



Blend Vidya  
EdTech

# C-PROGRAMMING

TRAINING PROGRAM



Visit Our Website:  
[www.blendvidya.com](http://www.blendvidya.com)



```
complement(A, B,  
display(&Avertex),  
display(B, &vertex);  
}  
25 }  
26 void generation(int A[max][max])  
26 void generation(int A[max][max])  
27 {  
27 {  
28     int i=0, j, res=0;  
28     int i=0, j, res=0;  
29     for(i=0; i<vertex; i++)  
29     for(i=0; i<vertex; i++)  
30     {  
30     {  
31         j=0;  
31         j=0;  
32         for(; j<=1; j++)  
32         for(; j<=1; j++)  
33         {  
33         {  
34             res= rand()%2;  
34             res= rand()%2;  
35             A[i][j]= res;  
35             A[i][j]= res;  
36         }  
36         }  
37     }  
37     }  
38 }
```



# About Company

Blend Vidya is the leading edtech platform. we're dedicated to transforming the way people learn, teach, and engage with educational content. Our mission is to empower educators and learners of all ages to achieve their full potential through cutting-edge technology and dynamic solutions. Founded in 2023, Blend Vidya emerged from a passion for leveraging technology to enhance education. We recognized the need for a more interactive, personalized, and accessible approach to learning. They set out to create a platform that bridges the gap between traditional educational methods and the digital age.





# Vision & Mission



## Company Vision

Our vision Blend Vidya is to revolutionize the global education landscape by harnessing the potential of technology to create a dynamic, inclusive, and personalized learning experience for every individual. We envision a world where education transcends physical barriers, empowers learners of all backgrounds, and fosters a lifelong love for learning. Through our innovative solutions, we aspire to cultivate a future where knowledge knows no bounds and opportunities for growth are limitless.

## Company Mission

Our mission is to empower learners and educators through innovative technology, fostering personalized, inclusive, and effective education for all. Our mission is to empower educators, learners, and institutions with cutting-edge educational technology that enhances engagement, personalization, and outcomes.





# What Is C Programming Language?

C programming is a computer programming language that was developed in the early 1970s by Dennis Ritchie at Bell Labs. It is a general-purpose, procedural programming language that has been widely used for developing a variety of applications, ranging from system software to desktop applications and even embedded systems. C played a significant role in the development of operating systems like Unix and has influenced the design of many other programming languages. C enforces strict typing rules, requiring variables to be declared with their data types before use. C is a procedural programming language, which means it focuses on breaking down a program into a series of functions or procedures. It emphasizes structured programming techniques. This catches type-related errors at compile time.

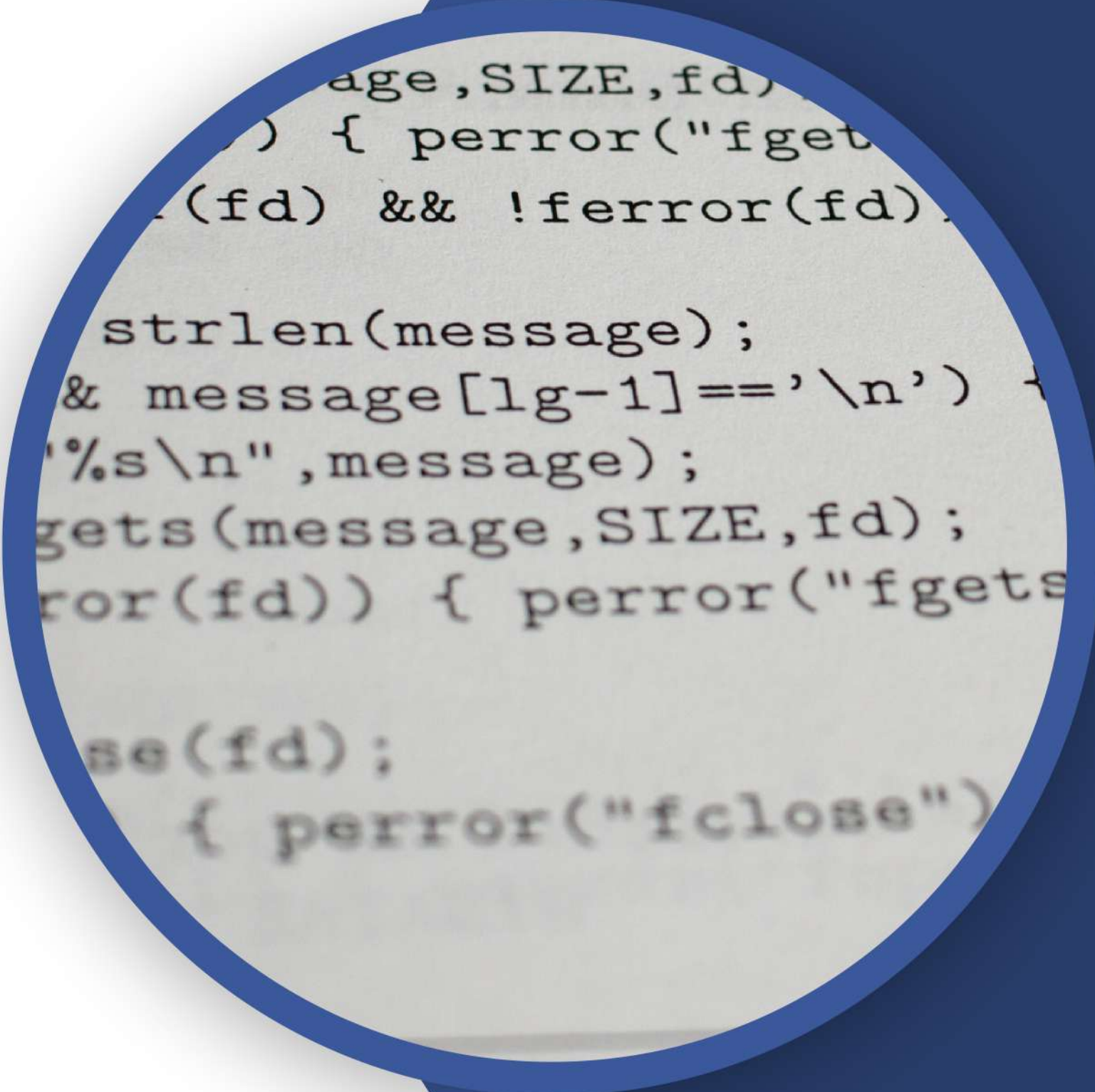
```
double dbTemp;  
bool again = true;  
  
while (again) {  
    iN = -1;  
    again = false;  
    getline(cin, sInput);  
    system("cls");  
    stringstream(sInput) >> dbTemp;  
    iLength = sInput.length();  
    if (iLength < 4) {  
        again = true;  
        continue;  
    } else if (sInput[iLength - 3] != '.') {  
        again = true;  
        continue;  
    } while (++iN < iLength) {  
        if (isdigit(sInput[iN])) {  
            continue;  
        } else if (iN == (iLength - 3)) {  
            continue;  
        }  
    }  
}
```





# Why Is It Important To learn C-Programming?

C programming offers several benefits that have contributed to its enduring popularity and relevance in the field of computer programming & software development. C programming offers numerous benefits, it's important to note that it also requires careful memory management and attention to detail. C programs are highly portable across different platforms and operating systems. The language's separation of high-level concepts from machine-specific details enables C programs to be compiled and executed on a variety of hardware and software environments with minimal modifications. C can be used to develop a wide range of applications, from small utility programs to complex software systems. Its flexibility and wide range of features make it suitable for various programming tasks. C has been used extensively in the development of operating systems (such as Unix and its derivatives), embedded systems, and other critical software components. As a result, there is a vast repository of libraries, tools, and resources available for C programmers. Mistakes in C programming can lead to difficult-to-debug issues and security vulnerabilities. However, with proper understanding and best practices, these challenges can be effectively managed. Proficiency in C programming can open up career opportunities in areas such as systems programming, embedded systems development, game programming, scientific computing, and more.



```
...age, SIZE, fd);  
... { perror("fget  
... (fd) && !ferror(fd);  
  
strlen(message);  
& message[lg-1]=='\n') +  
"%s\n", message);  
gets(message, SIZE, fd);  
ror(fd)) { perror("fgets  
  
se(fd);  
{ perror("fclose")
```



# What you will learn?

## Introduction to C Programming::

- Overview of C programming language
- History and significance of C
- Setting up the development environment (compiler, IDE)
- Writing and compiling a simple C program
- Structure of a C program: main function, comments, statements





# What you will learn?

## Basic Concepts:

- Variables, data types, and constants
- Input and output using printf and scanf
- Arithmetic operators, expressions, and precedence
- Conditional statements: if, else, switch





# What you will learn?

## Control Structures:

- Loops:
  1. while
  2. do-while
  3. for
- Break and continue statements
- Nested loops and loop control





# What you will learn?

## Functions and Modular Programming:

- Defining and calling functions
- Function prototypes and headers
- Return values and void functions
- Scope and lifetime of variables
- Modular programming concepts





# What you will learn?

## Arrays and Strings:

- Declaring and initializing arrays
- Accessing array elements
- Multi-dimensional arrays
- Strings in C: declaration, initialization, manipulation
- String handling functions





## What you will learn?

### Pointers and Memory Management:

- Introduction to pointers
- Pointer arithmetic
- Pointers and arrays
- Dynamic memory allocation: malloc, calloc, realloc, free





# What you will learn?

## Structures and Unions:

- Defining and using structures
- Nested structures
- Unions and their applications
- Typedef for creating custom data types





# What you will learn?

## File Handling:

- Input and output operations using files
- fopen, fclose, fprintf, fscanf, etc.
- Error handling and file manipulation





# What you will learn?

## Debugging and Troubleshooting:

- Debugging techniques and tools
- Common programming errors and how to avoid them
- Reading and interpreting compiler and runtime errors





# What you will learn?

## Best Practices and Coding Standards:

- Writing clean and maintainable code
- Code indentation and formatting
- Meaningful variable and function naming
- Commenting and documentation practices





# What you will work on?

## Final Project :

- Application of learned concepts to develop a small project
- Demonstrating proficiency in programming skills





# Thank You

