Q1,

Sol:

According to the description of this problem, it is hard to generating such a participants list from scratch, because every employee will have two conditions: invited or uninvited, which means we will explore 2^n cases, and it cannot be solved in polynomial-time. However, it is esay to check if a given participants list is satisfactory, and this process can be finished in polynomial-time. So, this problem is a NP problem.
Since it is a NP problem, we would like to give up on optimality, and will use an approximation algorithm that gives "good enough" results.

Greedy algorithm:

1. Initialize two lists: participants list, and employees list.
2. Sort all employees in descending order based on their "likability".
3. Put sorted employees into employees list.
4. Select the first element of employees list (called A), and put it into participants list, and then remove A, and direct boss and subordinates of A from employees list.
5. Repeat step 1 ~ 4, until employees list is empty.

And to solve the example problem:

i), participants list: [ ]
employees list: [ ]

ii), participants list: [ ]
employees list: [5, 0, 2, 4, 3, 1]

iii), participants list : [5]

employees list : [0, 2, 3, 1]

iv), participants list : [0, 5]

employees list : [1]

v), participants list : [0, 1, 5]

employees list : []

Therefore, the participants list is : 0, 1, 5, and the maximum likability is 20 + 3 + 30 = 53.

Q2,

Sol :

1),

a, The set of clauses satisfying the formula $x = \bar{y}$ is :

$U = \{\bar{x} \vee \bar{y}, x \vee y\} = (\bar{x} \vee \bar{y}) \wedge (x \vee y)$

Since the formula $x = y$ is equivalent to $(\bar{x} \vee y) \wedge (x \vee \bar{y})$, this equivalence can be checked using a truth table.

b, The set of clauses satisfying the formula $x = y \wedge z$ is :

$U = \{\bar{x} \vee (y \wedge z), x \vee \overline{(y \wedge z)}\}$

$= (\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (x \vee \bar{y} \vee \bar{z})$

Since the formula $x = y$ is equivalent to $(\bar{x} \vee y) \wedge (x \vee \bar{y})$, this equivalence can be checked using a truth table.

C) The set of clauses satisfying the formula $x = y \vee z$ is:

$$U = \{ \bar{x} \cup (y \vee z),\ x \cup \overline{(y \vee z)} \}$$

$$= (\bar{x} \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (x \vee \bar{z})$$

Since the formula $x = y$ is equivalent to $(\bar{x} \vee y) \wedge (x \vee \bar{y})$, this equivalence can be checked using a truth table.

2),

The CIRCUIT-SAT and CNF-SAT consists of gates of AND, OR, and NOT, those gates takes one or two literals as input.

We can write the expression of CIRCUIT-SAT as:

$$x_1 \wedge x_2 \wedge \cdots \wedge x_n \wedge (x_n = 1)$$

where x_i is the output of i-th gate, and x_n must be 1 if circuit is satisfied.

According to the results of (1), each of the clauses can be converted into CNF form. So, we find a way that transform the CIRCUIT-SAT to CNF-SAT problem in polynomial-time, because the gates number is linear.

Therefore, the CIRCUIT-SAT is reducible to CNF-SAT.