Data structure Workshop 1

I. Exercise

Let n be an integer, find out the big O time complexity of each piece of code.

```
(1) i=1; k=0;
        while(i \le n-1)
             k+=10*i;
             i++;}
(2) i=1; k=0;
        do {
             k+=10*i;
             i++;
          \}while(i<=n-1);
(3) i=1; k=0;
        while(i \le n-1)
                i++;
                k+=10*i;}
(4)
       k=0;
       for(i=1;i \le n;i++){
          for(j=i;j \le n;j++)
                k++;}
(5) for(i=1;i<=n;i++){
          for(j=1;j<=i;j++){
               for(k=1;k<=j;k++)
                   x+=delta;
   }}
(6) i=1; j=0;
      while(i+j \le n){
           if(i>j) j++;
                else i++;
    }
(7) x=n; y=0; //n is greater or equal than 1
      while(x \ge (y+1)*(y+1)){
           y++;
(8) x=91; y=100;
      while(y>0){
           if (x>100)
                     \{x=10; y=:\}
               else x++;
      }
```

II. Experiment

Problem Description

A software company has n employees, and each employee has attributes such as name, position, and employee ID. You are required to store all employees' information using a **fixed-size array**, and implement functions for employee resignation, onboarding, and information query. When an employee resigns or joins the company, modify the content in the array and output the result in accordance with the requirements specified in the output format.

Input Format

The first line contains two positive integers, n and m.

The next n lines each contain information about one employee, including their name, position, and employee ID (id).

The next m lines each start with two positive integers, op and x:

- If op = 1, it indicates that the employee at position x in the array resigns.
- If op = 2, it indicates that a new employee is onboarded and needs to be inserted at position x in the array. Following this, the new employee's name, position, and id will be input.
- If op = 3, it indicates a query for the information of the employee at position x.

There may be **invalid operations** (e.g., illegal position x). In such cases, output the required result as specified and ignore the invalid operation.

Input Constraints

- $1 \le n, m \le 1000$
- The initial size of the array is fixed as n + m (sufficient to accommodate all possible operations).
- The length of "name" and "position" does not exceed 50 characters.
- The employee ID (id) is an integer satisfying 1 ≤ id ≤ 1,000,000.
- x is a positive integer and will not exceed the value range of the integer type.
- No two employees in the input have identical information (i.e., all employee records are unique).

Output Format

For each operation, output one line:

- If position x is **out of the range of the actual number of current employees**, output -1.
- Otherwise:
 - For resignation (op=1) and onboarding (op=2) operations: Output the information of all actual employees in the array after the operation, in the following format:

```
name1 position1 id1 name2 position2 id2
```

...

name m position m id m

(Here, m represents the actual number of current employees.)

 For query operations (op=3): Output the information of the employee at the specified position, in the following format: namex positionx idx

You may start coding like this..

.....