

**Name:** K.SaiKrishna

**Reg-No:** 192311106

28. Write a C program for simulation of GREP UNIX command

**Aim:**

The aim of this C program is to simulate the functionality of the `grep` command in Unix. The program will search for a specified pattern within a given file and print the lines that contain the pattern.

**Algorithm:**

1. Read the file name and the pattern to search for from the user.
2. Open the file for reading.
3. Read the file line by line.
4. For each line, check if the pattern exists in the line using `strstr`.
5. If the pattern is found, print the line.
6. Close the file.

**Procedure:**

1. Accept file name and pattern from the user.
2. Open the file using `fopen`.
3. Use `fgets` to read the file line by line.
4. Use `strstr` to check if the pattern is found in each line.
5. Print the lines that match the pattern.
6. Handle errors like file not found.

**Code:**

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int main() {
```

```
    FILE *file;
```

```
    char line[1000], pattern[100];
```

```
    printf("Enter the filename: ");
```

```
scanf("%s", pattern);

printf("Enter the pattern to search for: ");

scanf("%s", pattern);

file = fopen(pattern, "r");

if (file == NULL) {

    printf("File not found!\n");

    return 1;

}

while (fgets(line, sizeof(line), file)) {

    if (strstr(line, pattern)) {

        printf("%s", line);

    }

}

fclose(file);

return 0;

}
```

**Output:**

The screenshot displays an online compiler and debugger for C/C++ with a sidebar menu on the left and a main workspace on the right. The sidebar includes options like 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main workspace shows a C program named 'main.c' with the following code:

```
8 while (fgets(line, sizeof(line), file)) {
9     if (strstr(line, pattern)) {
10         printf("%s", line); // Print the line if pattern is found
11     }
12 }
13
14
15 int main(int argc, char *argv[]) {
16     if (argc != 3) {
17         fprintf(stderr, "Usage: %s <filename> <pattern>\n", argv[0]);
18         exit(1);
19     }
20
21     const char *filename = argv[1]; // File name from command line argument
22     const char *pattern = argv[2]; // Pattern to search for
23
24     FILE *file = fopen(filename, "r");
25     if (file == NULL) {
26         perror("Error opening file");
27         exit(1);
28     }
29
30     search_pattern(file, pattern);
31
32     fclose(file);
33     return 0;
34 }
35
```

Below the code editor, the console output shows the usage message: 'Usage: ./a.out <filename> <pattern>'. The program finished with exit code 1, and the console prompts the user to press ENTER to exit.

Footer text: out • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2024 GDB Online

## Result:

If the file contains the pattern, the matching lines will be printed to the console. If no match is found, nothing will be printed. If the file doesn't exist, an error message will be displayed.

