

CS2263 Assignment 3
Stephen Cole
3553803

Code:

findtags.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>
#include"htags.h"

#define MAX_TYPE_TAGS 99
#define MAX_TAG_SIZE 20
#define MAX_NUM_TAGS 5000
#define MAX_STRING_LENGTH 255

/*
 * This whole program is a mess
 */
int main(int argc, char* argv[])
{
    FILE *fp;
    char *line = (char*)malloc(MAX_STRING_LENGTH);

    fp = fopen(argv[1], "r");
    if(!fp)
        perror(argv[1]), exit(1);

    int numLines = getlines(fp, line);

    char** inputArr = (char**)malloc(numLines*sizeof(char*));
    instantiateInputArray(inputArr, line, fp);

    char** indexTable = (char**)malloc(MAX_NUM_TAGS);
    int i = 1;
    while(i < MAX_NUM_TAGS)
    {
        *(indexTable + i) = (char*)malloc(1);
        i++;
    }
}
```

```

int currentIndex = 1;
for(i=1; i<=numLines; i++)
{
    char* tagLocation;
    tagLocation = strstr(*(inputArr + i), "<");
    if(tagLocation != NULL)
    {
        *(indexTable + currentIndex) = tagLocation;
        currentIndex++;
        tagLocation++;
        tagLocation = strstr(tagLocation, "<");
        if(tagLocation != NULL)
        {
            *(indexTable + currentIndex) = tagLocation;
            currentIndex++;
        }
    }
}

char** foundTags = (char**)malloc(MAX_NUM_TAGS);
i = 1;
while(i < MAX_NUM_TAGS)
{
    *(foundTags + i) = (char*)malloc(MAX_TAG_SIZE);
    i++;
}

int numTagsFound = 1;
for(i=1; i<currentIndex; i++)
{
    char* start = *(indexTable + i);
    start++;

    if(*start == '/' || *start == '!')
    {
        start++;
        if(*start == '-')
            continue;
    }

    char* tag = (char*)malloc(MAX_STRING_LENGTH);
    scanWord(start, tag);

    numTagsFound = findSameTags(foundTags, tag, indexTable, numTagsFound,
currentIndex);
}

```

```

    }

    fclose(fp);
    free(line);
    free(indexTable);
    free(inputArr);

    return EXIT_SUCCESS;
}

```

htags.h

```

#ifndef INTEGER_H
#define INTEGER_H

void instantiateInputArray(char** inputArr, char* line, FILE* fp);

int getlines(FILE* fp, char* line);

void scanWord(char* start, char* word);

int findSameTags(char** foundTags, char* tag, char** indexTable, int numTagsFound, int
currentIndex);

#endif

```

htags.c

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>
#include"htags.h"

#define MAX_TYPE_TAGS 99
#define MAX_TAG_SIZE 20
#define MAX_NUM_TAGS 5000
#define MAX_STRING_LENGTH 255

int findSameTags(char** foundTags, char* tag, char** indexTable, int numTagsFound, int
currentIndex)
{
    int i;

```

```

    for(i=1; i<numTagsFound+1; i++)
    {
        if(strcmp(tag, *(foundTags + i)) == 0)
            return numTagsFound;
    }

    int numReps = 0;
    int j=1;
    for(j=1; j<currentIndex; j++)
    {
        char* start = *(indexTable + j);
        start++;

        if(*start == '/' || *start == '!')
        {
            start++;
            if(*start == '-')
                continue;
        }

        char* word = (char*)malloc(MAX_STRING_LENGTH);
        scanWord(start, word);

        if(strcmp(tag, word) == 0)
            numReps++;
    }

    printf("%s  %d\n", tag, numReps);
    strcpy(*(foundTags + i), tag);
    return ++numTagsFound;
}

void instantiateInputArray(char** inputArr, char* line, FILE* fp)
{
    int i = 1;
    while(fgets(line, MAX_STRING_LENGTH, fp))
    {
        *(inputArr + i) = (char*)malloc(MAX_STRING_LENGTH);
        strcpy(*(inputArr + i), line);
        i++;
    }
}

int getlines(FILE* fp, char* line)
{
    int numLines=0;

```

```

        while(fgets(line, MAX_STRING_LENGTH, fp))
            numLines++;

        rewind(fp);

        return numLines;
    }

void scanWord(char* start, char* word)
{
    int counter = 0;
    if(*start == '<')
        start++;
    while(*start != '>' && *start != ' ' && *start != '/')
    {
        *(word + counter) = *start;
        counter++;
        start++;
    }
    *(word + counter) = '\0';
    return;
}

```

output:

```
~/Documents/courses/cs2263/assignments/a3 $ ./tags A3Data/HelloWorld
html    2
head    2
meta    1
title   2
body    2
p       2
~/Documents/courses/cs2263/assignments/a3 $ ./tags A3Data/Sample.ht
html    2
head    2
meta    1
title   2
body    2
strong  2
ol      2
li      3
blink   1
p       4
```

In a few sentences describe the design of your program. Focus on what each of the data structures holds and how each of the functions acts on them.

Data Structures:

I used one string array named inputArr to hold the data passed in. Another string array indexTable to hold pointers to the start of each tag and a third string array to hold the tags I had already displayed so that they would not be repeated.

Functions:

The 4 main functions in my program were scanWord which takes the pointer to the start of the tag and a pointer in which to store the tag. getlines which is used to find the amount of lines in the file, which was needed to allocated space for input array. instantiateInputArray which allocates memory for the input array and fills it with the data from the file. Finally, findSameTags which uses the pointers from the index table and the tag passed into it to print out the tag and the number of times it appears in the file.

