Consider the definition:
map<string,set<string> > FRIENDS;

1) What does FRIENDS.count("Amit") return?  
1 point

- The number of friends Amit has, as stored in FRIENDS.
- Whether any friends are stored for Amit in FRIENDS, (0: none, 1:some).
- This expression is invalid.

No, the answer is incorrect.  
Score: 0

Accepted Answers:  
Whether any friends are stored for Amit in FRIENDS, (0: none, 1:some).

2) What does FRIENDS["Amit"].size() return?  
1 point

- How many friends Amit has, as stored in FRIENDS.
- Whether any friends are stored for Amit in FRIENDS, (0: none, 1:some).
- This expression is invalid.

No, the answer is incorrect.  
Score: 0

Accepted Answers:  
How many friends Amit has, as stored in FRIENDS.
3) The following code is supposed to print the friends of Amit.
   for(auto f : FRIENDS["Amit"]) cout << blank << ' ';
What should blank be?

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) f

4) Suppose I want to store the information that Dharm is a friend of Amit. Which of the following statements will accomplish this?

- FRIENDS["Amit"] = "Dharm";
- FRIENDS["Amit"].insert("Dharm");
- FRIENDS["Amit"]["Dharm"] = True;
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
FRIENDS["Amit"].insert("Dharm");

5) What expression would you write to determine if Dharm is a friend of Amit?
Note: While answering the question posed, please do not specify extra spaces in the answer you give.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) FRIENDS["Amit"].count("Dharm")
(Type: String) FRIENDS["Amit"].count("Dharm")>0

The following defines a data structure CHILDREN, which stores the children of an individual.
typedef map<string,vector<string> > onemany;
onemany CHILDREN;
CHILDREN["X"] = vector<string>({"X1","X2"});
// X has children X1, X2
CHILDREN["Y"] = vector<string>({"Y1","Y2","Y3"});
CHILDREN["X2"] = vector<string>({"X21","X22"});
CHILDREN["Y3"] = vector<string>({"Y31"});
We now want to write a function which prints the contents of this data structure in a nice way, i.e. it should print lines consisting of the name of a person, then a ':', and then the names of the children.
Here is the function that does it.
void print(onemany CHILDREN){
  for(auto mapentry : CHILDREN){
    cout << blank1 << ':';
    for(auto child : blank2)
      cout <<' '<<blank3;
    cout << endl;
  }
}
6) What should blank1 be?

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) mapentry.first

7) What should blank2 be?

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) mapentry.second

8) What should blank3 be?

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) child

Suppose in a certain circuit node 5 is connected to nodes 6, 7, and 8 respectively using conductances of magnitude 10, 15 and 20. Suppose node 5 is not connected to any other nodes. Assume that the 'A' used in the questions below is as defined in the lectures.

9) What will be the value of A[5][6]?

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) -10

10) What will be the value of A[5][5]?

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) 45