

COMPUTER PROGRAMMING

QUIZ 2

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

Problems are in comments.

Name:

Roll No:

Tie the question paper to the answer sheet.

For Evaluators only

No.	1	2	3	4	Evaluator
Marks					

Total: / 50

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #include <string.h>
5  // defines strcmp(first_str, second_str);
6  // if 'first_str' is equal to 'second_str'
7  // returns 0. Otherwise returns a nonzero value
8
9  #include <stdbool.h>
10 // defines type 'bool' and macros 'true' and 'false'
11
12 typedef enum RelStatus {
13     NotMentioned,
14     Single,
15     Engaged,
16     Married
17 } RelStatus;
18
19 typedef struct Node Node;
20
21 typedef Node* LinkedList;
22
23 typedef struct Person {
24     char name[100];
25     int age;
26     RelStatus relstatus;
27     LinkedList friends;
28 } Person;
29
30 struct Node {
31     struct Person* data;
32     struct Node* next;
33 };
34
35 typedef struct SocialNet {
36     LinkedList members;
37 } SocialNet;
38
39 LinkedList append(Person* p, LinkedList l) {
40     if (l == NULL) {
41         Node* D = (Node*) malloc(sizeof(Node));
42         D->data = p;
43         D->next = NULL;
44         return D;
45     } else {
46         l->next = append(p, l->next);
47     }
48     return l;
49 }
50
51 int size(LinkedList l) {
52     return l==NULL? 0: 1+ size(l->next);
53 }
54
55 Person* find_person_by_name(char* name, LinkedList l) {
56     // Q1: Return the pointer to the Person with name
57     // given by argument 'name' in the LinkedList 'l'
58     // (10 marks)
59 }
60
61 bool common_single_friend(char* name1, char* name2,
62     SocialNet* s) {
63     // Q2: Check if the Persons with name = name1
64     // and name = name2 has a common friend who
65     // is Single. Return 'true' or 'false'
66     // (10 marks)
67 }
68
69 char* most_popular_person(SocialNet* s) {
70     // Q3: Return the name of the person who is in the
71     // friends list of most number of people
72     // (15 marks)
73 }
74
75 bool all_members_with_only_two_young_friends(
76     SocialNet* s, int age_upper) {
77     // Q4: Check if all members in the social
78     // network 's' have exactly two friends
79     // whose age is <= 'age_upper'.
80     // Return 'true' or 'false'. (15 marks)
81 }
82 }
```