

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

```
1  ∨ #include <stdlib.h>
2  #include <stdio.h>
3  #include <string.h>
4  char input[100000];
5  char *inputArray[100000];
6  char c[100];
7  char *occurance1[150];
8  char *occurance2[150];
9  char *str1;
10 char *str2;
11 int tc = 0;
12 int i;
13
14 ∨ int main(int argc, char *argv[])
15 {
16     i = 0;
17     char inputChar = fgetc(stdin);
18 ∨ while (inputChar != EOF)
19     {
20         input[i] = inputChar;
21         inputChar = fgetc(stdin);
22         i++;
23     }
24     int length = strlen(input);
25     int x = 0;
26
27     char *inputptr2 = &input[0];
28     int y = 0;
29
30 ∨ for (i = 0; i < length; i++)
31     {
32 ∨         if (*(inputptr2 + i) == '<' && *(inputptr2 + i + 1) != '!' && *(inputptr2 + i + 1) != '/')
33         {
34             *(occurance1 + x) = inputptr2 + i + 1;
35             x++;
36             tc++;
37         }
```

```

37     }
38
39 }
40 char *end;
41 int h = 0;
42 for (i = 0; i < tc; i++)
43 {
44     char *ptr;
45     ptr = *(occurance1 + i);
46     while (*ptr != '\0')
47     {
48         if (*ptr == ' ' || *ptr == '>')
49         {
50             *(occurance2 + h) = ptr;
51             *ptr++;
52             h++;
53             break;
54         }
55         else
56             *ptr++;
57     }
58 }
59 int s;
60 for (s = 0; s < tc; s++)
61 {
62     char *ptr2;
63     ptr2 = *(occurance1 + s);
64     char *ptr3;
65     ptr3 = *(occurance2 + s);
66     int l = 0;
67
68     while (*ptr2 != *ptr3)
69     {
70         l++;
71         *ptr2++;

```

```

71         *ptr2++;
72     }
73     char *ptr4;
74     ptr4 = *(occurance1 + s);
75     char *pw = (char *)malloc(1);
76     strncpy(pw, ptr4, 1);
77     int ints;
78     for (ints = 0; ints < tc; ints++)
79     {
80         char *ptrq;
81         ptrq = *(occurance1 + ints);
82         char *ptrw;
83         ptrw = *(occurance2 + ints);
84         int ls = 0;
85         while (*ptrq != *ptrw)
86         {
87             ls++;
88             *ptrq++;
89         }
90         char *ptrg;
91         ptrg = *(occurance1 + ints);
92         char *mw = (char *)malloc(ls);
93         strncpy(mw, ptrg, ls);
94         if (strcmp(pw, mw) == 0)
95             *(c + ints) = *(c + ints) + 1;
96     }
97 }
98 for (i = 0; i < tc; i++)
99 {
100     if (*(occurance1 + i) == NULL)
101         continue;
102     else
103     {
104         int j;
105         for (j = i + 1; j < tc; j++)
106         {
107             if (*(occurance1 + j) == NULL)

```

```
107     if (*(occurance1 + j) == NULL)
108     |     continue;
109     else
110     {
111         char *ptr2;
112         ptr2 = *(occurance1 + i);
113         char *ptr3;
114         ptr3 = *(occurance2 + i);
115         int l = 0;
116         while (*ptr2 != *ptr3)
117         {
118             l++;
119             *ptr2++;
120         }
121         char *ptr4;
122         ptr4 = *(occurance1 + i);
123         str1 = (char *)malloc(l);
124         strncpy(str1, ptr4, l);
125         char *ptrq;
126         ptrq = *(occurance1 + j);
127         char *ptrw;
128         ptrw = *(occurance2 + j);
129         int ls = 0;
130         while (*ptrq != *ptrw)
131         {
132             ls++;
133             *ptrq++;
134         }
135         char *ptrg;
136         ptrg = *(occurance1 + j);
137         str2 = (char *)malloc(ls);
138         strncpy(str2, ptrg, ls);
139         if (strcmp(str1, str2) == 0)
140         {
141             *(occurance1 + j) = NULL;
142             *(occurance2 + j) = NULL;
143         }
```

```

143     }
144 }
145 }
146 }
147 }
148     finallydisplay();
149     return EXIT_SUCCESS;
150 }
151

```

htagspm.c

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <string.h>
4  int tc;
5  char *occurance1[150];
6  char *occurance2[150];
7  char c[100];
8  void finallydisplay()
9  {
10     int i;
11     for (i = 0; i < tc; i++)
12     {
13         if (*(occurance1 + i) != NULL)
14         {
15             char *ptr2;
16             ptr2 = *(occurance1 + i);
17             char *ptr3;
18             ptr3 = *(occurance2 + i);
19             int loopsey = 0;
20             while (*ptr2 != *ptr3)
21             {
22                 loopsey++;
23                 *ptr2++;
24             }
25             printf("%.s --> %d \n", loopsey, *(occurance1 + i), *(c + i));
26         }
27     }
28 }

```

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
```