

COMPUTER PROGRAMMING : SECTION B

QUIZ 2

Instructor: Girish Varma • Course Code: CS0.101 • IIIT Hyderabad

Problems are in comments, highlighted.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <stdbool.h>
5
6 typedef enum RelStatus {
7     NotMentioned, Single, Engaged, Married
8 } RelStatus;
9
10 typedef struct Node Node;
11
12 typedef Node* LinkedList;
13
14 typedef struct Person {
15     char name[100]; int age;
16     RelStatus relstatus;
17     LinkedList friends;
18 } Person;
19
20 struct Node {
21     struct Person* data; struct Node* next;
22 };
23
24 typedef struct SocialNet {
25     LinkedList members;
26 } SocialNet;
27
28 LinkedList append(Person* p, LinkedList l) {
29     if (l == NULL) {
30         Node* D = (Node*) malloc(sizeof(Node));
31         D->data = p;
32         D->next = NULL;
33         return D;
34     } else {
35         l->next = append(p, l->next);
36     }
37     return l;
38 }
39
40 void print_person(Person* p) {
41     char status_string[] [15] = {
42         "Not Mentioned", "Single", "Married", "Engaged"
43     };
44     printf("%s\t%d\t%s\t%d\t",
45            p->name, p->age, status_string[p->relstatus]);
46     LinkedList f = p->friends;
47     while (f != NULL) {
48         printf("%s, ", f->data->name);
49         f = f->next;
50     }
51     printf("\n");
52 }
53
54 void print_network(LinkedList m) {
55     printf(
56     "Name\tAge\tStatus\tFriends\n"
57     "-----\n");
58     while (m != NULL) {
59         print_person(m->data);
60         m = m->next;
61     }
62     printf(
63     "-----\n");
64 }
65
66 Person* find_person(char* name, LinkedList l) {
67     // Either find the person with a particular name
68     // if not found return NULL
69     while(l != NULL) {
70         if (strcmp(l->data->name, name) == 0) {
71             return l->data;
72         }
73     }
74     l = l->next;
75 }
76
77 return NULL;
78 }
79
80 int popularity(char* name, LinkedList l) {
81     // Q1: Return the number of people who has the person
82     // named 'name' among their friends. (3 marks)
83 }
84
85 LinkedList filterby_age(LinkedList l, int lower, int upper) {
86     // Q2: Return the link list of people in l with age
87     // between lower and upper (3 marks)
88 }
89
90 bool transitive_friendship(LinkedList members) {
91     // Q3: check if the friendship relation is transitive
92     // ie for any X, Y, Z, if Y is a friend of X and
93     // Z is a friend of Y then Z is a friend of X
94     // Also print all the links that violates transitivity
95     // (4 marks)
96 }
97
98 int main()
99 {
100     SocialNet s = { NULL };
101     Person A = {"Alice", 23, Single, NULL};
102     Person B = {"Bob", 26, Engaged, NULL};
103     Person C = {"Charlie", 21, NotMentioned, NULL};
104     Person D = {"Don", 28, Married, NULL};
105
106     s.members = append(&A, s.members);
107     s.members = append(&B, s.members);
108     s.members = append(&C, s.members);
109     s.members = append(&D, s.members);
110
111     A.friends = append(&B, A.friends);
112     A.friends = append(&C, A.friends);
113     B.friends = append(&D, B.friends);
114     C.friends = append(&D, C.friends);
115     D.friends = append(&A, D.friends);
116
117     // prints
118     // -----
119     // Name      Age   Status    Friends
120     // -----
121     // Bob       26    Married   Don,
122     // Don       28    Engaged   Alice,
123     // -----
124     print_network(filterby_age(s.members, 24, 28));
125
126     // For the above social network,
127     // 'transitive_friendship(s.members)'
128     // returns false and prints
129     // -----
130     // Links that are not Transitive
131     // -----
132     // Alice->Bob->Don, but there is no Alice->Don
133     // Alice->Charlie->Don, but there is no Alice->Don
134     // Bob->Don->Alice, but there is no Bob->Alice
135     // Charlie->Don->Alice, but there is no Charlie->Alice
136     // Don->Alice->Bob, but there is no Don->Bob
137     // Don->Alice->Charlie, but there is no Don->Charlie
138
139     transitive_friendship(s.members);
140
141 }
```