

26043 - 식당 메뉴

26043 - 식당 메뉴

문제

Queue

여러 명의 학생이 식사하기 위하여 학교 식당을 향해 달려가고 있다. 학교 식당에 도착한 학생은 식당 입구에 줄을 서서 대기한다. 학교 식당에 먼저 도착한 학생이 나중에 도착한 학생보다 식당 입구의 앞쪽에서 대기한다. 식사는 1인분씩 준비된다. 식사 1인분이 준비되면 식당 입구의 맨 앞에서 대기 중인 학생 1명이 식당으로 들어가서 식사를 시작한다. 식사를 시작한 학생은 항상 식사를 마친다.

학교 식당에서는 두 가지 메뉴가 제공되고 각각의 학생은 두 가지 메뉴 중에서 본인이 좋아하는 메뉴를 결정한 상태다. 학생이 학교 식당에 도착하고 식사가 준비되는 n 개의 정보가 저장된 S 가 주어진다. S 에 저장된 첫 번째 정보부터 n 번째 정보까지 순서대로 처리한 경우, 본인이 좋아하는 메뉴를 먹은 학생 목록 A 와 본인이 좋아하지 않는 메뉴를 먹은 학생 목록 B 와 학교 식당에 도착하였으나 식사를 하지 못한 학생 목록 C 를 출력하자.

S 에 저장된 n 개의 식당 정보는 아래 두 가지 유형으로 구분된다. 첫 번째가 유형 1, 두 번째가 유형 2다.

- 1 a b : 학생 번호가 양의 정수 a 이고 좋아하는 메뉴 번호가 양의 정수 b 인 학생 1명이 학교 식당에 도착하여 식당 입구의 맨 뒤에 줄을 서기 시작한다.
- 2 b : 메뉴 번호가 양의 정수 b 인 식사 1인분이 준비되어 식당 입구의 맨 앞에서 대기 중인 학생 1명이 식사를 시작한다.

식사 1인분이 준비될 때는 식당 입구에서 대기 중인 학생이 항상 존재한다. 식당 입구에 줄을 서서 대기하였으나 식사가 준비 안 된 학생은 식사를 못 한다.

26043 - 식당 메뉴

문제

Queue

여러 명의 학생이 식사하기 위하여 학교 식당을 향해 달려가고 있다. 학교 식당에 도착한 학생은 식당 입구의 앞쪽에서 대기한다. 식사는 1인분씩 준비된다. 식사 1인분이 준비되면 식당 입구에서 대기 중인 학생은 항상 식사를 마친다.

학교 식당에서는 두 가지 메뉴가 제공되고 각각의 학생은 두 가지 메뉴 중에서 본인이 좋아하는 메뉴를 결정한 상태다. 학생이 학교 식당에 도착하고 식사가 준비되는 n 개의 정보가 저장된 S 가 주어진다. S 에 저장된 첫 번째 정보부터 n 번째 정보까지 순서대로 처리한 경우, 본인이 좋아하는 메뉴를 먹은 학생 목록 A 와 본인이 좋아하지 않는 메뉴를 먹은 학생 목록 B 와 학교 식당에 도착하였으나 식사를 하지 못한 학생 목록 C 를 출력하자.

S 에 저장된 n 개의 식당 정보는 아래 두 가지 유형으로 구분된다. 첫 번째가 유형 1, 두 번째가 유형 2다.

- 1 a b : 학생 번호가 양의 정수 a 이고 좋아하는 메뉴 번호가 양의 정수 b 인 학생 1명이 학교 식당에 도착하여 식당 입구의 맨 뒤에 줄을 서기 시작한다.
- 2 b : 메뉴 번호가 양의 정수 b 인 식사 1인분이 준비되어 식당 입구의 맨 앞에서 대기 중인 학생 1명이 식사를 시작한다.

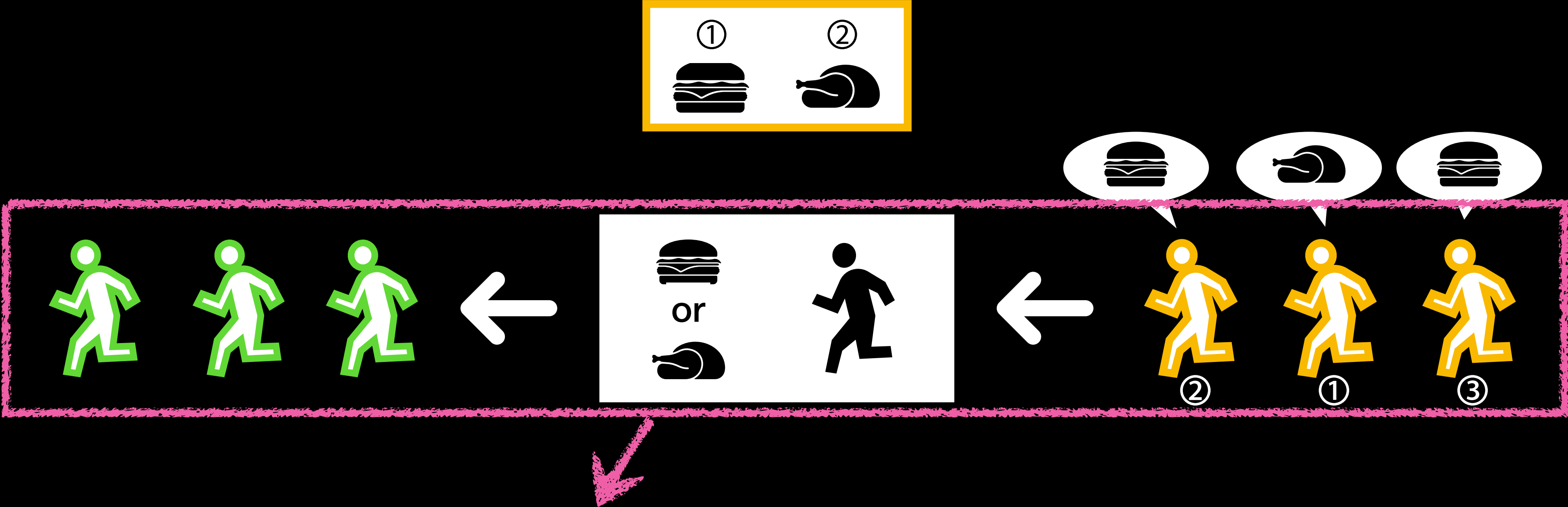
식사 1인분이 준비될 때는 식당 입구에서 대기 중인 학생이 항상 존재한다. 식당 입구에 줄을 서서 대기하였으나 식사가 준비 안 된 학생은 식사를 못 한다.

예제 입력 1 복사

```
6
1 2 1
1 1 1
2 1
1 3 2
2 2
2 2
```

에 도착한 학생보
. 식사를 시작한

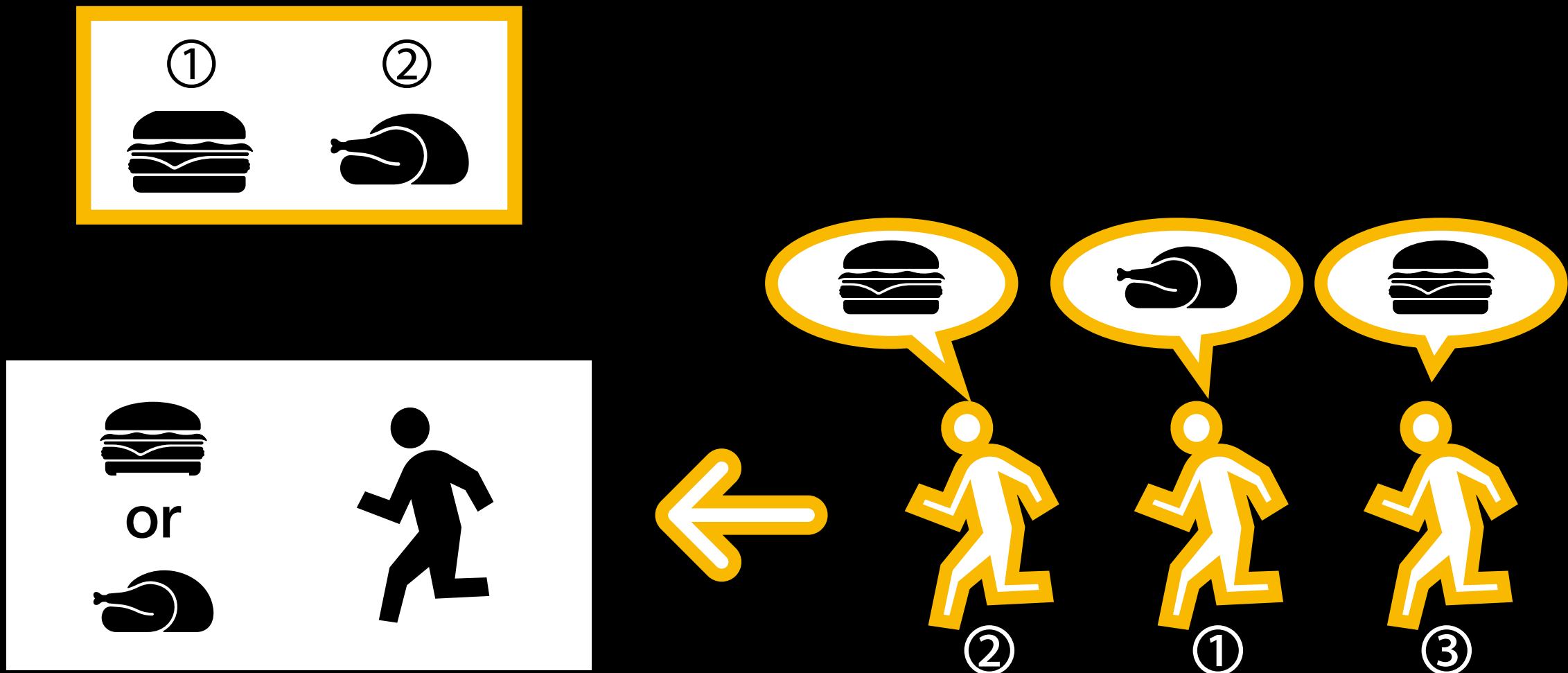
26043 - 식당 메뉴



Queue

26043 - 식당 메뉴

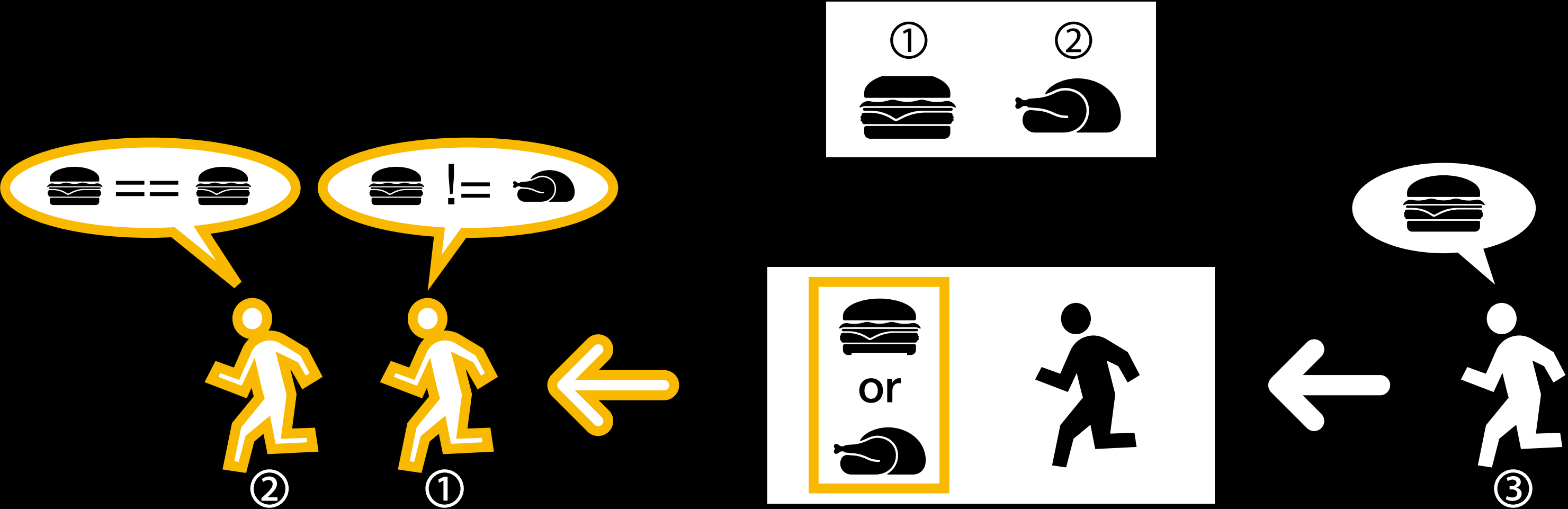
입력: 정보 S - 유형 1



1	3	2
유형번호	학생번호	음식번호

26043 - 식당 메뉴

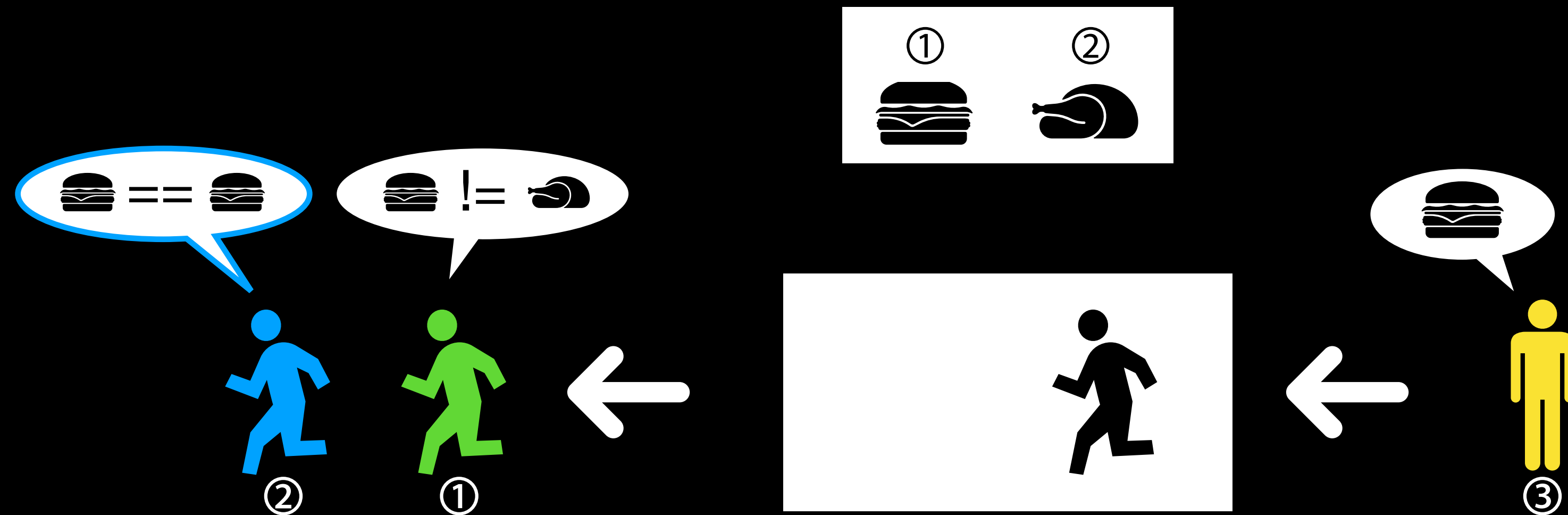
입력: 정보 S - 유형 2



2	1
유형번호	음식번호

26043 - 식당 메뉴

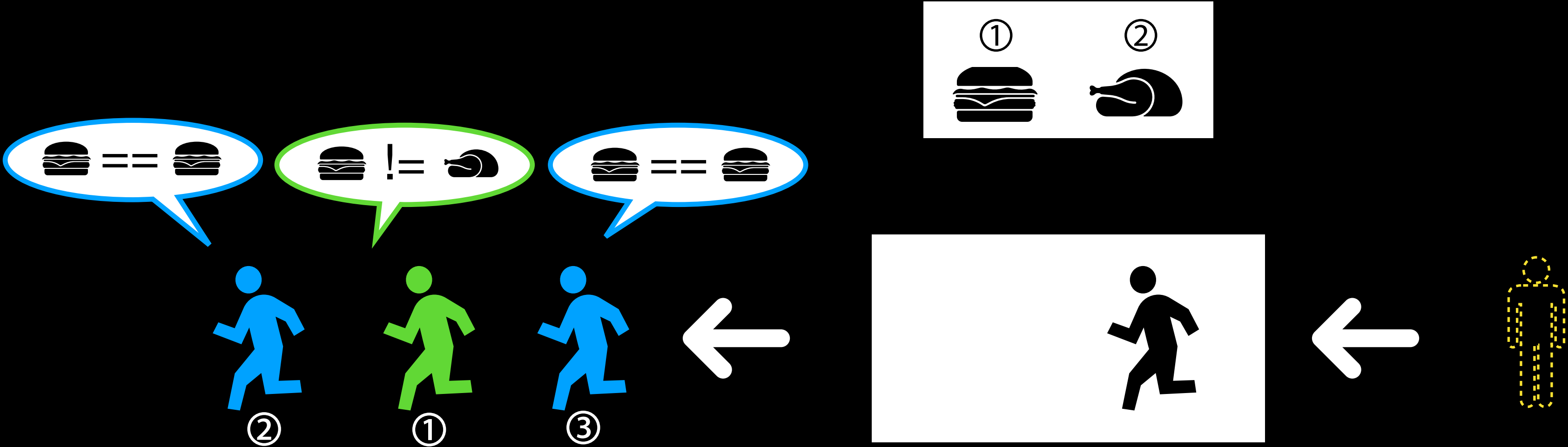
출력: 목록 A, B, C



목록 A : 좋아하는 메뉴 먹음 : 2
목록 B : 좋아하는 메뉴 못먹음 : 1
목록 C : 밥을 못먹음 : 3

26043 - 식당 메뉴

출력: 목록 A, B, C



목록 A : 좋아하는 메뉴 먹음 : 2 3
목록 B : 좋아하는 메뉴 못먹음 : 1
목록 C : 밥을 못먹음 : None

26043 - 식당 메뉴

```
int main() {
    int n, type, i;
    int id, desired, supplied;
    queue<Student> stand;
    vector<int> list_a, list_b, list_c;

    cin >> n;
    for (i = 0; i < n; i++) {
        cin >> type;
        if (type == 1) {          // 유형1
            cin >> id >> desired;
            stand.push(v: Student(id, desired));
        } else if (type == 2) {   // 유형2
            cin >> supplied;
            if (stand.front().get_desired() == supplied)
                list_a.push_back(stand.front().get_id());
            else if (stand.front().get_desired() != supplied)
                list_b.push_back(stand.front().get_id());
            stand.pop();
        }
    }

    while (!stand.empty()) {
        list_c.push_back(stand.front().get_id());
        stand.pop();
    }

    print_answer(list: list_a);
    print_answer(list: list_b);
    print_answer(list: list_c);
    return (0);
}
```

```
class Student {
private:
    int id;
    int desired;
public:
    Student(int _id, int _desired) {
        id = _id;
        desired = _desired;
    }

    int get_id() const {
        return (id);
    }

    int get_desired() const {
        return (desired);
    }
};
```

```
void print_answer(vector<int> list) {
    size_t len, i;

    if (list.empty()) {
        cout << "None" << endl;
        return;
    }
    sort(first: list.begin(), last: list.end());
    len = list.size();
    for (i = 0; i < len; i++) {
        if (i != 0)
            cout << " ";
        cout << list[i];
    }
    cout << endl;
}
```

```
6
1 2 1
1 1 1
2 1
1 3 2
2 2
2 2
2 3
1
None
```

26043 - 식당 메뉴

```
int main() {
    int n, type, i;
    int id, desired, supplied;
    queue<Student> stand;
    vector<int> list_a, list_b, list_c;

    cin >> n;
    for (i = 0; i < n; i++) {
        cin >> type;
        if (type == 1) {          // 유형1
            cin >> id >> desired;
            stand.push(v: Student(id, desired));
        } else if (type == 2) {   // 유형2
            cin >> supplied;
            if (stand.front().get_desired() == supplied)
                list_a.push_back(stand.front().get_id());
            else if (stand.front().get_desired() != supplied)
                list_b.push_back(stand.front().get_id());
            stand.pop();
        }
    }

    while (!stand.empty()) {
        list_c.push_back(stand.front().get_id());
        stand.pop();
    }

    print_answer(list: list_a);
    print_answer(list: list_b);
    print_answer(list: list_c);
    return (0);
}
```

```
class Student {
private:
    int id;
    int desired;
public:
    Student(int _id, int _desired) {
        id = _id;
        desired = _desired;
    }

    int get_id() const {
        return (id);
    }

    int get_desired() const {
        return (desired);
    }
};
```

```
void print_answer(vector<int> list) {
    size_t len, i;

    if (list.empty()) {
        cout << "None" << endl;
        return;
    }
    sort(first: list.begin(), last: list.end());
    len = list.size();
    for (i = 0; i < len; i++) {
        if (i != 0)
            cout << " ";
        cout << list[i];
    }
    cout << endl;
}
```

```
6
1 2 1
1 1 1
2 1
1 3 2
2 2
2 2
2 3
1
None
```

26043 - 식당 메뉴

```
int main() {
    int n, type, i;
    int id, desired, supplied;
    queue<Student> stand;
    vector<int> list_a, list_b, list_c;

    cin >> n;
    for (i = 0; i < n; i++) {
        cin >> type;
        if (type == 1) {          // 유형1
            cin >> id >> desired;
            stand.push(v: Student(id, desired));
        } else if (type == 2) {   // 유형2
            cin >> supplied;
            if (stand.front().get_desired() == supplied)
                list_a.push_back(stand.front().get_id());
            else if (stand.front().get_desired() != supplied)
                list_b.push_back(stand.front().get_id());
            stand.pop();
        }
    }

    while (!stand.empty()) {
        list_c.push_back(stand.front().get_id());
        stand.pop();
    }

    print_answer(list: list_a);
    print_answer(list: list_b);
    print_answer(list: list_c);
    return (0);
}
```

```
class Student {
private:
    int id;
    int desired;
public:
    Student(int _id, int _desired) {
        id = _id;
        desired = _desired;
    }

    int get_id() const {
        return (id);
    }

    int get_desired() const {
        return (desired);
    }
};
```

```
void print_answer(vector<int> list) {
    size_t len, i;

    if (list.empty()) {
        cout << "None" << endl;
        return;
    }
    sort(first: list.begin(), last: list.end());
    len = list.size();
    for (i = 0; i < len; i++) {
        if (i != 0)
            cout << " ";
        cout << list[i];
    }
    cout << endl;
}
```

```
6
1 2 1
1 1 1
2 1
1 3 2
2 2
2 2
2 3
1
None
```

26043 - 식당 메뉴

```
int main() {
    int n, type, i;
    int id, desired, supplied;
    queue<Student> stand;
    vector<int> list_a, list_b, list_c;

    cin >> n;
    for (i = 0; i < n; i++) {
        cin >> type;
        if (type == 1) {          // 유형1
            cin >> id >> desired;
            stand.push(v: Student(id, desired));
        } else if (type == 2) {   // 유형2
            cin >> supplied;
            if (stand.front().get_desired() == supplied)
                list_a.push_back(stand.front().get_id());
            else if (stand.front().get_desired() != supplied)
                list_b.push_back(stand.front().get_id());
            stand.pop();
        }
    }

    while (!stand.empty()) {
        list_c.push_back(stand.front().get_id());
        stand.pop();
    }

    print_answer(list: list_a);
    print_answer(list: list_b);
    print_answer(list: list_c);
    return (0);
}
```

```
class Student {
private:
    int id;
    int desired;
public:
    Student(int _id, int _desired) {
        id = _id;
        desired = _desired;
    }

    int get_id() const {
        return (id);
    }

    int get_desired() const {
        return (desired);
    }
};
```

```
void print_answer(vector<int> list) {
    size_t len, i;

    if (list.empty()) {
        cout << "None" << endl;
        return;
    }
    sort(first: list.begin(), last: list.end());
    len = list.size();
    for (i = 0; i < len; i++) {
        if (i != 0)
            cout << " ";
        cout << list[i];
    }
    cout << endl;
}
```

```
6
1 2 1
1 1 1
2 1
1 3 2
2 2
2 2
2 3
1
None
```