//Laura Casas Torres

#include <iostream>

#include <fstream>

#include <algorithm>

#include <sstream>

#include <cctype>

#include <iterator>

#include <functional>

using namespace std;

#include "IndexPQ.h"

class Usuario {

public:

int id;

int periodo;

int prioridad;

Usuario() {};

Usuario(int i, int p, int pr) : id(i), periodo(p), prioridad(pr) {};

friend bool operator<(Usuario a, Usuario b);

};

bool operator<(Usuario a, Usuario b) {

if (a.prioridad < b.prioridad) return true;

else if (a.prioridad == b.prioridad) {

if (a.id < b.id) return true;

else return false;

}

else return false;

}

// resuelve un caso de prueba

void resuelveCaso(int n) {

IndexPQ<Usuario, std::less<Usuario>> pq(n);

for (int i = 0; i < n; i++)

{

int id, periodo;

cin >> id;

cin >> periodo;

Usuario usuario(id, periodo, periodo);

pq.push(i, usuario);

}

int envios = 0;

cin >> envios;

for (int i = 0; i < envios; i++) {

IndexPQ<Usuario, std::less<Usuario>>::Par p = pq.top();

cout << p.prioridad.id << endl;

p.prioridad.prioridad = p.prioridad.prioridad + p.prioridad.periodo;

pq.update(p.elem, p.prioridad);

}

cout << "----" << endl;

}

int main() {

// ajuste para que cin extraiga directamente de un fichero

#ifndef DOMJUDGE

std::ifstream in("Casos.txt");

auto cinbuf = std::cin.rdbuf(in.rdbuf());

#endif

int numUsuarios = 0;

cin >> numUsuarios;

while (numUsuarios != 0) {

resuelveCaso(numUsuarios);

cin >> numUsuarios;

}

// restablecimiento de cin

#ifndef DOMJUDGE

std::cin.rdbuf(cinbuf);

system("pause");

#endif

return 0;

}