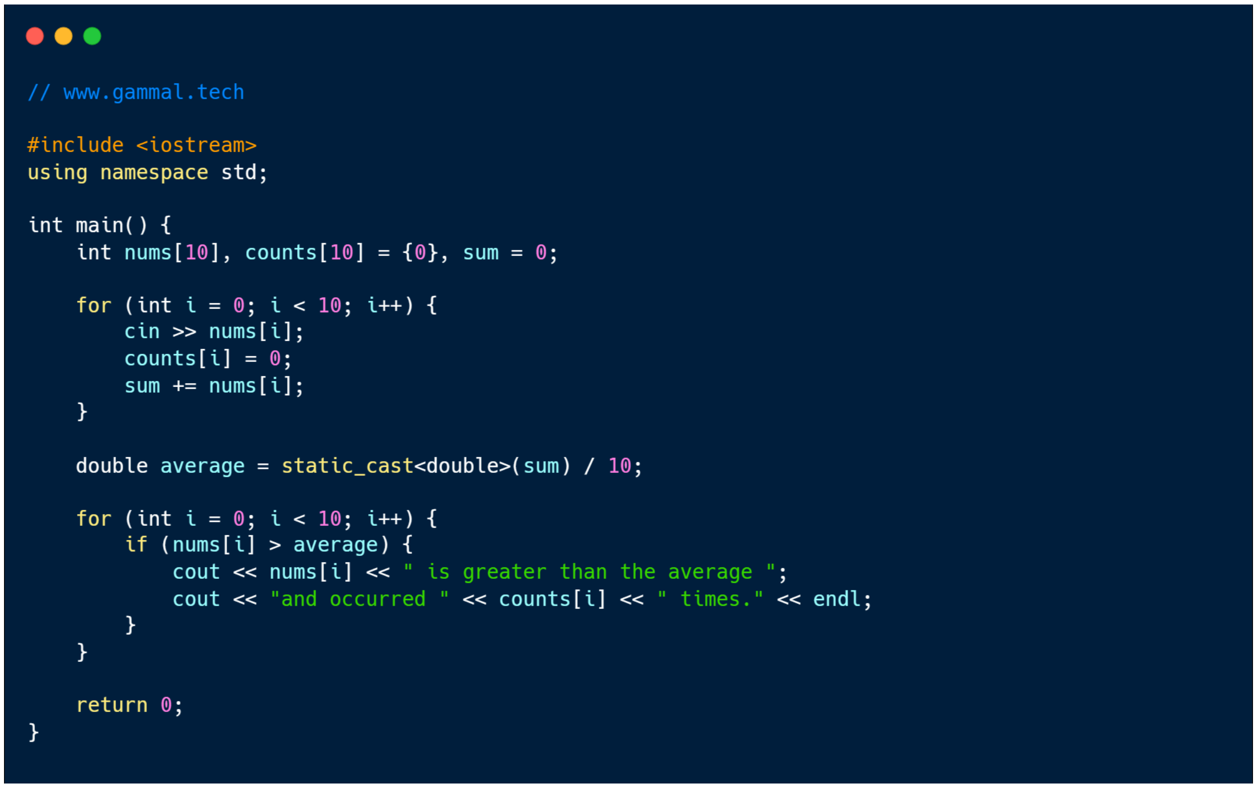
Solution

```
1 is 0 num:0
2 is 0 num:0
10 is 0 num:0
11 is 1 num:1
12 is 1 num:0
3 is 0 num:0
6 is 0 num:0
8 is 0 num:0
9 is 0 num:0
11 is 1 num:0
```

7- Trace the following program and predict the output.

Input

```
1 2 3 4 5 6 7 8 9 10
```



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[10], counts[10] = {0}, sum = 0;

    for (int i = 0; i < 10; i++) {
        cin >> nums[i];
        counts[i] = 0;
        sum += nums[i];
    }

    double average = static_cast<double>(sum) / 10;

    for (int i = 0; i < 10; i++) {
        if (nums[i] > average) {
            cout << nums[i] << " is greater than the average ";
            cout << "and occurred " << counts[i] << " times." << endl;
        }
    }

    return 0;
}
```

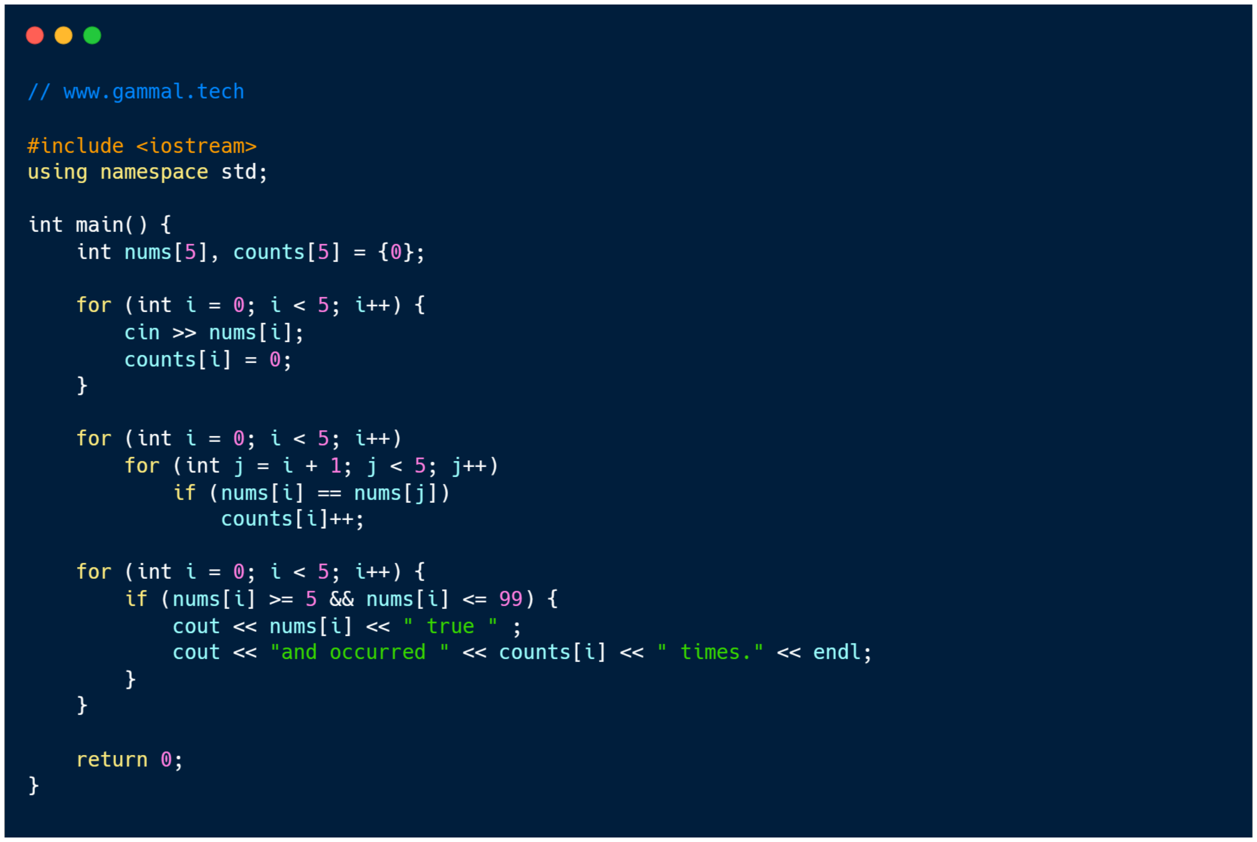
Solution

```
6 is greater than the average and occurred 0 times.
7 is greater than the average and occurred 0 times.
8 is greater than the average and occurred 0 times.
9 is greater than the average and occurred 0 times.
10 is greater than the average and occurred 0 times.
```

8- Trace the following program and predict the output.

Input

```
1 33 599 5 33
```



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[5], counts[5] = {0};

    for (int i = 0; i < 5; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 5; i++)
        for (int j = i + 1; j < 5; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 5; i++) {
        if (nums[i] >= 5 && nums[i] <= 99) {
            cout << nums[i] << " true " ;
            cout << "and occurred " << counts[i] << " times." << endl;
        }
    }

    return 0;
}
```

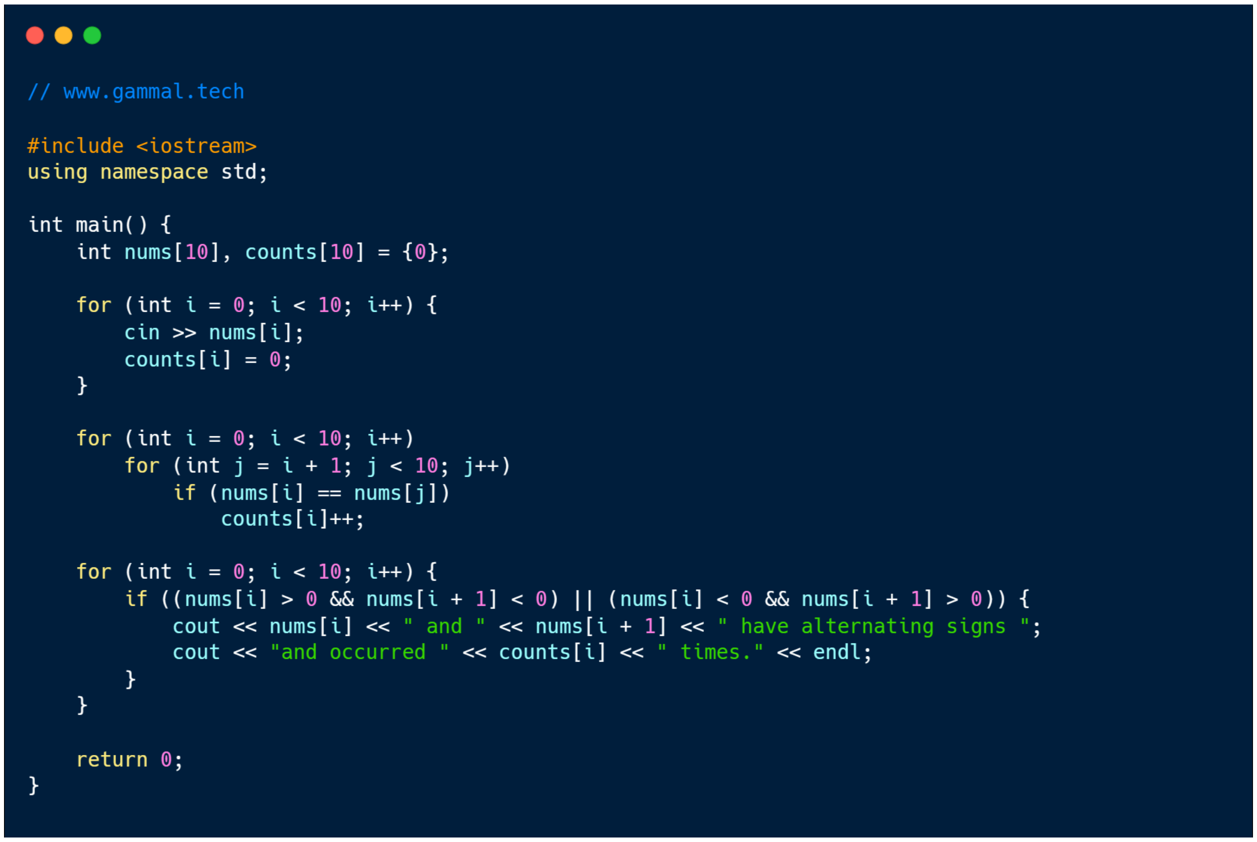
Solution

```
33 true and occurred 1 times.
5 true and occurred 0 times.
33 true and occurred 0 times.
```

9- Trace the following program and predict the output.

Input

```
1 -3 55 6 44 -5 95 -7 3 2
```



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[10], counts[10] = {0};

    for (int i = 0; i < 10; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 10; i++)
        for (int j = i + 1; j < 10; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 10; i++) {
        if ((nums[i] > 0 && nums[i + 1] < 0) || (nums[i] < 0 && nums[i + 1] > 0)) {
            cout << nums[i] << " and " << nums[i + 1] << " have alternating signs ";
            cout << "and occurred " << counts[i] << " times." << endl;
        }
    }

    return 0;
}
```

Solution

```
1 and -3 have alternating signs and occurred 0 times.
-3 and 55 have alternating signs and occurred 0 times.
44 and -5 have alternating signs and occurred 0 times.
-5 and 95 have alternating signs and occurred 0 times.
95 and -7 have alternating signs and occurred 0 times.
-7 and 3 have alternating signs and occurred 0 times.
```

10- Trace the following program and predict the output.

Input

```
1 55 35 7 8 9 10 7 5 3
```

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int nums[10], counts[10] = {0};

    for (int i = 0; i < 10; i++) {
        cin >> nums[i];
        counts[i] = 0;
    }

    for (int i = 0; i < 10; i++)
        for (int j = i + 1; j < 10; j++)
            if (nums[i] == nums[j])
                counts[i]++;

    for (int i = 0; i < 10; i++) {
        if (nums[i] % 5 == 0 && nums[i] % 7 == 0) {
            cout << nums[i] << " is divisible by 5 and 7 ";
            cout << "and occurred " << counts[i] << " times." << endl;
        }
    }

    return 0;
}
```

Solution

```
35 is divisible by 5 and 7 and occurred 0 times.
```