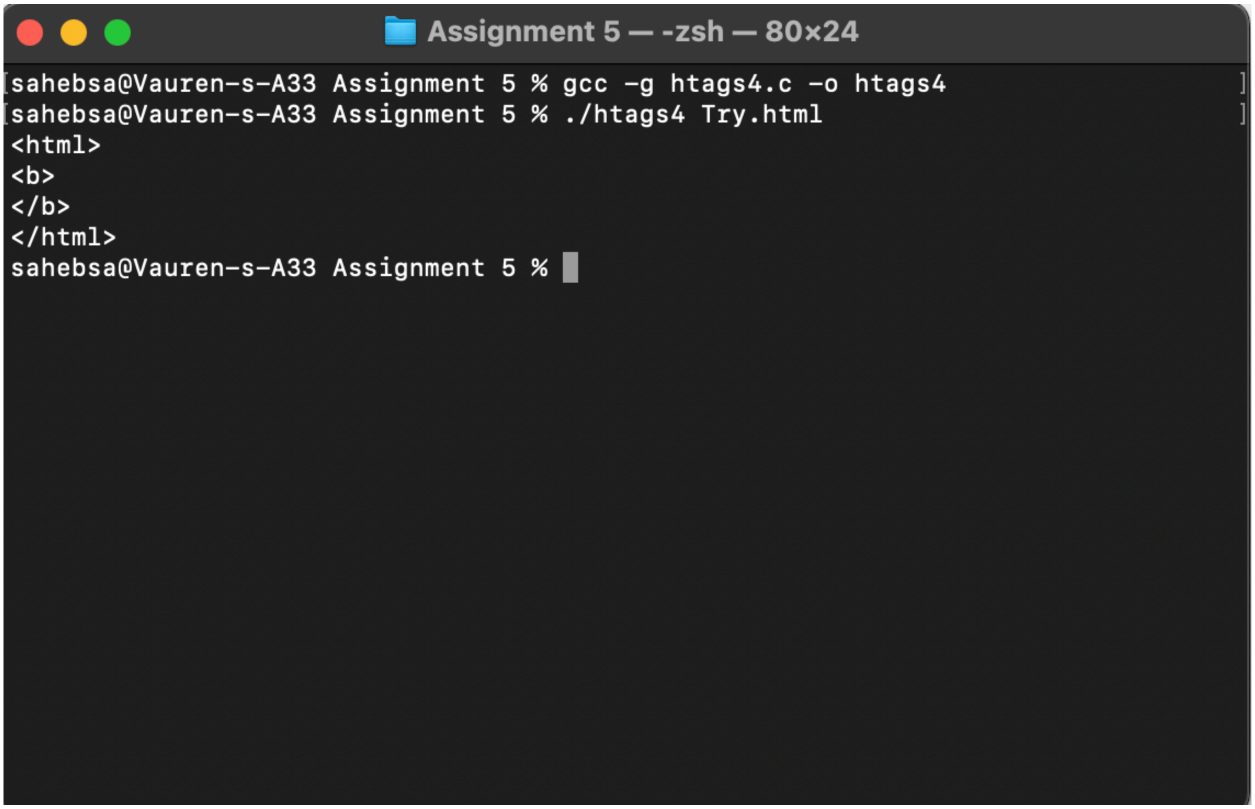
Assignment 5
2263

1)
The htags4 program is designed to examine HTML tags in an HTML file, detecting and printing each unique tag only once. It utilizes a linked list data structure to store the tags dynamically on the heap. The program reads the HTML file line by line, scans each line to extract HTML tags, and checks for duplicate tags using the linked list. New tags are added to the linked list if they are not already present. The program then prints the unique tags stored in the linked list. Additionally, if the -m option is specified, the program tracks memory allocations for each tag and prints allocation details along with the tags. Overall, the program efficiently manages memory and provides accurate analysis of HTML tags in the input file.

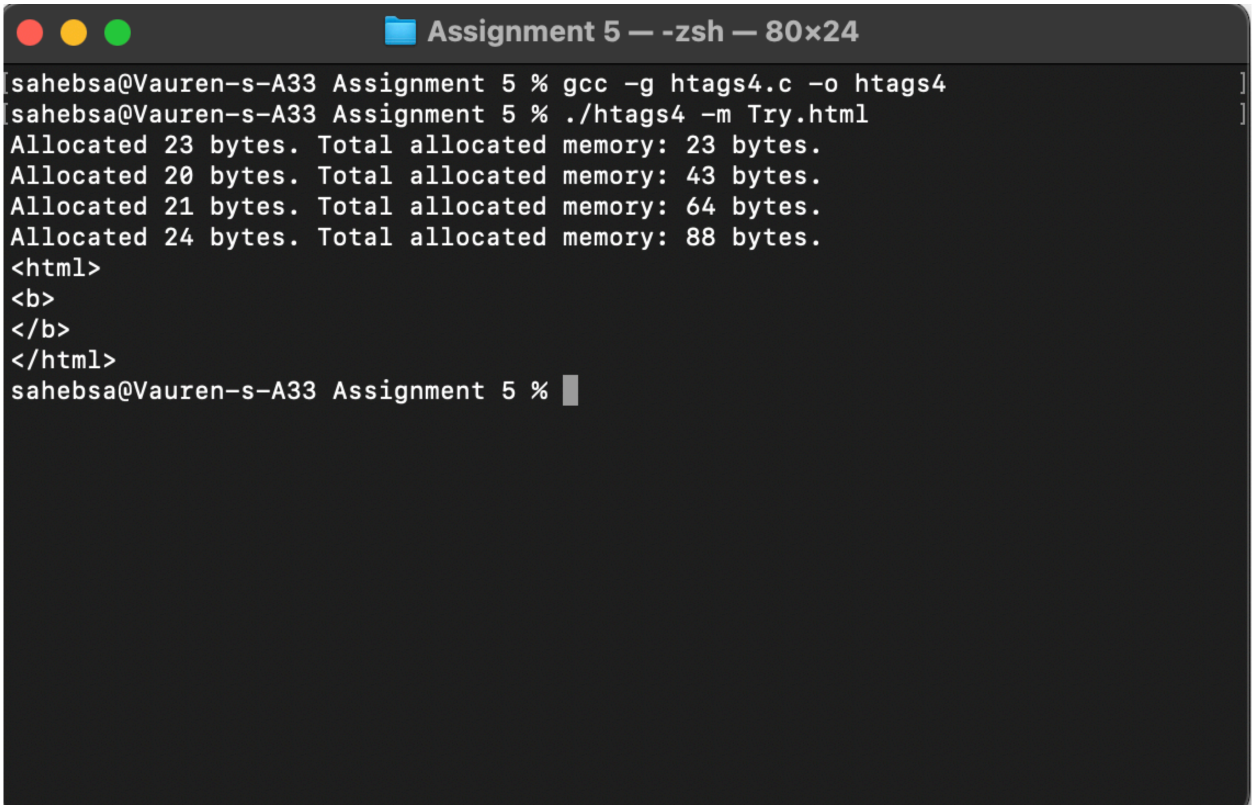
2)
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct listnode {
    struct listnode *next;
    char *tag;
} Node;

Node *head = NULL;
int totalAllocatedMemory = 0;

int isComment(const char *line) {
    return (line[0] == '<' && line[1] == '!' && line[2] == '-'
&& line[3] == '-');
}

Node *Node_construct(char *tag) {
    Node *new_node = (Node *)malloc(sizeof(Node));
    if (new_node == NULL) {
        perror("Memory allocation failed");
        exit(EXIT_FAILURE);
    }
    new_node->next = NULL;
    new_node->tag = strdup(tag);
    if (new_node->tag == NULL) {
        perror("Memory allocation failed");
        exit(EXIT_FAILURE);
    }
    return new_node;
}

void List_add(Node **head_ref, Node *n) {
    if (*head_ref == NULL) {
        *head_ref = n;
        return;
    }
    Node *current = *head_ref;
    while (current->next != NULL) {
        current = current->next;
    }
    current->next = n;
}

int List_search(Node *head, Node *n) {
    Node *current = head;
    while (current != NULL) {
        if (strcmp(current->tag, n->tag) == 0)
            return 1;
        current = current->next;
    }
    return 0;
}

void List_print(Node *head) {
    Node *current = head;
    while (current != NULL) {
        printf("%s\n", current->tag);
        current = current->next;
    }
}

void freeMemory(Node *head) {
    Node *current = head;
    while (current != NULL) {
        Node *temp = current;
        current = current->next;
        free(temp->tag);
        free(temp);
    }
}

int main(int argc, char *argv[]) {
    if (argc < 2) {
        fprintf(stderr, "Usage: %s [-m] <filename>\n", argv[0]);
        return EXIT_FAILURE;
    }

    int memory_tracking = 0;
    if (argc == 3 && strcmp(argv[1], "-m") == 0)
        memory_tracking = 1;

    FILE *file = fopen(argv[argc - 1], "r");
    if (file == NULL) {
        perror("Error opening file");
        return EXIT_FAILURE;
    }

    char line[BUFSIZ];
    while (fgets(line, sizeof(line), file)) {
        const char *start = line;
        while ((start = strchr(start, '<')) != NULL) {
            const char *end = strchr(start, '>');
            if (end == NULL)
                break;
            if (!isComment(start)) {
                int length = end - start + 1;
                char *tag = strndup(start, length);
                if (tag != NULL) {
                    Node *new_node = Node_construct(tag);
                    if (!List_search(head, new_node)) {
                        List_add(&head, new_node);
                        if (memory_tracking) {
                            totalAllocatedMemory +=
(sizeof(Node) + strlen(tag) + 1);
                            printf("Allocated %d bytes. Total
allocated memory: %d bytes.\n", (int)(sizeof(Node) + strlen(tag)
+ 1), totalAllocatedMemory);
                        }
                    } else {
                        free(new_node->tag);
                        free(new_node);
                    }
                } else {
                    perror("Memory allocation failed");
                }
                start = end + 1;
            }
        }
    }
    fclose(file);

    List_print(head);
    freeMemory(head);

    return EXIT_SUCCESS;
}
```

3)



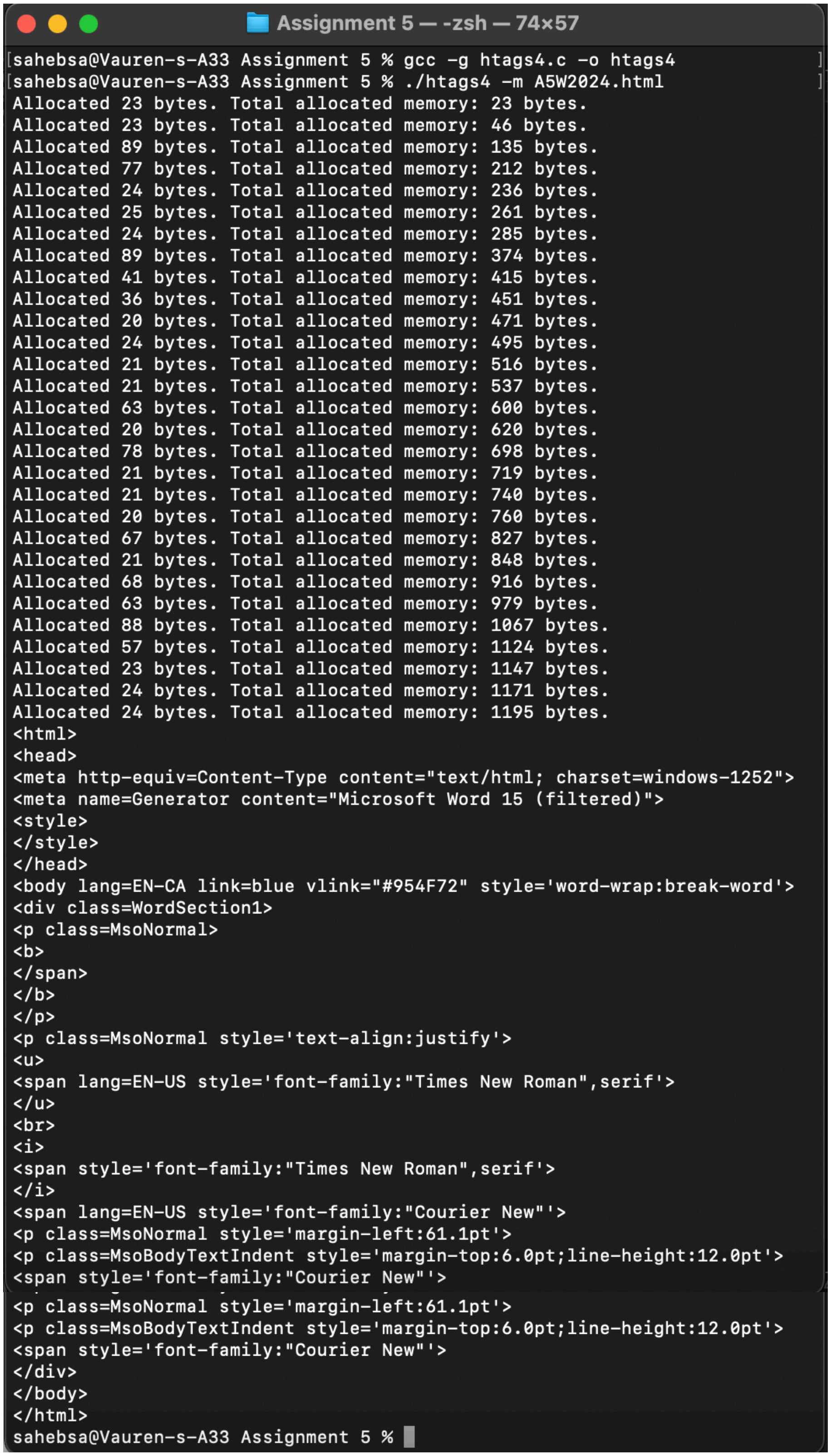
```
[sahebsa@Vauren-s-A33 Assignment 5 % gcc -g htags4.c -o htags4                 ]
[sahebsa@Vauren-s-A33 Assignment 5 % ./htags4 Try.html                         ]
<html>
<b>
</b>
</html>
sahebsa@Vauren-s-A33 Assignment 5 %
```

4)



```
sahebsa@Vauren-s-A33 Assignment 5 % gcc -g htags4.c -o htags4
sahebsa@Vauren-s-A33 Assignment 5 % ./htags4 -m Try.html
Allocated 23 bytes. Total allocated memory: 23 bytes.
Allocated 20 bytes. Total allocated memory: 43 bytes.
Allocated 21 bytes. Total allocated memory: 64 bytes.
Allocated 24 bytes. Total allocated memory: 88 bytes.
<html>
<b>
</b>
</html>
sahebsa@Vauren-s-A33 Assignment 5 %
```

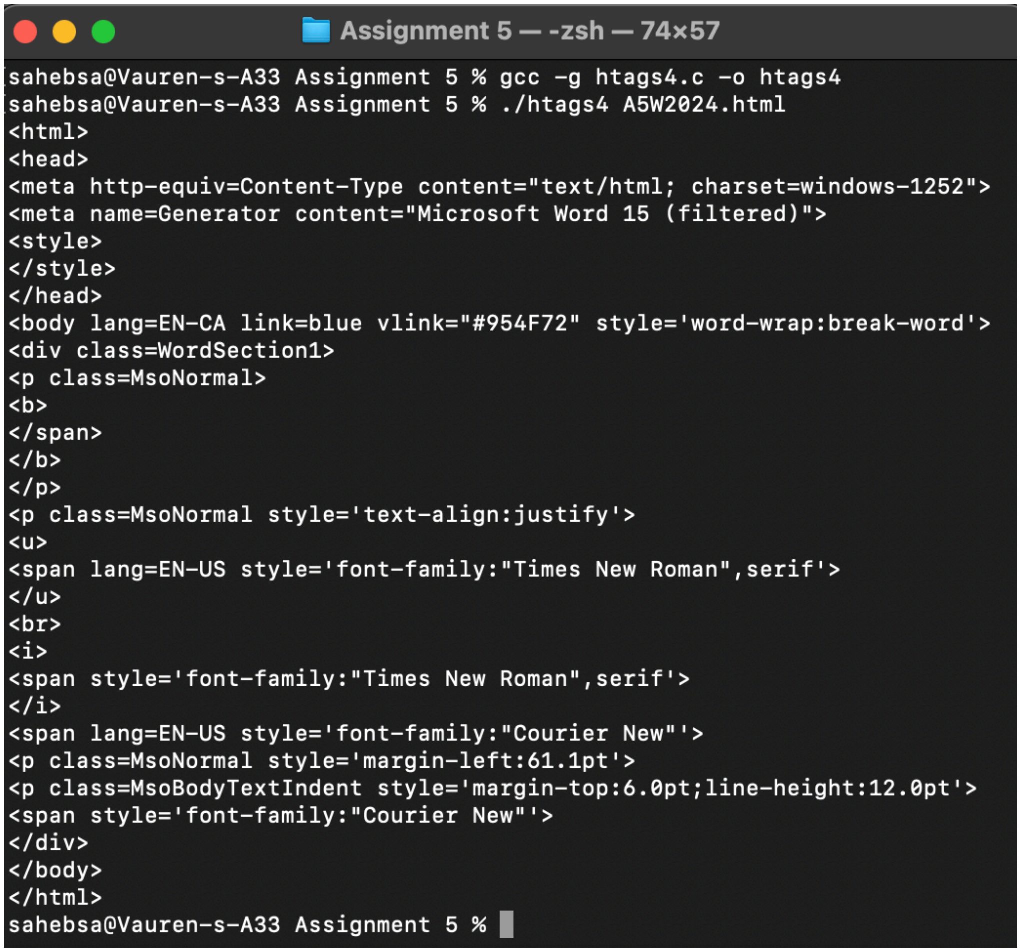
5) **using -m**



```
[sahebsa@Vauren-s-A33 Assignment 5 % gcc -g htags4.c -o htags4          ]
[sahebsa@Vauren-s-A33 Assignment 5 % ./htags4 -m A5W2024.html           ]
Allocated 23 bytes. Total allocated memory: 23 bytes.
Allocated 23 bytes. Total allocated memory: 46 bytes.
Allocated 89 bytes. Total allocated memory: 135 bytes.
Allocated 77 bytes. Total allocated memory: 212 bytes.
Allocated 24 bytes. Total allocated memory: 236 bytes.
Allocated 25 bytes. Total allocated memory: 261 bytes.
Allocated 24 bytes. Total allocated memory: 285 bytes.
Allocated 89 bytes. Total allocated memory: 374 bytes.
Allocated 41 bytes. Total allocated memory: 415 bytes.
Allocated 36 bytes. Total allocated memory: 451 bytes.
Allocated 20 bytes. Total allocated memory: 471 bytes.
Allocated 24 bytes. Total allocated memory: 495 bytes.
Allocated 21 bytes. Total allocated memory: 516 bytes.
Allocated 21 bytes. Total allocated memory: 537 bytes.
Allocated 63 bytes. Total allocated memory: 600 bytes.
Allocated 20 bytes. Total allocated memory: 620 bytes.
Allocated 78 bytes. Total allocated memory: 698 bytes.
Allocated 21 bytes. Total allocated memory: 719 bytes.
Allocated 21 bytes. Total allocated memory: 740 bytes.
Allocated 20 bytes. Total allocated memory: 760 bytes.
Allocated 67 bytes. Total allocated memory: 827 bytes.
Allocated 21 bytes. Total allocated memory: 848 bytes.
Allocated 68 bytes. Total allocated memory: 916 bytes.
Allocated 63 bytes. Total allocated memory: 979 bytes.
Allocated 88 bytes. Total allocated memory: 1067 bytes.
Allocated 57 bytes. Total allocated memory: 1124 bytes.
Allocated 23 bytes. Total allocated memory: 1147 bytes.
Allocated 24 bytes. Total allocated memory: 1171 bytes.
Allocated 24 bytes. Total allocated memory: 1195 bytes.
<html>
<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1252">
<meta name=Generator content="Microsoft Word 15 (filtered)">
<style>
</style>
</head>
<body lang=EN-CA link=blue vlink="#954F72" style='word-wrap:break-word'>
<div class=WordSection1>
<p class=MsoNormal>
<b>
</span>
</b>
</p>
<p class=MsoNormal style='text-align:justify'>
<u>
<span lang=EN-US style='font-family:"Times New Roman",serif'>
</u>
<br>
<i>
<span style='font-family:"Times New Roman",serif'>
</i>
<span lang=EN-US style='font-family:"Courier New"'>
<p class=MsoNormal style='margin-left:61.1pt'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;line-height:12.0pt'>
<span style='font-family:"Courier New"'>
```

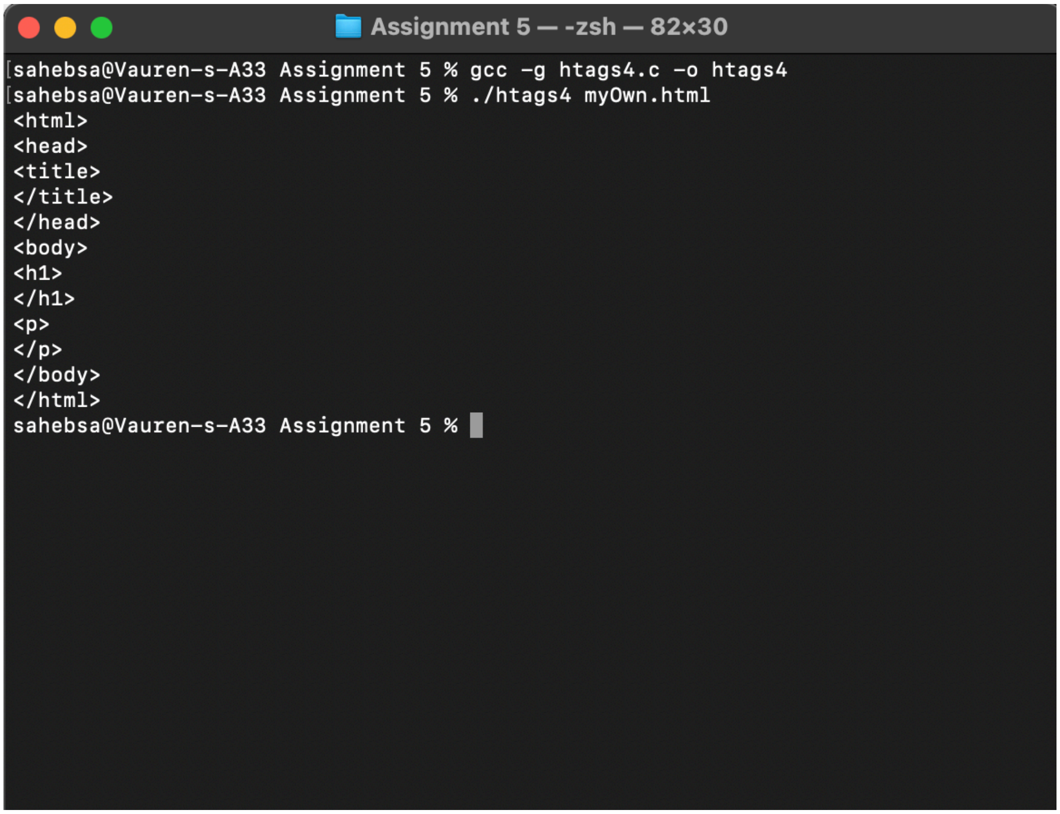
```
<p class=MsoNormal style='margin-left:61.1pt'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;line-height:12.0pt'>
<span style='font-family:"Courier New"'>
</div>
</body>
</html>
sahebsa@Vauren-s-A33 Assignment 5 % 
```

Without using -m



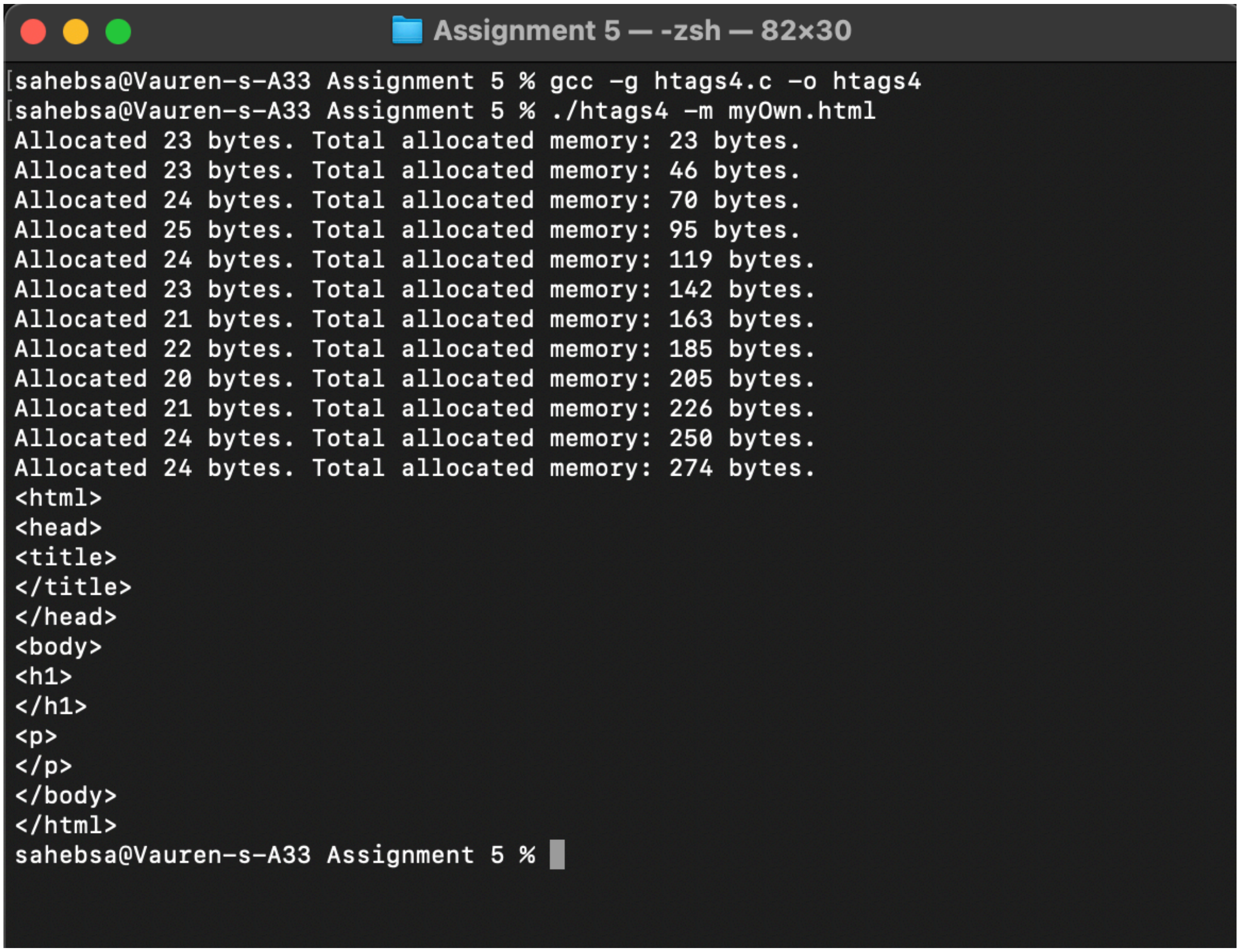
```
sahebsa@Vauren-s-A33 Assignment 5 % gcc -g htags4.c -o htags4
sahebsa@Vauren-s-A33 Assignment 5 % ./htags4 A5W2024.html
<html>
<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1252">
<meta name=Generator content="Microsoft Word 15 (filtered)">
<style>
</style>
</head>
<body lang=EN-CA link=blue vlink="#954F72" style='word-wrap:break-word'>
<div class=WordSection1>
<p class=MsoNormal>
<b>
</span>
</b>
</p>
<p class=MsoNormal style='text-align:justify'>
<u>
<span lang=EN-US style='font-family:"Times New Roman",serif'>
</u>
<br>
<i>
<span style='font-family:"Times New Roman",serif'>
</i>
<span lang=EN-US style='font-family:"Courier New"'>
<p class=MsoNormal style='margin-left:61.1pt'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;line-height:12.0pt'>
<span style='font-family:"Courier New"'>
</div>
</body>
</html>
sahebsa@Vauren-s-A33 Assignment 5 %
```

6) **Without -m**



```
[sahebsa@Vauren-s-A33 Assignment 5 % gcc -g htags4.c -o htags4
[sahebsa@Vauren-s-A33 Assignment 5 % ./htags4 myOwn.html
<html>
<head>
<title>
</title>
</head>
<body>
<h1>
</h1>
<p>
</p>
</body>
</html>
sahebsa@Vauren-s-A33 Assignment 5 %
```

Using -m



```
[sahebsa@Vauren-s-A33 Assignment 5 % gcc -g htags4.c -o htags4
[sahebsa@Vauren-s-A33 Assignment 5 % ./htags4 -m myOwn.html
Allocated 23 bytes. Total allocated memory: 23 bytes.
Allocated 23 bytes. Total allocated memory: 46 bytes.
Allocated 24 bytes. Total allocated memory: 70 bytes.
Allocated 25 bytes. Total allocated memory: 95 bytes.
Allocated 24 bytes. Total allocated memory: 119 bytes.
Allocated 23 bytes. Total allocated memory: 142 bytes.
Allocated 21 bytes. Total allocated memory: 163 bytes.
Allocated 22 bytes. Total allocated memory: 185 bytes.
Allocated 20 bytes. Total allocated memory: 205 bytes.
Allocated 21 bytes. Total allocated memory: 226 bytes.
Allocated 24 bytes. Total allocated memory: 250 bytes.
Allocated 24 bytes. Total allocated memory: 274 bytes.
<html>
<head>
<title>
</title>
</head>
<body>
<h1>
</h1>
<p>
</p>
</body>
</html>
sahebsa@Vauren-s-A33 Assignment 5 % 
```