

ENG EK 125 - Worksheet C Chapter 5B

Name: Ande Chen

Section: C1

1) The following program will be your starting point. Type it in and execute it to make sure that you entered it correctly (note: there is no output yet).

```
#include <stdio.h>
#include <stdlib.h>

typedef struct linked_list
{
    int data;
    struct linked_list *next;
} element;

typedef element * elementptr;

int main()
{
    elementptr first = NULL,
               last = NULL;

    /* Create a linked list with one element */
    /* NOTE: the first element is always a special case */

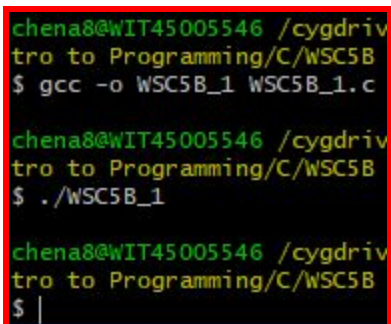
    first = (elementptr) malloc(sizeof(element));
    last = first;
    last -> data = 5;
    last -> next = NULL;

    /* Add another element to the end of the list */

    last -> next = (elementptr) malloc(sizeof(element));
    last = last -> next;
    last -> data = 12;
    last -> next = NULL;

    free(first);
    free(last);

    return 0;
}
```

A terminal window with a black background and green text. The prompt is 'chena8@WIT45005546 /cygdrive'. The user enters 'tro to Programming/C/WSC5B', then '\$ gcc -o WSC5B_1 WSC5B_1.c'. The prompt changes to 'chena8@WIT45005546 /cygdrive'. The user enters 'tro to Programming/C/WSC5B', then '\$./WSC5B_1'. The prompt changes to 'chena8@WIT45005546 /cygdrive'. The user enters 'tro to Programming/C/WSC5B', then '\$ |'.

```
chena8@WIT45005546 /cygdrive
tro to Programming/C/WSC5B
$ gcc -o WSC5B_1 WSC5B_1.c

chena8@WIT45005546 /cygdrive
tro to Programming/C/WSC5B
$ ./WSC5B_1

chena8@WIT45005546 /cygdrive
tro to Programming/C/WSC5B
$ |
```

2) Add a function *trav_and_print* to your program that will traverse the linked list and print the values of all of the data members.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct linked_list
5  {
6      int data;
7      struct linked_list *next;
8  }   element;
9
10 typedef element * elementptr;
11
12 void trav_and_print(elementptr);
13
14 int main()
15 {
16     elementptr first = NULL,
17                 last = NULL,
18                 current;
19
20     /* Create a linked list with one element */
21     /* NOTE: the first element is always a special case */
22
23     first = (elementptr) malloc(sizeof(element));
24     current = first;
25     last = first;
26     last -> data = 5;
27     last -> next = NULL;
28
29     /* Add another element to the end of the list */
30
31     last -> next = (elementptr) malloc(sizeof(element));
32     last = last -> next;
33     last -> data = 12;
34     last -> next = NULL;
35
36     free(first);
37     free(last);
38
39     return 0;
40 }
41
42 void trav_and_print(elementptr current)
43 {
44     while(current != NULL)
45     {
46         printf("The data value is %d.\n", current -> data);
47         current = current -> next;
48     }
49 }
```

3) Add code to your program that will add another element to the end of the linked list. Prompt the user for the data value. Call your *trav_and_print* function, and also from **main** print the value of *last* -> *data* to make sure that the element was added correctly.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct linked_list
5  {
6      int data;
7      struct linked_list *next;
8  } element;
9
10 typedef element * elementptr;
11
12 void trav_and_print(elementptr);
13
14 int main()
15 {
16     elementptr first = NULL,
17                 last = NULL,
18                 current;
19     int counter;
20
21     /* Create a linked list with one element */
22     /* NOTE: the first element is always a special case */
23
24     first = (elementptr) malloc(sizeof(element));
25     current = first;
26     last = first;
27     last -> data = 5;
28     last -> next = NULL;
29
30     /* Add another element to the end of the list */
31
32     last -> next = (elementptr) malloc(sizeof(element));
33     last = last -> next;
34     last -> data = 12;
35     last -> next = NULL;
36
37     last -> next = (elementptr) malloc(sizeof(element));
38     last = last -> next;
39     printf("Enter an integer data value: ");
40     scanf("%d", last);
41     last -> data;
42     last -> next = NULL;
43
44     trav_and_print(current);
45
46     printf("%d\n", last->data);
47
48     free(first);
49     free(last);
50
51     return 0;
52 }
53
54 void trav_and_print(elementptr current)
55 {
56     while(current != NULL)
57     {
58         printf("The data value is %d.\n", current -> data);
59         current = current -> next;
60     }
61 }
```

```
chena8@WIT45005546 /cygdrive/c/
tro to Programming/C/WSC5B
$ ./WSC5B_3
Enter an integer data value: 3
The data value is 5.
The data value is 12.
The data value is 3.
3
```

4) Add another function `count_elems` that will traverse the linked list, count how many elements there are, and return that count. Test this function by calling it in various places in `main`, including before there are any elements in the linked list.cy

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct linked_list
5  {
6      int data;
7      struct linked_list *next;
8  } element;
9
10 typedef element * elementptr;
11
12 void trav_and_print(elementptr);
13 int count_elems(elementptr, int);
14
15 int main()
16 {
17     elementptr first = NULL,
18                 last = NULL,
19                 current;
20     int counter = 0;
21
22     /* Create a linked list with one element */
23     /* NOTE: the first element is always a special case */
24
25     first = (elementptr) malloc(sizeof(element));
26     current = first;
27     last = first;
28     last -> data = 5;
29     last -> next = NULL;
30
31     /* Add another element to the end of the list */
32
33     last -> next = (elementptr) malloc(sizeof(element));
34     last = last -> next;
35     last -> data = 12;
36     last -> next = NULL;
37
38     last -> next = (elementptr) malloc(sizeof(element));
39     last = last -> next;
40     printf("Enter an integer data value: ");
41     scanf("%d", last);
42     last -> data;
43     last -> next = NULL;
44
45     trav_and_print(current);
46     counter = count_elems(current, counter);
47
48     printf("%d\n", last->data);
49     printf("There are %d elements.\n", counter);
50
51     free(first);
52     free(last);
53
54     return 0;
55 }
56
57 void trav_and_print(elementptr current)
58 {
59     while(current != NULL)
60     {
61         printf("The data value is %d.\n", current -> data);
62         current = current -> next;
63     }
64 }
65
66 int count_elems (elementptr current, int counter)
67 {
68     while(current != NULL)
69     {
70         current = current -> next;
71         counter = ++counter;
72     }
73     return counter;
74 }
```

```
chena8@WIT45005546 /cygdrive/c/
tro to Programming/C/WSC5B
$ ./WSC5B_4
Enter an integer data value: 3
The data value is 5.
The data value is 12.
The data value is 3.
3
There are 3 elements.
```