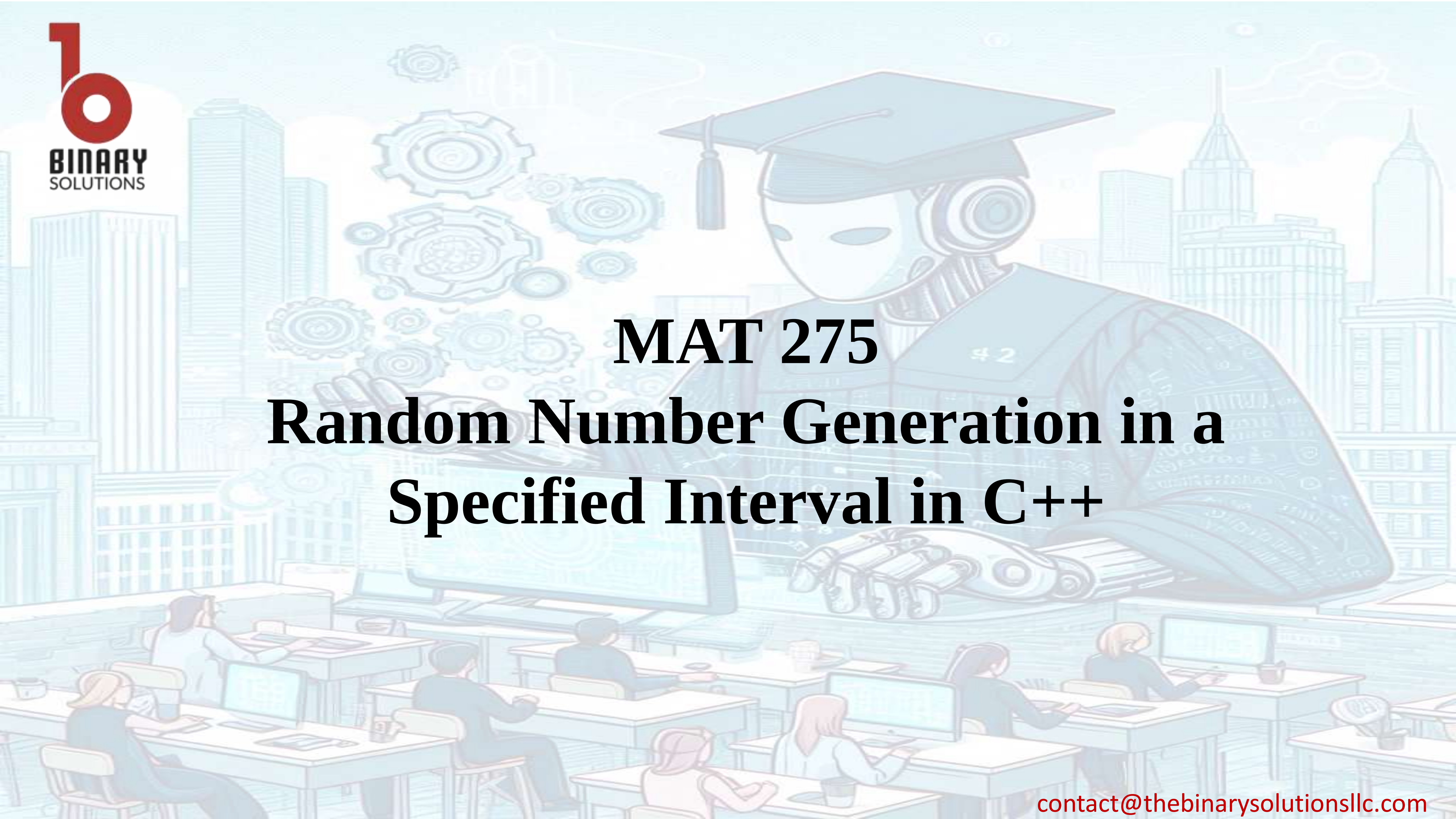


BINARY
SOLUTIONS

MAT 275
Random Number Generation in a Specified Interval in C++

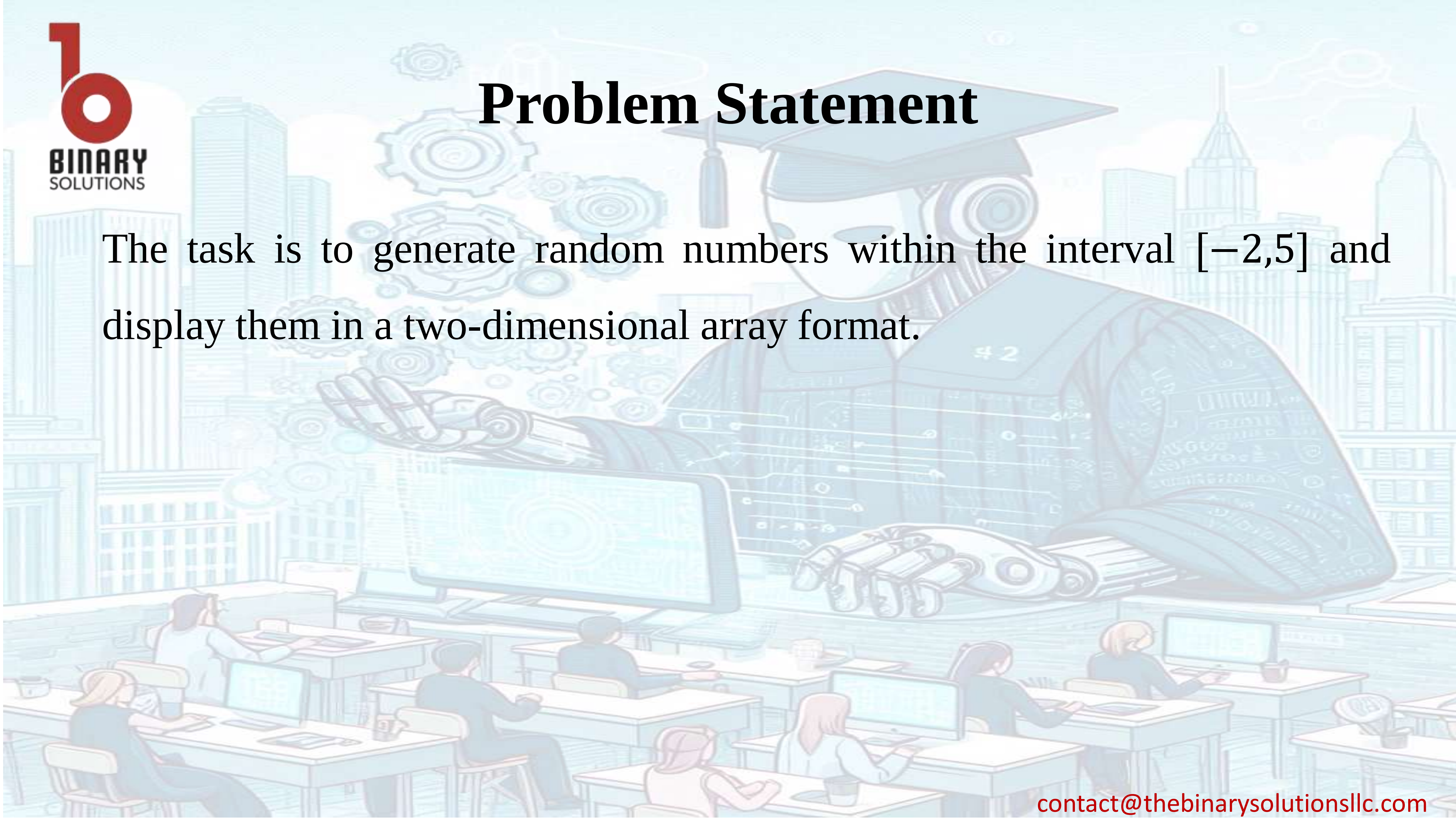
contact@thebinarysolutionsllc.com

Introduction

This program generates random numbers within a given interval [−2,5] and stores them in a two-dimensional array. It demonstrates the usage of the ‘rand()’ function in C++ to generate random numbers and the concept of seeding the random number generator.

Problem Statement

The task is to generate random numbers within the interval $[-2,5]$ and display them in a two-dimensional array format.



Solution Steps

- Define the interval $[-2,5]$ as variables ‘a’ and ‘b’.
- Seed the random number generator using the ‘srand()’ function with the current time as the seed.
- Generate random numbers within the interval $[-2,5]$ and store them in a two-dimensional array ‘r’.
- Display the generated random numbers in the specified format.

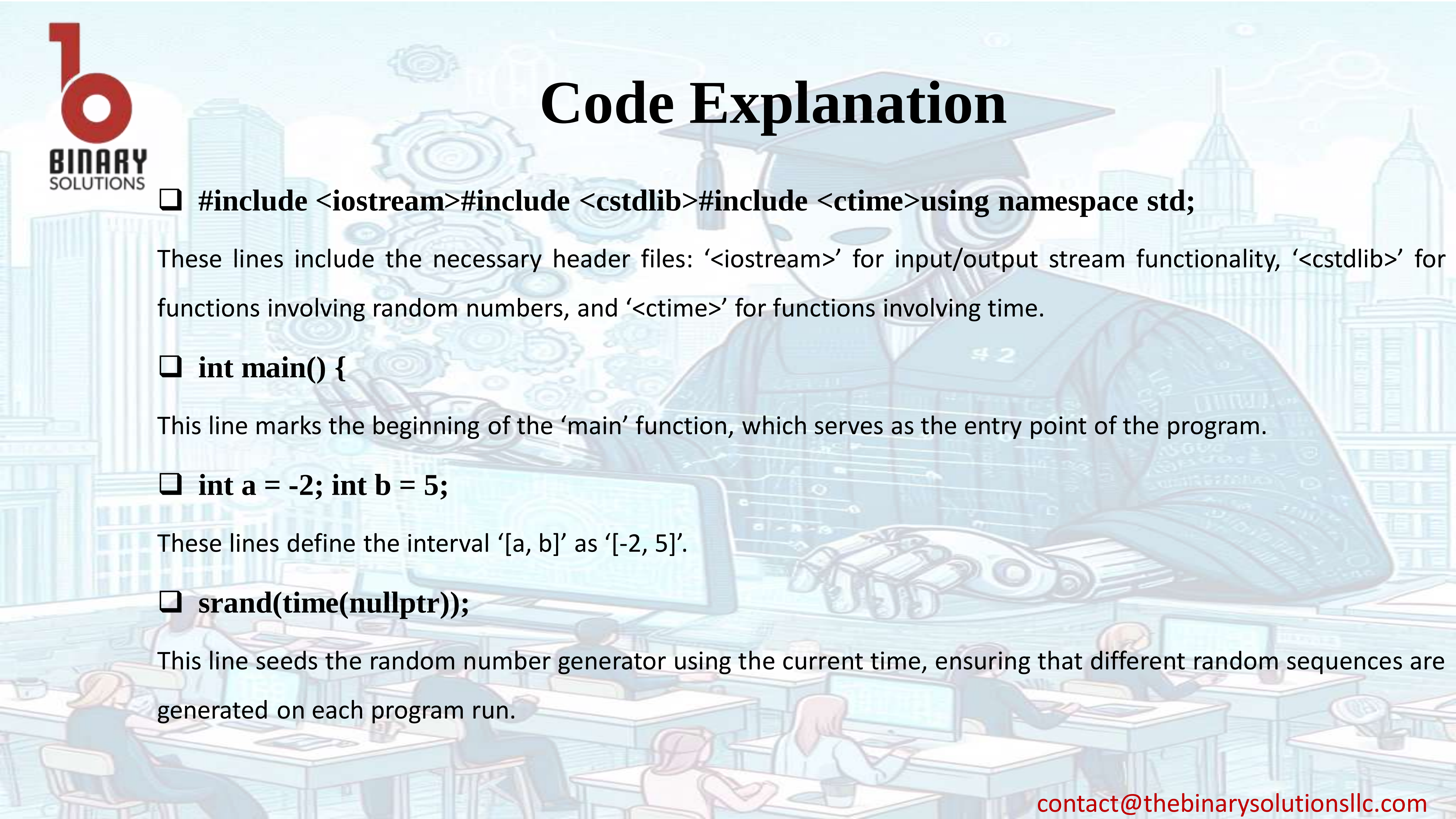
Pseudo Code

```
❑ Include necessary header files (iostream, cstdlib, ctime).
❑ Begin main function.
    ▪ Define interval [a, b] as [-2, 5]:
    o     Declare and initialize variables 'a' and 'b' as integers with values -2 and 5, respectively.
    ▪ Seed the random number generator:
    Use the current time as a seed to initialize the random number generator.
    o Generate random numbers in the interval [a, b]:
    o     Define constants 'rows' and 'cols' as 8 and 5, respectively.
    o     Declare a 2D array 'r' of integers with dimensions 'rows' by 'cols'.
    o     Iterate over each element in the array 'r':
    o         Generate a random number in the interval [a, b] and store it in the current element.
    ▪ Display the generated random numbers:
    o     Output a label "Generated Random Numbers:" to the console.
    o     Iterate over each row in the array 'r':
    o         Iterate over each element in the current row:
    o             Output the value of the current element followed by a space.
    o         Output a newline to move to the next row.
    ▪ End main function
```

BINARY
SOLUTIONS

C ++ Code

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main() {
int a = -2;
    int b = 5;
srand(time(nullptr));
    const int rows = 8;
    const int cols = 5;
    int r[rows][cols];
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            r[i][j] = a + rand() % (b - a + 1); }
    }
cout << "Generated Random Numbers:" << endl;
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            cout << r[i][j] << " ";
        }
        cout << endl }
    return 0;}
```



Code Explanation

- □ **#include <iostream>#include <cstdlib>#include <ctime>using namespace std;**

These lines include the necessary header files: ‘<iostream>’ for input/output stream functionality, ‘<cstdlib>’ for functions involving random numbers, and ‘<ctime>’ for functions involving time.

- □ **int main() {**

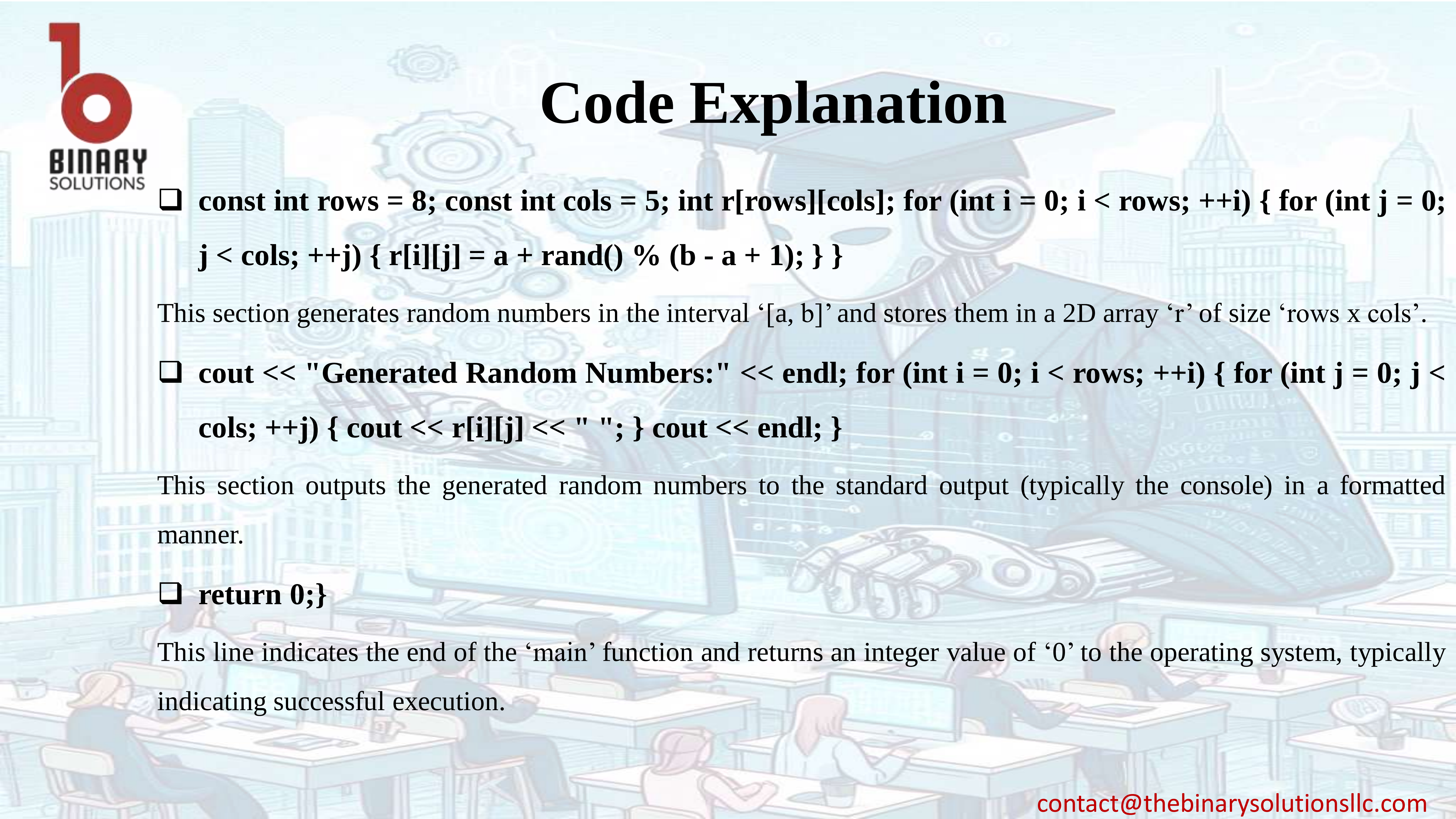
This line marks the beginning of the ‘main’ function, which serves as the entry point of the program.

- □ **int a = -2; int b = 5;**

These lines define the interval ‘[a, b]’ as ‘[-2, 5]’.

- □ **srand(time(nullptr));**

This line seeds the random number generator using the current time, ensuring that different random sequences are generated on each program run.



Code Explanation

- **const int rows = 8; const int cols = 5; int r[rows][cols]; for (int i = 0; i < rows; ++i) { for (int j = 0; j < cols; ++j) { r[i][j] = a + rand() % (b - a + 1); } }**

This section generates random numbers in the interval ‘[a, b]’ and stores them in a 2D array ‘r’ of size ‘rows x cols’.

- **cout << "Generated Random Numbers:" << endl; for (int i = 0; i < rows; ++i) { for (int j = 0; j < cols; ++j) { cout << r[i][j] << " "; } cout << endl; }**

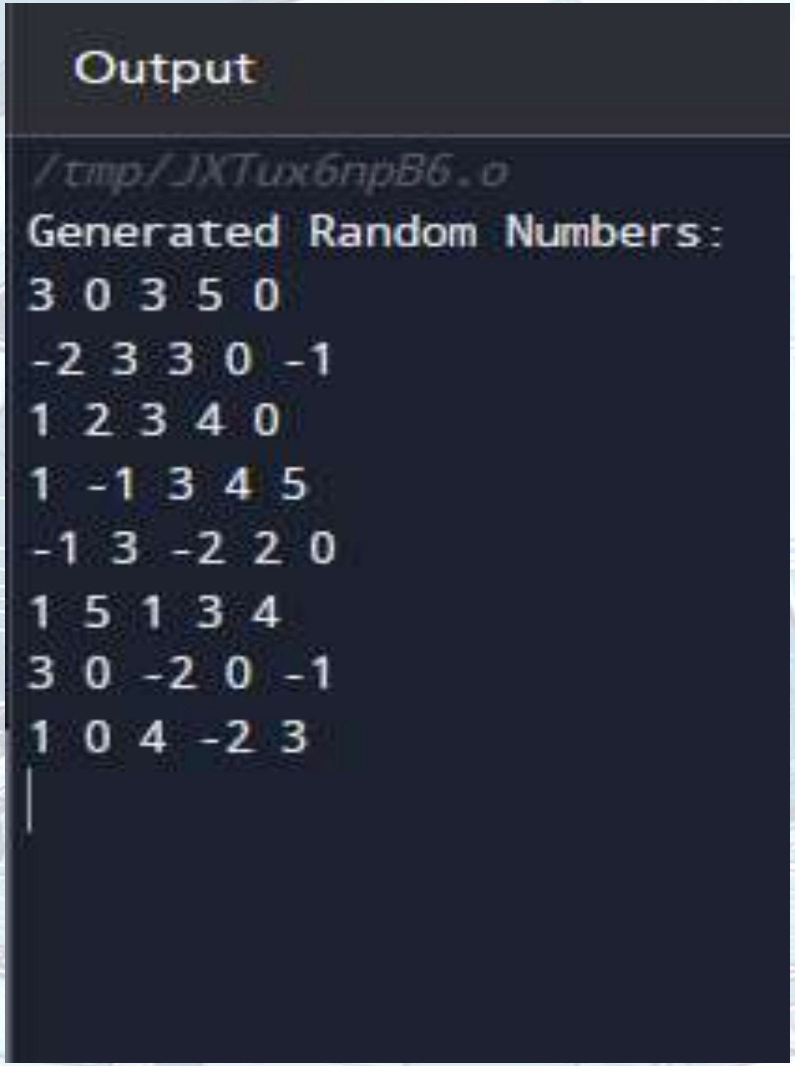
This section outputs the generated random numbers to the standard output (typically the console) in a formatted manner.

- **return 0;}**

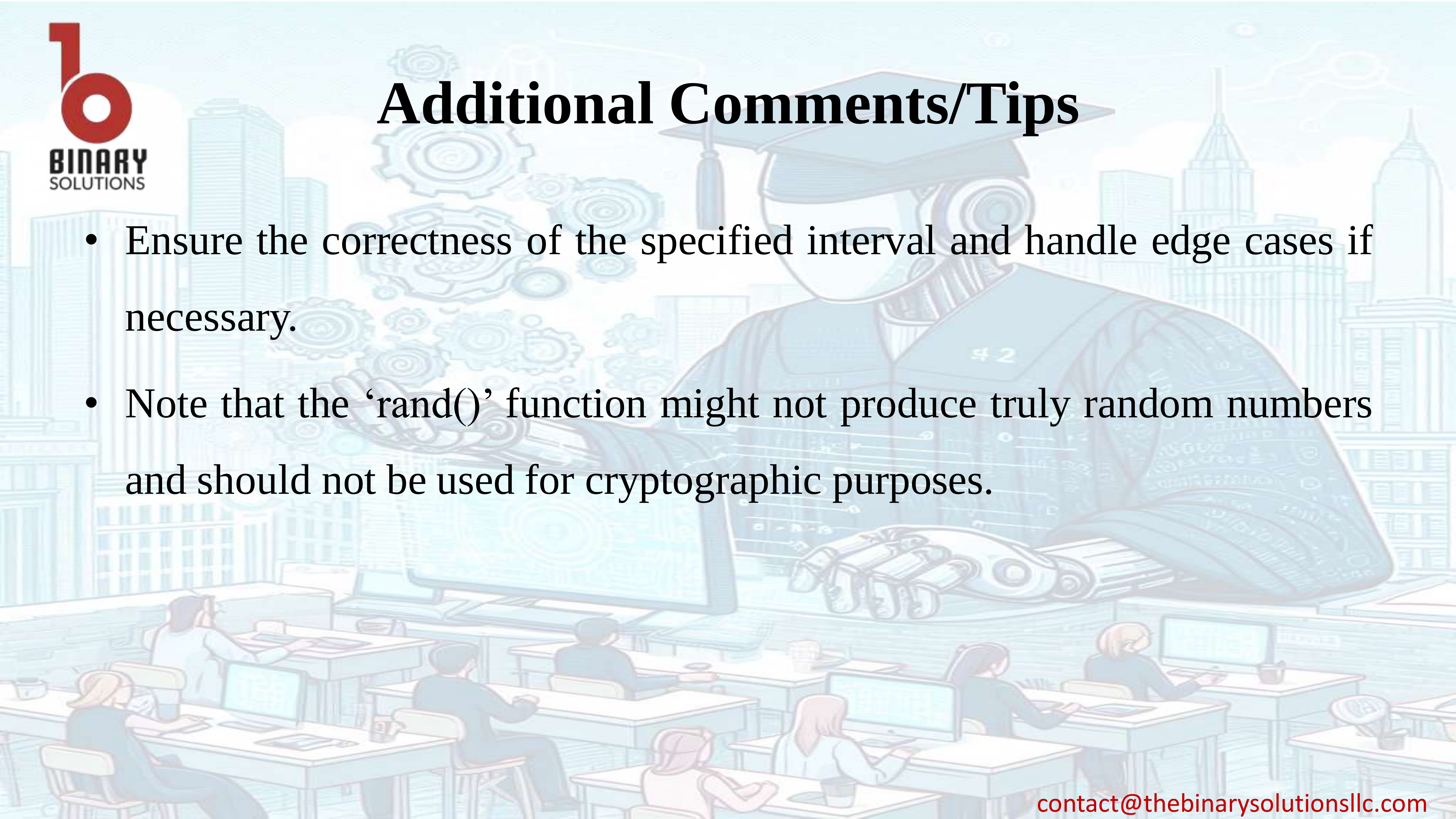
This line indicates the end of the ‘main’ function and returns an integer value of ‘0’ to the operating system, typically indicating successful execution.

Final Answer

- The final output is the matrix rr containing random numbers within the interval $[-2,5]$.



```
/tmp/JXTux6npB6.o
Generated Random Numbers:
3 0 3 5 0
-2 3 3 0 -1
1 2 3 4 0
1 -1 3 4 5
-1 3 -2 2 0
1 5 1 3 4
3 0 -2 0 -1
1 0 4 -2 3
```



Additional Comments/Tips

- Ensure the correctness of the specified interval and handle edge cases if necessary.
- Note that the ‘rand()’ function might not produce truly random numbers and should not be used for cryptographic purposes.

Conclusion

This program demonstrates the generation of random numbers within a specified interval in C++, which is useful in various applications such as simulations, games, and statistical analysis.