
MAIN PROGRAM FILE in searchString.c

/*****

Alanna Hupe

Section 1

HW #5

The function of the program is to search a file input by the user for a pattern also provided by the user and then return the number of times the pattern appears in the file

Variable Directory:

i, j,	counters
lineCount	counts number of lines in input file
filename	name of file input by the user
pattern	string array entered by the user that will be searched for
line	file is read in, then using the getf function stores the content in line in order to perform a series of actions on the string array
length_text	length of the textline (81 characters)
length_pattern	length of the pattern input by the user
match	number of matches found in the text file
file	pointer to the file opened

*****/

```
#include <stdio.h>
#define FILEMAX 20
#define LINEMAX 81
```

```
//Prototype Functions
```

```
int find_string ( char text[LINEMAX], char pattern[LINEMAX], int
length_text, int length_pattern);
```

```
int getPatternLength( char pattern[LINEMAX]);
```

```

int main()
{
//Declare all variables

    char filename[FILEMAX];
    char line [LINEMAX];
    int lineCount = 0;
    int length_text = 0;
    int length_pattern = 0;
    char pattern[LINEMAX];
    int i = 0;
    int j = 0;
    int match = 0;
    FILE *file;

//Get file input from user and echo print

    printf("Enter the file: ");
    scanf("%s", filename);
    printf("\nYou entered the file: %s", filename);

//Create file pointer to input file

    file = fopen( filename, "r");

//Get pattern string from user and echo print

    printf("\nEnter the pattern you are looking for in the file: ");
    scanf("%s", pattern);
    printf("\nYou entered the pattern: %s \n\n", pattern);

//Clear the line

    for(j = 0; j < LINEMAX; j++)
    {
        line[j] = (char)0;
    }

//Get the length of the pattern that user input

    length_pattern = getPatternLength( pattern);

//Here's where the fun starts
//Use fgets to read the text file line by line
//Prints the line number and the text from each line
//For each line, add characters from line together until the file end

```

```

//use find_string function to add up pattern matches found in file
//at the end, clear the lines

while (fgets(line, sizeof(line), file))
{
    lineCount++;
    printf("%1d %10s \n",lineCount, line);
    for (i = 0; i < line[i]; i++)
    {
        if(line[i] != (char)0)
        {
            length_text++;
        }
    }
    match = find_string(line, pattern, length_text,
length_pattern) + match;

    length_text = 0;
    for(j = 0; j < LINEMAX; j++)
    {
        line[j] = (char)0;
    }
}
//Echo print the number of matches found

printf("Number of matches for the pattern %s: %d\n\n",pattern, match);

//Close the file and return 0

fclose(file);
return 0;
}

```

```

//FUNCTION

//Return pattern length by counting the characters in the pattern
/*****
Variable Directory:
i, j      counters
length    length of pattern argument
pattern    string array that user has entered
*****/

int getPatternLength( char pattern[LINEMAX])
{
    int i;
    int length = 0;

    for (i = 0; i < pattern[i]; i++)
    {
        if( pattern[i] != ' '){
            length++;
        }
    }
    return length;
}

```

Function in find_string.c

```

/*****
Variable Directory:
i,j           counters
match         number of matches found in the text
matchIndex    number of indexes that the pattern
               and text have matched
index         index number
length_text   length of the line
length_pattern length of the pattern
text          character string array from input file
pattern       string array to be searched for within text
*****/
#define LINEMAX 81

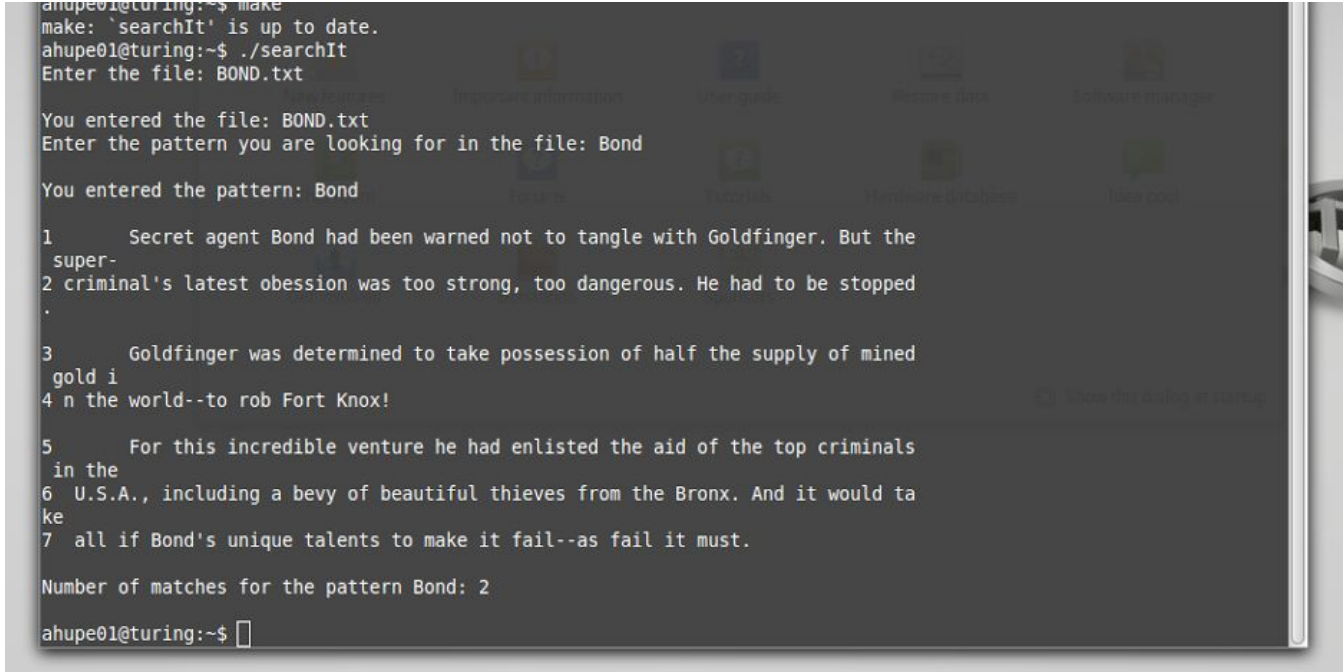
int find_string ( char text[LINEMAX], char pattern[LINEMAX], int
length_text, int length_pattern){
    int j;
    int index = 0;
    int match = 0;
    int matchIndex = 1;

    for ( j = 0; j < length_text; j++){
        while ( text[j + index] == pattern[index] ){
            index++;
            matchIndex++;
            if (matchIndex == length_pattern){
                match++;
            }
        }
        matchIndex = 1;
    }
    return match;
}

```

Makefile OUTPUT stored in file searchIt

OUTPUT

A terminal window with a dark background and light-colored text. The prompt is 'ahupe01@turing:~\$'. The user enters 'make', and the output is 'make: `searchIt' is up to date.'. The user then enters './searchIt', and the prompt changes to 'ahupe01@turing:~\$./searchIt'. The script asks 'Enter the file: BOND.txt'. The user enters 'BOND.txt', and the script asks 'Enter the pattern you are looking for in the file: Bond'. The user enters 'Bond', and the script displays the contents of the file BOND.txt, which is a James Bond story snippet. The script then outputs 'Number of matches for the pattern Bond: 2'. The prompt returns to 'ahupe01@turing:~\$'.

```
ahupe01@turing:~$ make
make: `searchIt' is up to date.
ahupe01@turing:~$ ./searchIt
Enter the file: BOND.txt

You entered the file: BOND.txt
Enter the pattern you are looking for in the file: Bond

You entered the pattern: Bond

1      Secret agent Bond had been warned not to tangle with Goldfinger. But the
super-
2 criminal's latest obsession was too strong, too dangerous. He had to be stopped
.

3      Goldfinger was determined to take possession of half the supply of mined
gold i
4 n the world--to rob Fort Knox!

5      For this incredible venture he had enlisted the aid of the top criminals
in the
6 U.S.A., including a bevy of beautiful thieves from the Bronx. And it would ta
ke
7 all if Bond's unique talents to make it fail--as fail it must.

Number of matches for the pattern Bond: 2

ahupe01@turing:~$
```