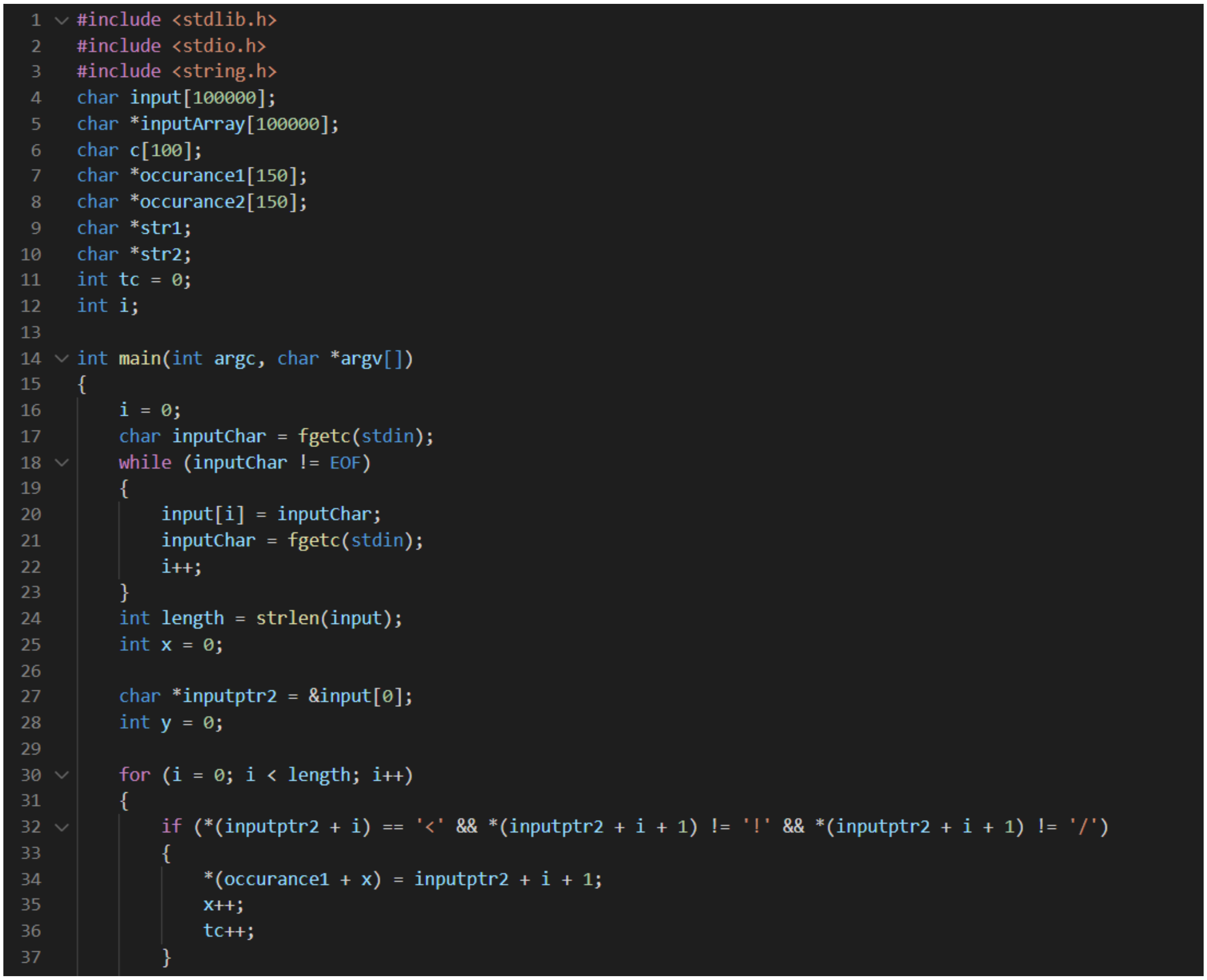
Assignment 3

Summary

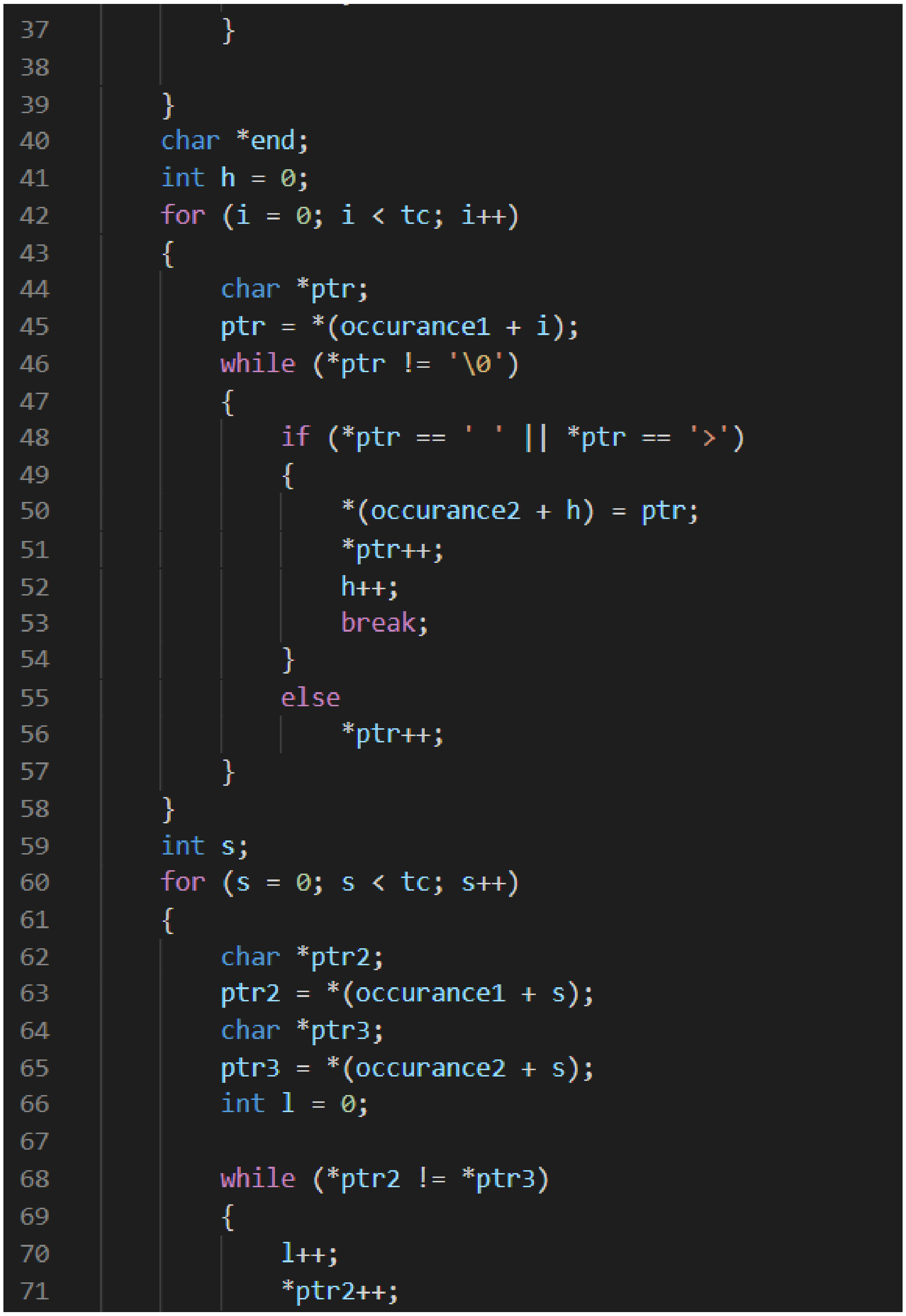
The program is divided into 3 functions, each written in a different file. htagspm is responsible printing the output. htags is responsible for everything else from handling the contents of the html files to pointer manipulation, string storage and everything else needed in the program.

Source codes

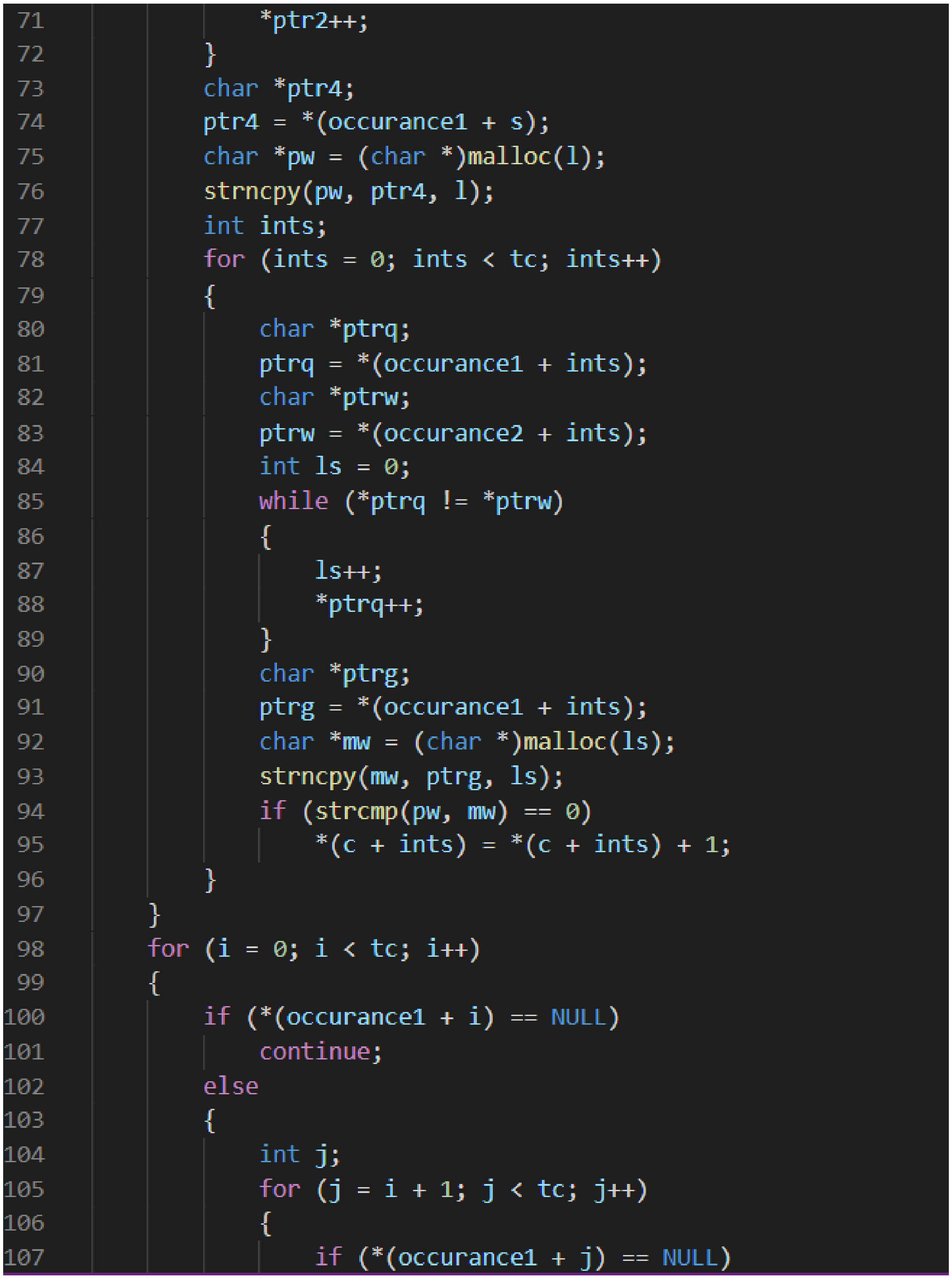
htags.c



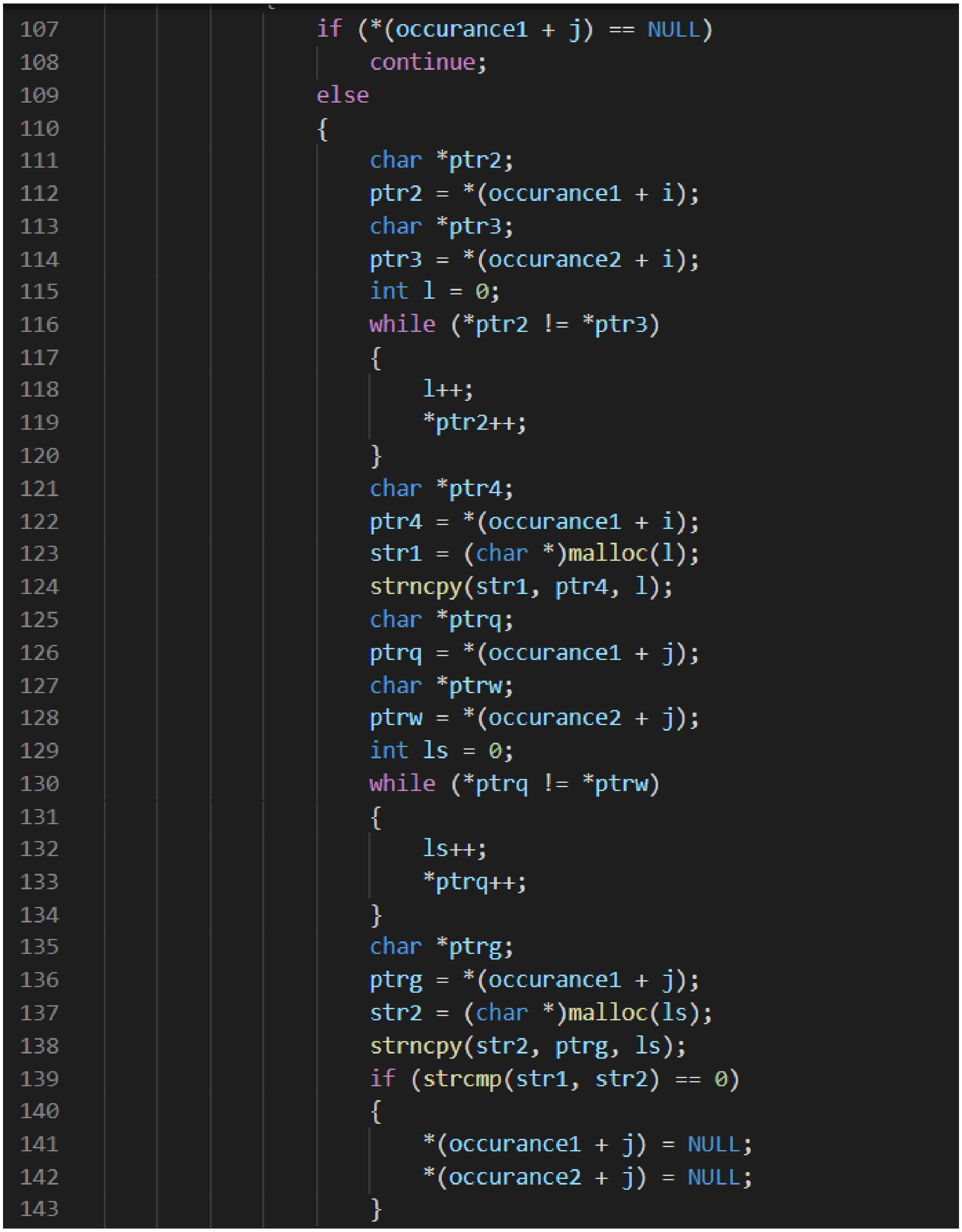
```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
char input[100000];
char *inputArray[100000];
char c[100];
char *occurance1[150];
char *occurance2[150];
char *str1;
char *str2;
int tc = 0;
int i;

int main(int argc, char *argv[])
{
    i = 0;
    char inputChar = fgetc(stdin);
    while (inputChar != EOF)
    {
        input[i] = inputChar;
        inputChar = fgetc(stdin);
        i++;
    }
    int length = strlen(input);
    int x = 0;

    char *inputptr2 = &input[0];
    int y = 0;

    for (i = 0; i < length; i++)
    {
        if (*(inputptr2 + i) == '<' && *(inputptr2 + i + 1) != '!' && *(inputptr2 + i + 1) != '/')
        {
            *(occurance1 + x) = inputptr2 + i + 1;
            x++;
            tc++;
        }
```



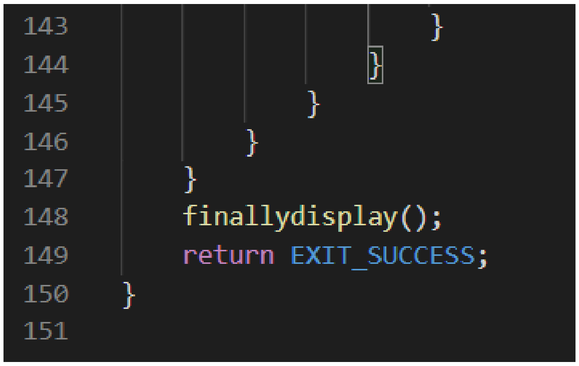
```
        }

    }
    char *end;
    int h = 0;
    for (i = 0; i < tc; i++)
    {
        char *ptr;
        ptr = *(occurance1 + i);
        while (*ptr != '\0')
        {
            if (*ptr == ' ' || *ptr == '>')
            {
                *(occurance2 + h) = ptr;
                *ptr++;
                h++;
                break;
            }
            else
                *ptr++;
        }
    }
    int s;
    for (s = 0; s < tc; s++)
    {
        char *ptr2;
        ptr2 = *(occurance1 + s);
        char *ptr3;
        ptr3 = *(occurance2 + s);
        int l = 0;

        while (*ptr2 != *ptr3)
        {
            l++;
            *ptr2++;
```



```c
            *ptr2++;
        }
        char *ptr4;
        ptr4 = *(occurance1 + s);
        char *pw = (char *)malloc(l);
        strncpy(pw, ptr4, l);
        int ints;
        for (ints = 0; ints < tc; ints++)
        {
            char *ptrq;
            ptrq = *(occurance1 + ints);
            char *ptrw;
            ptrw = *(occurance2 + ints);
            int ls = 0;
            while (*ptrq != *ptrw)
            {
                ls++;
                *ptrq++;
            }
            char *ptrg;
            ptrg = *(occurance1 + ints);
            char *mw = (char *)malloc(ls);
            strncpy(mw, ptrg, ls);
            if (strcmp(pw, mw) == 0)
                *(c + ints) = *(c + ints) + 1;
        }
    }
    for (i = 0; i < tc; i++)
    {
        if (*(occurance1 + i) == NULL)
            continue;
        else
        {
            int j;
            for (j = i + 1; j < tc; j++)
            {
                if (*(occurance1 + j) == NULL)
```

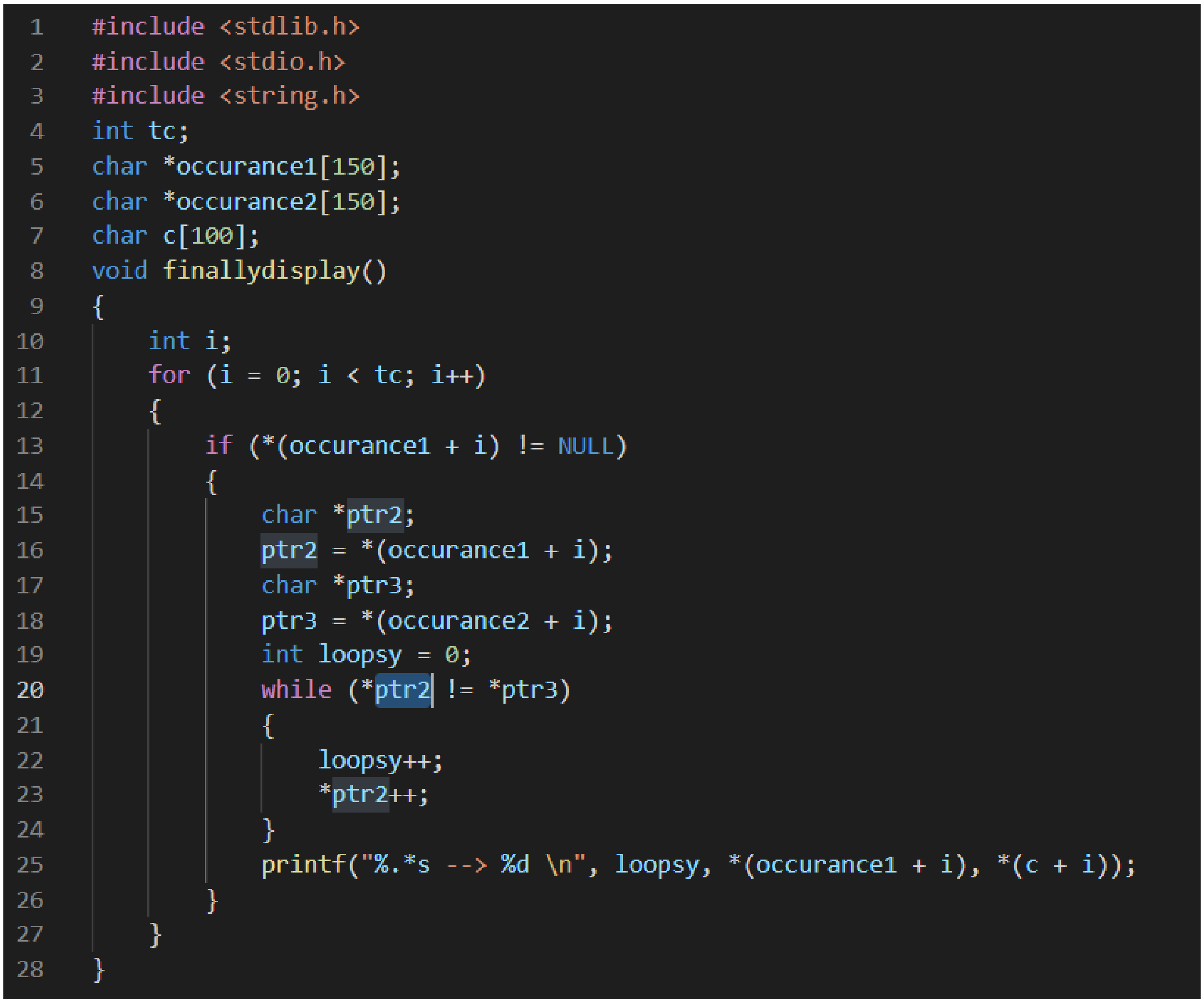


```
                if (*(occurance1 + j) == NULL)
                    continue;
                else
                {
                    char *ptr2;
                    ptr2 = *(occurance1 + i);
                    char *ptr3;
                    ptr3 = *(occurance2 + i);
                    int l = 0;
                    while (*ptr2 != *ptr3)
                    {
                        l++;
                        *ptr2++;
                    }
                    char *ptr4;
                    ptr4 = *(occurance1 + i);
                    str1 = (char *)malloc(l);
                    strncpy(str1, ptr4, l);
                    char *ptrq;
                    ptrq = *(occurance1 + j);
                    char *ptrw;
                    ptrw = *(occurance2 + j);
                    int ls = 0;
                    while (*ptrq != *ptrw)
                    {
                        ls++;
                        *ptrq++;
                    }
                    char *ptrg;
                    ptrg = *(occurance1 + j);
                    str2 = (char *)malloc(ls);
                    strncpy(str2, ptrg, ls);
                    if (strcmp(str1, str2) == 0)
                    {
                        *(occurance1 + j) = NULL;
                        *(occurance2 + j) = NULL;
                    }
```



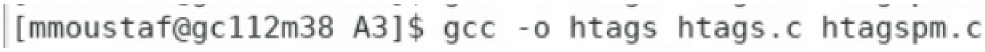
```c
                    }
                }
            }
        }
    }
    finallydisplay();
    return EXIT_SUCCESS;
}

```

htagspm.c



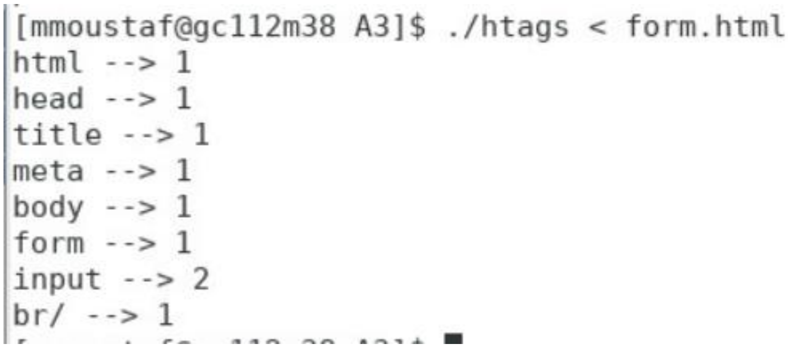
```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int tc;
char *occurance1[150];
char *occurance2[150];
char c[100];
void finallydisplay()
{
    int i;
    for (i = 0; i < tc; i++)
    {
        if (*(occurance1 + i) != NULL)
        {
            char *ptr2;
            ptr2 = *(occurance1 + i);
            char *ptr3;
            ptr3 = *(occurance2 + i);
            int loopsy = 0;
            while (*ptr2 != *ptr3)
            {
                loopsy++;
                *ptr2++;
            }
            printf("%.*s --> %d \n", loopsy, *(occurance1 + i), *(c + i));
        }
    }
}
```

Compile statement



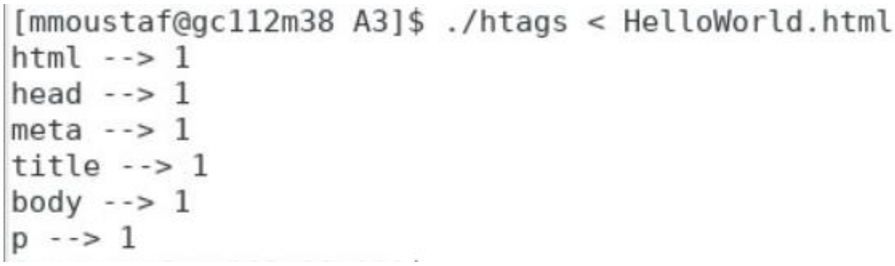
```
[mmoustaf@gc112m38 A3]$ gcc -o htags htags.c htagspm.c
```

htags.c tested with form.html



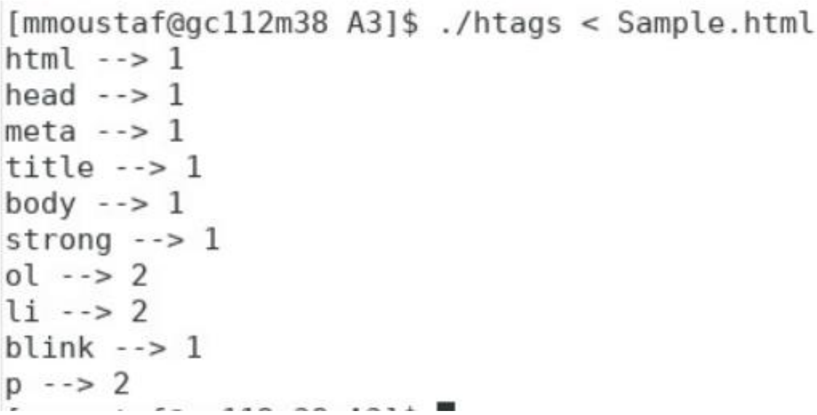
```
[mmoustaf@gc112m38 A3]$ ./htags < form.html
html --> 1
head --> 1
title --> 1
meta --> 1
body --> 1
form --> 1
input --> 2
br/ --> 1
```

Output of HelloWorld.html



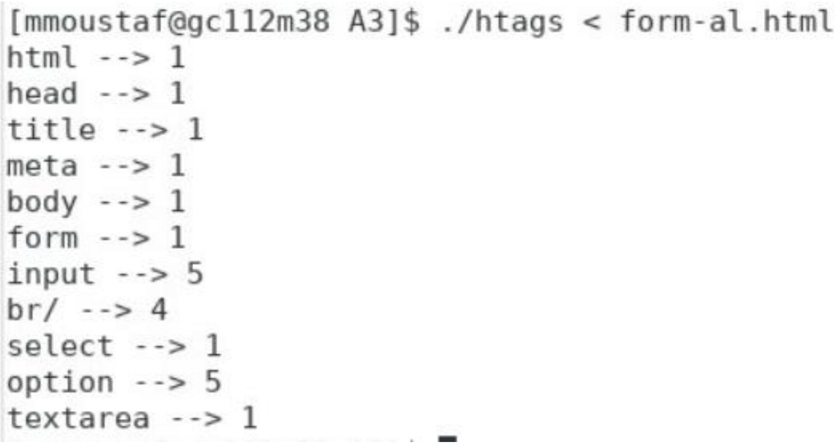
```
[mmoustaf@gc112m38 A3]$ ./htags < HelloWorld.html
html --> 1
head --> 1
meta --> 1
title --> 1
body --> 1
p --> 1
```

Output of Sample.html



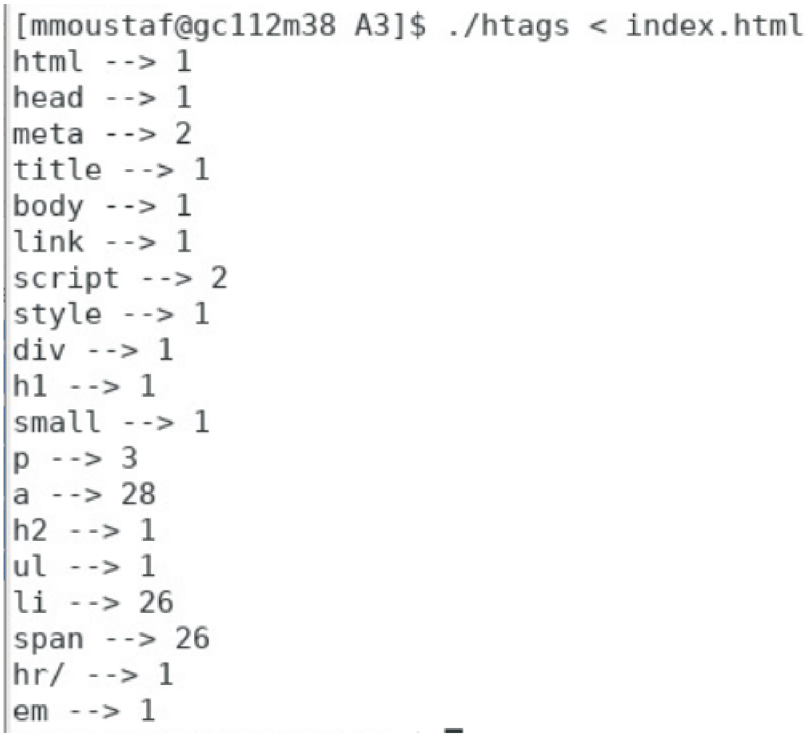
```
[mmoustaf@gc112m38 A3]$ ./htags < Sample.html
html --> 1
head --> 1
meta --> 1
title --> 1
body --> 1
strong --> 1
ol --> 2
li --> 2
blink --> 1
p --> 2
```

Output of form-al.html



```
[mmoustaf@gc112m38 A3]$ ./htags < form-al.html
html --> 1
head --> 1
title --> 1
meta --> 1
body --> 1
form --> 1
input --> 5
br/ --> 4
select --> 1
option --> 5
textarea --> 1
```

Output of index.html



```
[mmoustaf@gc112m38 A3]$ ./htags < index.html
html --> 1
head --> 1
meta --> 2
title --> 1
body --> 1
link --> 1
script --> 2
style --> 1
div --> 1
h1 --> 1
small --> 1
p --> 3
a --> 28
h2 --> 1
ul --> 1
li --> 26
span --> 26
hr/ --> 1
em --> 1
```