

Question 7

```
main.cpp x q7.cpp x
15 | printf("\n"); //initial brackets
16 | while(temp->next != NULL) { //till the end is reached
17 |     if(temp->sublist != NULL) { //if a sublist is there
18 |         struct Node *temp2 = temp->sublist; //store the address of the sublist in temp2
19 |         printf("\n"); //initial brackets
20 |         while(temp2 != NULL) { //till the end is reached
21 |             printf("%c", temp2->data); //print the data
22 |             temp2 = temp2->next; //move to next node
23 |         }
24 |         printf("\n"); //closing brackets
25 |     } else { //if sublist is not there
26 |         if(temp->data != '\0') { //if data is to be printed, skip the first node data
27 |             else printf("%c", temp->data); //else print the data
28 |         }
29 |         temp = temp->next; //move to the next node
30 |     }
31 |     printf("\n"); //closing brackets
32 | }
33 |
34 | //function to print CDM of the linked list
35 | void ca(struct Node *head) {
36 |     printf("cdm of head node\n"); //print the data of the head node
37 | }
38 | //function to print CDM of the linked list
39 | void cd(struct Node *head) {
40 |     printf("cdm of head node\n"); //print the data of the head node
41 | }
42 |
43 | //function to free up the space of the linked list
44 | void freelist(struct Node *head) {
45 |     struct Node *temp = head; //store head node
46 |     while(temp != NULL) { //till the end is reached
47 |         if(temp->sublist != NULL) { //if sublist is there
48 |             struct Node *temp2 = temp->sublist; //store the sublist node
49 |             while(temp2 != NULL) { //till the end is reached
50 |                 struct Node *t1 = temp2; //store the address in a new variable
51 |                 temp2 = temp2->next; //move to next node
52 |                 free(t1); //free the current node
53 |             }
54 |             struct Node *t2 = temp; //store the address in a new variable
55 |             temp = temp->next; //move to next node
56 |             free(t2); //free the current node
57 |         }
58 |         temp = temp->next; //move to next node
59 |     }
60 | }
61 |
62 | //main function
63 | int main() {
64 |     //creating a linked list as per the question. Here we are using malloc to dynamically allocate space for the nodes.
65 |     struct Node *head = new Node();
66 |     head->data = 'a';
67 |     struct Node *link = new Node();
68 |     link->next = link;
69 |     link->sublist = new Node();
70 |     link->sublist->data = 'b';
71 |     link->sublist->next = new Node();
72 |     link->sublist->next->data = 'c';
73 |     link->next = new Node();
74 |     link->next->data = 'd';
75 |     link->next->next = new Node();
76 |     link->next->next->data = 'e';
77 | }
78 | //print list
79 | printf("list\n");
80 | //print CDM
81 | ca(head);
82 | //print CDM
83 | cd(head);
84 | //free up the space
85 | freelist(head);
86 | }
```

C:\Users\Chelle PC\Documents\COMP 349\q7.exe

(a(bc)de)

Process exited after 0.01920 seconds with return value 0

Press any key to continue . . .

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Chelle PC\Documents\COMP 349\q7.exe
- Output Size: 556,170,021,25 B
- Compilation Time: 0.19s

C:\Users\Chelle PC\Documents\COMP 349\q7.exe

(a(bc)de)

Process exited after 0.01889 seconds with return value 0

Press any key to continue . . .

Question 8

```
main.cpp x | g++ x |
15 | printf("\n"); //initial brackets
16 | while(temp->next != NULL) { //fill the end is reached
17 |     if(temp->sublist != NULL) { //if a sublist is there
18 |         struct Node *temp2 = temp->sublist; //store the address of the sublist in temp2
19 |         printf("\n"); //initial brackets
20 |         while(temp2->next != NULL) { //fill the end is reached
21 |             printf("%c", temp2->data); //print the data
22 |             temp2 = temp2->next; //move to next node
23 |         }
24 |         printf("\n"); //closing brackets
25 |     } else { //if sublist is not there
26 |         if(temp->data != '\0') { //if data is to be printed, skip the first node data
27 |             else printf("%c", temp->data); //else print the data
28 |         }
29 |         temp = temp->next; //move to the next node
30 |     }
31 |     printf("\n"); //closing brackets
32 | }
33 |
34 | //function to print CDR of the linked list
35 | void car(struct Node *head) {
36 |     printf("cdr of head node\n"); //print the data of the head node
37 | }
38 | //function to print CDR of the linked list
39 | void cdr(struct Node *head) {
40 |     printfList(head, 1); //call printfList function with isCDR equals to 1
41 | }
42 |
43 | //function to free up the space of the linked list
44 | void freeList(struct Node *head) {
45 |     struct Node *temp=head; //store head node
46 |     while(temp->next != NULL) { //fill the end is reached
47 |         if(temp->sublist != NULL) { //if sublist is there
48 |             struct Node *temp2 = temp->sublist; //store the sublist node
49 |             while(temp2->next != NULL) { //fill the end is reached
50 |                 struct Node *t=temp2; //store the address in a new variable
51 |                 temp2 = temp2->next; //move to next node
52 |                 free(t); //free the current node
53 |             }
54 |             struct Node *t2 = temp; //store the address in a new variable
55 |             temp = temp->next; //move to next node
56 |             free(t2); //free the current node
57 |         }
58 |         free(temp); //free the current node
59 |     }
60 | }
61 |
62 | //main function
63 | int main() {
64 |     //creating a linked list as per the question. Here we are using malloc to dynamically allocate space for the nodes.
65 |     struct Node *head = new Node();
66 |     head->data = 'a';
67 |     struct Node *link = new Node();
68 |     head->next = link;
69 |     link->sublist = new Node();
70 |     link->sublist->data = 'b';
71 |     link->sublist->next = new Node();
72 |     link->sublist->next->data = 'c';
73 |     link->next = new Node();
74 |     link->next->data = 'd';
75 |     link->next->next = new Node();
76 |     link->next->next->data = 'e';
77 |
78 |     //print list
79 |     printfList(head, 0);
80 |     //print CDR
81 |     car(head);
82 |     //print CDR
83 |     cdr(head);
84 |     //free up the space
85 |     freeList(head);
86 | }
```

```
C:\Users\Chelle PC\Documents\COMP 348\main.exe
(a)(bc)(de)
a
((bc)(de))
-----
Process exited after 0.0188 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\Chelle PC\Documents\COMP 348\main.exe
(a)(bc)(de)
a
((bc)(de))
-----
Process exited after 0.0188 seconds with return value 0
Press any key to continue . . .
```