B.BHANUTEJA REDDY-192325016

27. Develop a C program for simulating the function of ls UNIX Command.

AIM

To develop a C program that simulates the functionality of the ls UNIX command.

ALGORITHM

1. **Start**
2. Include necessary libraries for handling directory operations.
3. Accept a directory path from the user (or use the current directory by default).
4. Open the specified directory using opendir().
5. If the directory is valid:
   - Read the entries in the directory using readdir().
   - Display the names of all files and subdirectories, excluding . and ...
6. Close the directory after reading all entries.
7. Display an error message if the directory cannot be opened.
8. **Stop**

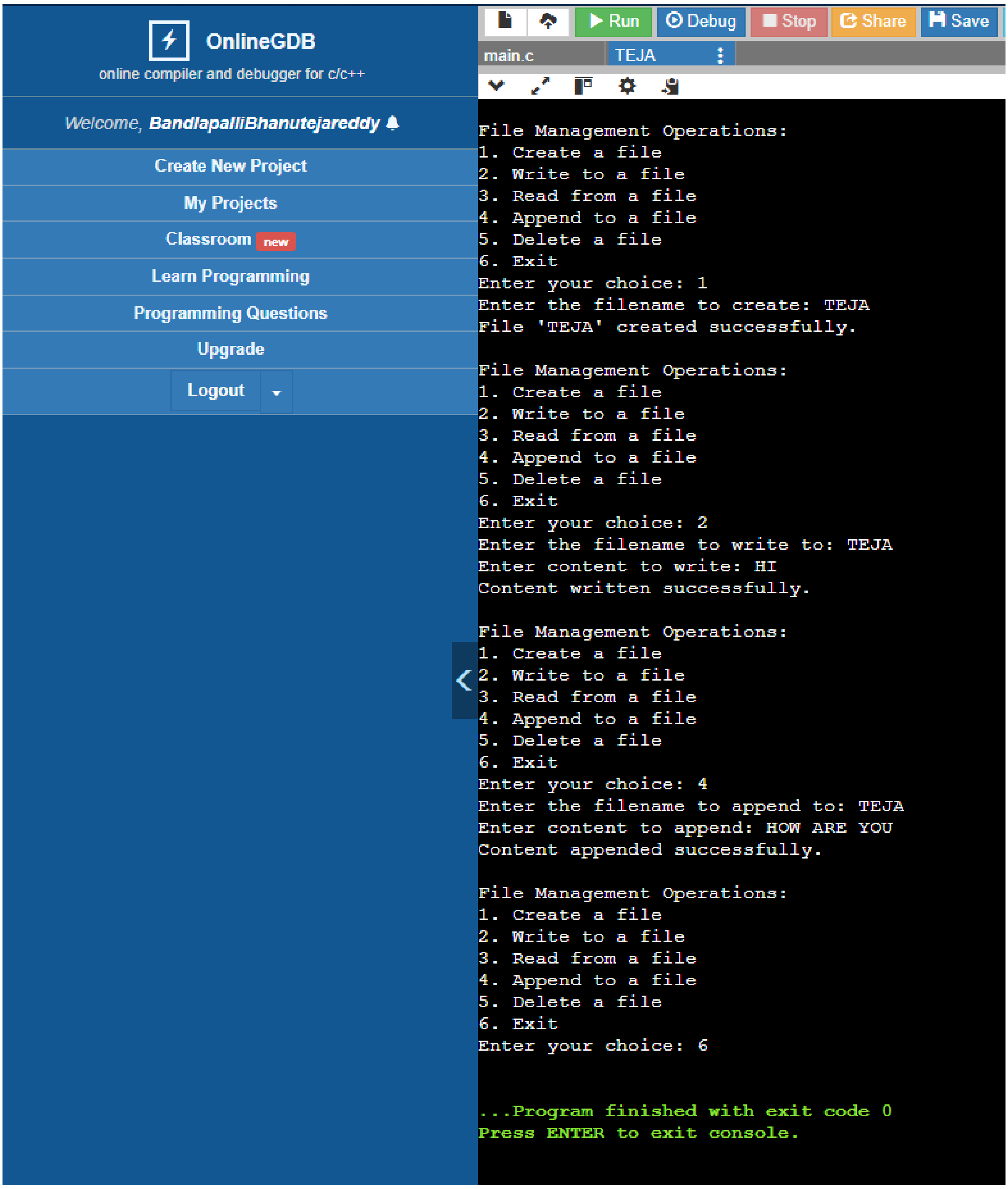
PROCEDURE

1. Use the <dirent.h> library for directory handling.
2. Implement error checking to handle invalid directories.
3. Use a loop to read and display all entries in the directory.
4. Use conditional checks to exclude special entries like . and ...

CODE:

```
#include <stdio.h>

#include <stdlib.h>

#include <dirent.h>


void listDirectoryContents(const char *path) {
    struct dirent *entry;
    DIR *directory = opendir(path);

    if (directory == NULL) {
        printf("Error: Unable to open directory '%s'\n", path);
        return;
    }
    printf("Contents of directory '%s':\n", path);
    while ((entry = readdir(directory)) != NULL) {
        if (entry->d_name[0] != '.') {
            printf("%s\n", entry->d_name);
        }
    }
    closedir(directory);
}

int main() {
    char path[256];
    printf("Enter the directory path (or press Enter for current directory): ");
    fgets(path, sizeof(path), stdin);
    size_t len = strlen(path);
    if (len > 0 && path[len - 1] == '\n') {
        path[len - 1] = '\0';
    }

    if (path[0] == '\0') {
        listDirectoryContents(".");
    } else {
        listDirectoryContents(path);
    }

    return 0;
}
```

OUTPUT:



```
File Management Operations:
1. Create a file
2. Write to a file
3. Read from a file
4. Append to a file
5. Delete a file
6. Exit
Enter your choice: 1
Enter the filename to create: TEJA
File 'TEJA' created successfully.

File Management Operations:
1. Create a file
2. Write to a file
3. Read from a file
4. Append to a file
5. Delete a file
6. Exit
Enter your choice: 2
Enter the filename to write to: TEJA
Enter content to write: HI
Content written successfully.

File Management Operations:
1. Create a file
2. Write to a file
3. Read from a file
4. Append to a file
5. Delete a file
6. Exit
Enter your choice: 4
Enter the filename to append to: TEJA
Enter content to append: HOW ARE YOU
Content appended successfully.

File Management Operations:
1. Create a file
2. Write to a file
3. Read from a file
4. Append to a file
5. Delete a file
6. Exit
Enter your choice: 6


...Program finished with exit code 0
Press ENTER to exit console.
```