**EXPERIMENT 16**

**Develop a C program for implementing random access file for processing the employee details**

## AIM :

To develop a C program for implementing random access file for processing the employee details

## ALGORITHM :

1. Define Structure: Define a structure to represent employee details. Include attributes like employee ID, name, salary, and any other relevant information.
2. Open File in Binary Mode: Open a file in binary mode using the fopen function. Specify the file path and mode ("rb+" for reading and writing binary files).
3. Menu-Driven Interface: Create a menu-driven interface for the user to perform operations. Options could include adding a new employee, updating existing employee details, searching for an employee, deleting an employee, listing all employees, and exiting the program.
4. Implement Functions: Implement functions corresponding to each menu option. For example, implement functions to add a new employee, update employee details, search for an employee by ID, delete an employee, and list all employees. These functions should perform file operations like reading and writing records.
5. Random Access File Operations: Utilize fseek and ftell functions to perform random access file operations. Use fseek to move the file pointer to the desired record based on the employee ID and ftell to determine the current position of the file pointer.
6. File Read and Write: Implement functions to read and write employee records to the file. Use fread and fwrite functions to read and write structures to the file.
7. Error Handling: Implement error handling to deal with situations where the file cannot be opened or when operations like adding, updating, or deleting employees fail. Display appropriate error messages to the user.
8. Close the File: Close the file using the fclose function when the program is exiting or when the file operations are completed.
9. Testing: Test the program thoroughly by adding, updating, searching, and deleting employee records. Ensure that the program handles edge cases and errors gracefully.

Documentation (Optional): Add comments and documentation to your code to explain the functionality of different sections, making it easier for others (and yourself) to understand the code in the future.

OUTPUT:

