

FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110211 INTRODUCTIONS TO DATA STRUCTURE
Year II, First Semester, Midterm Examination, Sep 28, 2023 08:30-11:30

ชื่อ-นามสกุล.....Lucass.....เลขประจำตัว.....ตอนเรียนที่.....เลขที่ใน CR58.....

หมายเหตุ

1. ข้อสอบมีทั้งหมด 9 ข้อ ในกระดาษคำ答คำตอบ 11 หน้า
2. **ไม่อนุญาตให้นำเข้ามาและเอกสารใดๆ เข้าในห้องสอบ**
3. **ไม่อนุญาตให้ใช้เครื่องคำนวนใดๆ**
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่เจ้าหน้าที่ควบคุมการสอบจะหยิบยืมให้
5. ห้ามน้ำส่วนได้ส่วน失利ของข้อสอบและสมุดคำตอบออกจากห้องสอบ
6. ผู้เข้าสอบสามารถออกจากห้องสอบได้ หลังจากผ่านการสอบไปแล้ว 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. นิสิตกระทำการผิดเกี่ยวกับการสอบ ตามข้อบังคับจุฬาลงกรณ์มหาวิทยาลัย มหิดล คือ พ้นสภาพการเป็นนิสิต หรือ ได้รับ สัญลักษณ์ F ในรายวิชาที่กระทำการ และอาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

ห้ามนิสิตพกโทรศัพท์และอุปกรณ์สื่อสารไว้กับตัวระหว่างสอบ หากตรวจพบจะถือว่า นิสิตกระทำการผิดเกี่ยวกับการสอบ อาจต้องพ้นสภาพการเป็นนิสิต หรือ ให้ได้รับ F และ อาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

* ร่วมรณรงค์การไม่กระทำการผิดและไม่ทุจริตการสอบ *

ข้าพเจ้ายอมรับในข้อกำหนดที่กล่าวมานี้ ข้าพเจ้าเป็นผู้ทำข้อสอบนี้ด้วยตนเองโดยมิได้รับการช่วยเหลือ หรือให้ความช่วยเหลือในการทำข้อสอบนี้

ลงชื่อนิสิต.....
วันที่.....

- ใช้ดินสอเขียนคำตอบได้
- ให้เขียนเลขที่ในใบเซ็นชื่อเข้าสอบทุกหน้า
- หากพื้นที่สำหรับเขียนคำตอบไม่เพียงพอ ให้เขียนไว้ด้านหลังของหน้านี้ ห้ามเขียนข้ามไปหน้าอื่น และให้ระบุไว้ในพื้นที่สำหรับเขียนคำตอบว่า “มีต่อ ด้านหลัง”

1. (9 คะแนน) จากตัวแปร `a` ในประเภทต่าง ๆ ที่กำหนดไว้ในแต่ละข้อย่อย จงเขียนส่วนของโปรแกรมที่ทำการ “เปลี่ยน” เฉพาะข้อมูลประเภท `int` ทั้งหมดที่เก็บอยู่ในตัวแปร `a` ที่มีค่าเป็น `from` ให้กลายเป็นค่า `to` โดยที่ค่าอื่น ๆ ที่ไม่ตรงตามเงื่อนไขดังกล่าวจะต้องไม่หายไปและมีค่าคงเดิม (หากมีการลบค่าใดออกไป ก็อย่าลืมใส่ค่านั้นกลับเข้ามาด้วย)

1.0. ตัวอย่าง `vector<pair<bool,int>> a;`

```
for (auto &x : a)
    if (x.second == from)
        x.second = to;
```

1.1. `stack<vector<int>> a;`

```
stack<vector<int>> tmp;
while (!a.empty()) {
    auto v = a.top(); a.pop();
    for (auto &e : v) {
        if (e == from) e = to;
    }
    tmp.push(v);
}
while (!tmp.empty()) (a.push(tmp.top()); tmp.pop());
```

1.2. `priority_queue<pair<int,char>> a;`

```
priority_queue<pair<int,char>> tmp;
while (!a.empty()) {
    auto p = a.top(); a.pop();
    if (p.first == from) p.first = to;
    tmp.push(p);
}
a = tmp;
```

1.3. `map<int,vector<pair<char,int>>> a;` (ข้อนี้รับประกันว่า `a` ไม่มี key ที่มีค่าเป็น `to`)

```
for (auto &[i,v] : a)
    for (auto &[c,in] : v)
        if (in == from) in = to;
```

2. (4 คะแนน) ในแต่ละข้อย่อยต่อไปนี้ จงเติมคำสั่งลงในช่องว่างเพื่อให้ส่วนของโปรแกรมดังกล่าวทำงานตรงตามที่โจทย์กำหนดให้

2.1. มี `std::map<std::string, std::vector<int>> data` เราต้องการลบข้อมูลใน `data` ที่มีความยาวของ `vector` น้อยกว่า 5

```
for (auto it = data.begin(); it != data.end();) {
    if (it->second.size() < 5)
        it = data.erase(it);
    else
        ++it;
}
```

- 2.2. มี priority queue ที่เก็บข้อมูลประเภท Person ซึ่งเป็นประเภทที่เรากำหนดเอง โดย Person มีข้อมูลภายในเป็น age และ name เราต้องการให้ priority queue เรียงลำดับตาม age จากมากไปน้อย (กล่าวคือ ตัวที่อยู่บนสุดของ priority queue จะมี age มากสุด) และหากมีสอง persons ที่มีอายุเท่ากัน, ต้องเรียงลำดับตาม name จากน้อยไปมาก เราจะสร้าง priority queue ได้ดังต่อไปนี้

```
class ComparePersons {
    bool operator() (const Person& a, const Person& b)
        {return a.age < b.age || (a.age == b.age && a.name > b.name);}
};

std::priority_queue<Person, std::vector<Person>, ComparePersons> pq;
```

3. (6 คะแนน) ให้พิจารณาส่วนของโปรแกรมในแต่ละข้อย่อย แล้วระบุว่าผลลัพธ์ที่พิมพ์ออกหน้าจอคืออะไร

3.1.

```
std::map<int, std::vector<std::string>> data = {
    {1, {"apple", "cherry", "elderberry", "grape"}},
    {3, {"banana", "date", "fig", "apple"}},
    {4, {"guava", "kiwi", "mango"}},
    {5, {"banana", "lemon", "cherry"}};

std::set<int> keys = {1, 3, 5};
std::vector<std::string> fruitsToExclude = {"banana", "lemon", "kiwi"};
std::vector<std::string> result;
for (const auto& [key, fruits] : data)
    if (keys.find(key) != keys.end())
        for (const auto& fruit : fruits)
            if (std::find(fruitsToExclude.begin(), fruitsToExclude.end(), fruit) == fruitsToExclude.end())
                result.push_back(fruit);
std::sort(result.begin(), result.end());
for (auto &x : result) std::cout << x << " ";
```

ตอบ: apple apple cherry cherry date elderberry fig grape

3.2.

```
std::set<int> dataSet = {4, 10, 7, 2, 15, 12};
std::priority_queue<int, std::vector<int>, std::greater<int>> mq; ให้ mq แบบ
for (const auto& num : dataSet) {
    mq.push(num);
}
std::set<int> ex = {10, 2, 7};
while (!mq.empty() && ex.find(mq.top()) != ex.end()) {
    mq.pop();
}
std::cout << mq.top() << std::endl;
```

ตอบ: 4

✓ ✓ ✓

3.3.

```
vector<pair<int,char>> v = { {1,'a'}, {2,'a'}, {0,'b'}, {-2,'a'}, {-3,'b'}, {0,'c'} } ;
map<char,int> m;
for (auto &x : v) {
    if (m[x.second] == 0): ให้ m[0]
        m[x.second] = x.first;
    else
        m[x.second] = std::max(m[x.second],x.first);
}
for (auto &x : m)
    cout << x.first << ":" << x.second << " ";
```

ตอบ: a:2 b:-3 c:0

4. (4 คะแนน) ฟังก์ชันต่อไปนี้เป็นฟังก์ชัน ensureCapacity ของ CP::queue<T> ที่ “ไม่เหมือน” กับที่ได้เรียนมาในวิชา จงอธิบายว่า การทำ ensureCapacity ตามแบบนี้ ทำให้ CP::queue ทำงานถูกต้องหรือไม่ และมีข้อดีข้อเสียอย่างไร?

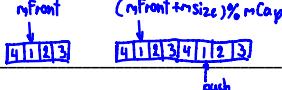
```

void ensureCapacity(size_t capacity) {
    if (capacity > mCap) {
        size_t s = (capacity > 2 * mCap) ? capacity : 2 * mCap;
        T *arr = new T[s]();
        for (size_t i = 0; i < mCap; i++)
            arr[i] = arr[(i+mCap)%mCap] = mData[i];
        delete [] mData;
        mData = arr;
        mCap = s;
    }
}

```

- 4.1. เมื่อนำ ensureCapacity นี้ไปใช้งาน CP::queue ยังคงทำงานถูกต้องหรือไม่สำหรับฟังก์ชัน push, pop, front, size, empty
พร้อมระบุเหตุผลสั้น ๆ

ทำงานถูกต้อง 因地ที่ข้อมูลนี้เป็น Circular queue



push

- 4.2. ไม่ว่า code ข้างบนจะทำงานถูกต้องหรือไม่ก็ตาม ให้ตอบว่า ensureCapacity ข้างบนนี้ทำงาน “เร็ว” หรือ “ช้า” กว่า ensureCapacity แบบเก่า พร้อมระบุเหตุผลสั้น ๆ

ต้องการเวลาสั้นๆ แต่ต้องใช้เวลาในการคำนวณ mCap

5. (20 คะแนน) ในข้อย่ออยู่ต่อไปนี้ให้ตอบโดยเลือกคำตอบที่ถูกต้องที่สุด แต่ละข้อย่ออยู่มีคะแนน 1 คะแนน หากไม่ตอบในข้อใด จะได้คะแนน 0 แต่ถ้าหากตอบผิดในข้อใด จะได้คะแนน -0.5 ต่อข้อย่ออยู่ อย่างไรก็ตาม ถึงแม้จะตอบผิดจนได้คะแนนรวมติดลบ จะถือว่า ข้อนี้ได้คะแนนเป็น 0 **ให้เขียนคำตอบลงในตารางข้างล่างนี้เท่านั้น**

ข้อ 1	ข้อ 2	ข้อ 3	ข้อ 4	ข้อ 5	ข้อ 6	ข้อ 7	ข้อ 8	ข้อ 9	ข้อ 10
ข้อ 11	ข้อ 12	ข้อ 13	ข้อ 14	ข้อ 15	ข้อ 16	ข้อ 17	ข้อ 18	ข้อ 19	ข้อ 20

- 5.1. ข้อใดต่อไปนี้เกี่ยวกับฟังก์ชัน std::vector::resize() ใน C++ ที่เป็นจริง?

- ก. มันสามารถเพิ่มขนาดของ vector ได้เท่านั้น เมื่อลดขนาดของ vector อาจจะไม่มีการลดขนาดหน่วยความจำที่ใช้
- ค. มันจะจดหน่วยความจำใหม่สำหรับ vector ทุกครั้งที่เรียก
- ง. มันจะกำหนดค่าตั้งต้น (default value) ให้กับข้อมูลใหม่ได้เท่านั้น

- 5.2. ให้มี `std::vector<int> v={2, 4, 6, 8, 10};` คำสั่งต่อไปนี้ข้อใดจะลบค่า 6 ออกจาก v?

- ก. `v.erase(2);`
- ข. ~~v.erase(vec.begin() + 2);~~
- ค. `v.pop_back(3);`
- ง. `v.delete(2);`

- 5.3. พิจารณาโค้ด C++ ต่อไปนี้:

```

std::vector<int> d = {10, 20, 30, 40, 50};
auto it = d.insert(d.begin() + 3, 35);

```

หลังจากการดำเนินการของโค้ดนี้, ค่าของ *it คืออะไร?

- ก. 30
- ข. ~~35~~
- ค. 40
- ง. พฤติกรรมที่ไม่ได้กำหนดไว้

- 5.4. สำหรับ `std::vector<int> vec` การดำเนินการไหนต่อไปนี้ใช้เวลาในการทำงานที่เป็นไปได้ว่าจะแปรผันตามจำนวนข้อมูลใน vec

- ก. ~~vec.push_back(42);~~
- ข. `vec[3]`
- ค. `vec.front();`
- ง. `vec.capacity();`

- 5.5. ให้ q1 และ q2 เป็น `std::queue<int>` ถ้า q1 มีข้อมูล [1, 2, 3, 4] และ q2 มีข้อมูล [3, 4] (ให้หัวคิวอยู่ด้านซ้าย) การดำเนินการใดทำให้ q1 และ q2 มีข้อมูลเหมือนกัน?

- ก. `q1.push(3); q1.push(4);`
- ข. `q2.push(q1.front()); q2.push(q1.back());`
- ค. ~~q1.pop(); q1.pop();~~
- ง. `q1.swap(q2);`

- 5.6. ให้ q1 และ q2 เป็น `std::queue<int>` คุณต้องการเพิ่มเติมทุกข้อมูลใน q2 ไปยังท้ายสุดของ q1 โดยที่ q2 ยังต้องมีข้อมูลเหมือนเดิม วิธีใดถูกต้อง?

- ก. ใช้ `q1.merge(q2);`
- ข. ~~วนลูป q2.size() รอบ แต่ละรอบ pop จาก q2 ไปใส่ทั้ง q1 และ q2~~
- ค. ใช้ `q1.insert(q1.end(), q2.begin(), q2.end());`
- ง. ใช้ `q1.push(q2.back());`

- 5.7. พิจารณา `std::stack<int> s1 และ s2`, โดยที่ `r1` มีค่าเริ่มต้น `[3, 2, 1]` (1 เป็นสมาชิกที่อยู่บนสุด) และ `s2` มีค่าเริ่มต้น `[6, 5, 4]` (4 เป็นสมาชิกที่อยู่บนสุด) ถ้า เป้าหมายคือทำให้ `s2` มีสมาชิกทั้งหมดในลำดับ `[6, 5, 4, 3, 2, 1]`, วิธีใดถือว่าเหมาะสม?
- นำสมาชิกจาก `s2` ออกแล้วเพิ่มลงใน `s1` จนกว่า `s2` จะว่าง
 - ใช้ `std::queue` เพื่อเก็บสมาชิกจาก `s1` ชั่วคราว เพิ่มสมาชิกจาก `s2` ลงใน `s1`, แล้วจึงเพิ่มสมาชิก จาก `std::queue` ชั่วคราวกลับไปยัง `s1`
 - `swap(r1,s2)` จานนั้นนำสมาชิกจาก `s2` ออกแล้ว เพิ่มลงใน `s1` จนกว่า `s2` จะว่าง
 - ~~สร้าง `std::stack tmp` ขึ้นมา นำสมาชิกจาก `s1` ออกแล้วเพิ่มลงใน `tmp` จนกว่า `r1` จะว่าง จานนั้น นำสมาชิกออกจาก `tmp` แล้วเพิ่มลงใน `s2` จนกว่า `tmp` จะว่าง~~
- 5.8. พิจารณา `std::stack<int>` ส่องตัว `s1` และ `s2` ที่มีขนาดเท่ากันและมีสมาชิกที่เหมือนกันในลำดับเดียวกัน คำสั่งใดต่อไปนี้ถูกต้องเกี่ยวกับการดำเนินการ `s1 == s2`?
- ~~มันจะคืนค่า `true` เนื่องจากทั้งสอง `stack` มี สมาชิกที่เหมือนกันในลำดับเดียวกัน~~
 - มันจะทำให้เกิดข้อผิดพลาดขณะคอมไพล์, เนื่องจากตัวดำเนินการ `==` ไม่ได้ถูกกำหนดไว้สำหรับ `std::stack`
 - มันจะคืนค่า `false` เนื่องจากแต่ละ `stack` เป็น `instance` ที่แตกต่างกัน
 - มันจะรวมทั้งสอง `stack` เป็นหนึ่งเดียวโดยมี สมาชิกจากทั้งสอง `stack`
- 5.9. คุณมี `std::map<int, int>` ซึ่งว่า `numMap`, และคุณ ต้องการทราบว่าคีย์ 5 มีอยู่ใน `numMap` หรือไม่? คุณ จะใช้ฟังก์ชัน STL ตัวไหน?
- `numMap.find(5)`
 - `numMap.count(5)`
 - `numMap.exists(5)`
 - ~~ทั้ง ก และ ข สามารถใช้ได้~~
- 5.10. คุณมี `std::map<std::string, int>` ซึ่งว่า `wordFreq` ซึ่งประกอบด้วยความถี่ของคำในเอกสาร คุณต้องการ เรียงลำดับคำเหล่านี้ตามความถี่และเก็บคำที่มีความถี่ สูงที่สุด 5 คำลงใน `std::vector<std::string>` วิธีใด ต่อไปนี้ที่เหมาะสมสำหรับการดำเนินการนี้?
- ใช้ `std::sort` กับ `wordFreq` โดยตรงแล้วคัดลอก คีย์ 5 คีย์สูงสุดไปยังเวกเตอร์
 - ~~แปลงเป็นเวกเตอร์ของคู่คำและความถี่ แล้วใช้ `sort` พร้อมกับ `comparator` ที่เขียนเองเพื่อเรียง ตามความถี่ แล้วดึงคำ 5 คำสูงสุดออกมายังเวกเตอร์~~
 - ใช้ `std::lower_bound` กับ `wordFreq` เพื่อค้นหา ความถี่ที่สูงที่สุดลำดับที่ 5 แล้ววนลูปเพื่อรับรวม คำที่มีความถี่เกินเกณฑ์นี้

5. ค้นหาคำที่มีความถี่สูงที่สุดโดยใช้ `std::max_element` แล้วเพิ่มลงในเวกเตอร์ และ จานนั้นลบมันออกจาก `map` นี้แล้วทำไป 5 ครั้ง
- 5.11. คุณสมบัติใดของข้อมูลใน `std::set` ถูกต้องที่สุด?
- ข้อมูลถูกเรียงลำดับจากมากไปน้อย ~~อยู่~~
 - ข้อมูลถูกเก็บตามลำดับตามที่ใส่ค่าเข้ามา ~~มา~~
 - อนุญาตให้มีค่าที่ซ้ำกันในเซ็ต ~~อยู่~~
 - ~~ข้อมูลถูกเรียงลำดับตามเกณฑ์ที่ระบุไว้ในการ ประมวลผล (default เป็นจากน้อยไปมาก)~~
- 5.12. คำสั่งใดใช้ในการหาจำนวนเต็มที่น้อยที่สุดที่มากกว่า 15 ที่มีอยู่ใน `std::set<int> s`?
- `s.find(15).second;`
 - `++s.find(15);`
 - ~~`s.upper_bound(15);`~~
 - `s.lower_bound(15);`
- 5.13. ให้ `x` เป็นโครงสร้างข้อมูลในประเภทต่าง ๆ ต่อไปนี้ หาก `x` มีค่าที่เก็บเป็น 1 ถึง 1,000,000 ตามลำดับ โครงสร้างข้อมูลใดที่ทำให้การลบค่า 9 ออกจาก โครงสร้างข้อมูลดังกล่าวใช้เวลาน้อยที่สุด
- `vector<int>`
 - `set<int>`
 - ~~`queue<int>`~~
 - `priority_queue<int>`
- 5.14. สมมติว่าคุณมี `std::priority_queue<int> pq` และ `std::map<int, int> freqMap` ซึ่ง `freqMap` จะบันทึก ความถี่ของแต่ละจำนวนเต็มใน `priority queue` ถ้า คุณต้องการเอาสมาชิกที่มีความสำคัญสูงสุดออกจาก `priority queue` และอัปเดต `freqMap` ให้ถูกต้อง คำสั่งใดต่อไปนี้ที่ควรใช้?
- ~~`freqMap[pq.top()]-; pq.pop();`~~
 - `pq.pop(); freqMap[pq.top()]-;`
 - ~~`freqMap[pq.front()]-; pq.pop();`~~
 - `pq.pop(); freqMap[pq.front()]-;`
- 5.15. หากคุณมี `std::map<int, std::stack<char>> m` ซึ่ง มีคีย์เป็นระดับความสำคัญและค่าเป็น `stacks` ของ ตัวอักษรที่มีระดับความสำคัญนั้น หากคุณต้องการดึง และลบตัวอักษรที่มีระดับความสำคัญสูงที่สุดและอยู่ บนสุดของ `stack` ของตัวมัน วิธีใดต่อไปนี้ที่เหมาะสม ที่สุด?
- `char c = m.rbegin()->second.top();
m.begin()->second.pop();`
 - `char c = m.begin()->second.top();
m.begin()->second.pop();`
 - `char c = (--m.end())->second.top();
(--m.end())->second.pop();`
 - ~~for (auto it = m.rbegin(); it != m.rend(); ++it) {
if (!it->second.empty()) {
c = it->second.top();
it->second.pop();
break;
}~~

5.16. ให้ `v` เป็น `vector` ที่ `v.size()` คือ 1,000,000 ตัว และ `it` เป็น iterator ของ `v` ที่ถูกต้อง คำสั่งใดต่อไปนี้มีโอกาสใช้เวลามากที่สุด

- ก. `v.insert(it,x);` ผิด
- ข. `v.erase(it);`
- ค. `find(it,v.end(),x);`
- ง. ทุกข้อใช้เวลาเท่ากัน

5.17. ให้ `v` เป็น `vector` ที่ `v.size()` คือ 1,000,000 ตัว และ `it` เป็น iterator ของ `v` ที่ถูกต้อง คำสั่งใดต่อไปนี้มีโอกาสใช้เวลาอย่างมากที่สุด

- ก. `v.insert(it,x);` ผิด
- ข. `v.erase(it);` ผิด
- ค. `find(it,v.end(),x);` ถูก
- ด. ทุกข้อใช้เวลาเท่ากัน

5.18. ส่วนของโปรแกรมต่อไปนี้ ทำให้เกิดการเรียกใช้คำสั่ง `delete` กี่ครั้ง ก่อนจบการทำงานบรรทัดที่ 5

```
1: vector<int> v(4); int i;
2: while (v.size() <= 256) {
3:     v.push_back(0);
4: } // บรรทัดที่ 4 v.size() = 257
5: i = 10;
```

- ก. 5
- ข. 6
- ค. 7
- ง. 8

5.19. ข้อใดต่อไปนี้ `it` เป็น iterator ที่อ้างถึงตำแหน่งที่ใช้งานไม่ได้ (`invalid iterator`)

vector<int> v(16);

it = v.begin();

v.resize(17);

ค. map<int,int> m;

int n = 10;

while (n--) m[n] = -n;

auto it = m.find(5);

m.erase(4);m.erase(6);

ด. set<int> s;

s.insert(1); s.insert(2); s.insert(3);

auto it = s.find(2);

s.insert(4);

ง. vector<int> v(100);

it = v.begin() + 10;

v.resize(40);

5.20. พิจารณาส่วนของโปรแกรมต่อไปนี้ เรามีตัวแปร `pr` ประเภท `bool` อยู่ก็ตัวในหน่วยความจำเมื่อทำงาน จนถึงบรรทัดสุดท้าย

```
CP::stack<CP::vector<bool>> s;
CP::vector<bool> v(10);
while (v.size() > 0) {
    s.push_back(v); v.erase(v.end() - 1); }
```

- | | |
|---|----|
| ก. 55 | 10 |
| <input checked="" type="checkbox"/> ข. 65 | 9 |
| ค. 100 | 8 |
| ง. 110 | 7 |
| | 6 |
| | 5 |
| | 4 |
| | 3 |
| | 2 |
| | 1 |

ในข้อต่อๆ ต่อไปนี้เป็นการเขียน code คะแนนที่ได้จะประพันตามความถูกต้อง และ ประสิทธิภาพ ในการทำงาน โดยเฉพาะการเลือก data structure ในการใช้งานที่ถูกต้อง

6. (10 คะแนน) เกมประเภท rhythm game (เช่นเกมเดิน เกมจับจังหวะ) ผู้เล่นจะต้องกดปุ่มต่างๆ ให้ตรงตามจังหวะที่เกมกำหนด สมมติให้ในเกมหนึ่งมีปุ่มที่ผู้เล่นต้องกดได้แก่ `A, W, S, D` บันคีย์บอร์ด เรарат้องการคิดคะแนนของผู้เล่นจากการกดปุ่มเหล่านี้ กำหนดให้มี `vector<string>` ที่ระบุการกดปุ่มของผู้เล่นไว้ โดยแต่ละช่องของเวคเตอร์ คือ ปุ่มที่ผู้เล่นกดในหนึ่งช่วงเวลาในเกม โดยจะมีสตริงอักขระแบบ คือ “ ” ซึ่งแทนการไม่กด และ “_” ซึ่งแทนการกดปุ่มก่อนหน้าค้างไว้ ตัวอย่างเวคเตอร์ที่เอาไว้คิดคะแนนเช่น `{A,D,S,W,_W,W,S,S,_D,_S,A,_W,W,S,A}` (เพื่อให้อ่านง่าย จึงลงเครื่องหมาย “ ” ไป) โปรแกรมจะคิดคะแนนจาก `vector` ต้นแบบของเกม เพื่อดูว่าปุ่มที่ผู้เล่นกดตรงกับต้นแบบหรือไม่ การคิดคะแนนจะดูว่า
- ในช่องตำแหน่งเดียวกันของทั้งสอง `vector` ตรงกันก็อัน
 - มีข้อยกเว้น คือ เมื่อ `vector` ต้นแบบมีการกดค้าง ให้นับปุ่มที่กดรวมกับค้างนั้นเป็นหนึ่งคะแนน (นับตั้งแต่ตัวอักษร รวมไปถึง การกดค้างครั้งสุดท้าย) โดยจะได้คะแนนนี้ก็ต่อเมื่อกดถูกปุ่มและกดค้างครบเท่ากับจำนวนการกดค้างของต้นแบบเท่านั้น
 - ตัวอย่างเช่น ตัวต้นแบบเป็น `{A,A,_,_,A}` มันจะถูกเอาไปคิดเป็นสามคะแนนเท่านั้น คือ `A, A, ___, และ A`
 - ตัวอย่างการคิดคะแนนเป็นดังนี้

ต้นแบบ	ปุ่มที่ผู้เล่นกด	คะแนนที่ได้	คำอธิบาย
{A,A,_,_,A}	{A,A,_,_,A}	3	จับได้ตรงกับหมวด
{A,A,_,_,A}	{A,A,A,_,A}	2	A, _, _ จากต้นฉบับไม่ถูกจับคู่ แต่ A ที่อยู่ช้ายสุดกับขวาสุดจับคู่ได้
{A,A,_,_,A}	{A,A,_,S,A}	2	A, _, _ จากต้นฉบับไม่ถูกจับคู่ แต่ A ที่อยู่ช้ายสุดกับขวาสุดจับคู่ได้
D, ___, A, ___, W, ___, ___, S, ___,	D, ___, ___, W, ___, ___, S, ___,	2	D, ___, S, ___, เท่านั้นที่ได้คะแนน (ใช้เวคเตอร์ ต้นฉบับเป็นตัวตั้งเวลาคิดคะแนนเสมอ)

จงเขียนฟังก์ชัน calculateScore เพื่อคำนวณคะแนนตามวิธีข้างต้น รับประกันว่าต้นแบบ (key) และ ปุ่มที่ผู้เล่นกด (user) มีความยาวเท่ากัน

```
int calculateScore(vector<string> key, vector<string> user) { //key is the vector ต้นแบบ
    int score = 0;
    int n = key.size();
    for(int i=0; i!=n; i++)
    {
        string k = key[i];
        string p = user[i];
        while(i+1!=n && key[i+1] == "_")
        {
            k += key[i+1];
            p += user[i+1];
            i++;
        }
        score += (p==k);
    }
    return score;
}
```

7. (10 คะแนน) เกม SumaType เป็นเกมฝึกใช้แป้นพิมพ์ โดยผู้เล่นจะต้องสู้กับศัตรูต่าง ๆ ด้วยการกดปุ่มให้ตรงตามลำดับ ศัตรูแต่ละตัวจะมีค่าพลังเป็น $\text{vector<char>} \text{ power}$ เช่น $\{\text{l}, \text{a}, \text{m}, \text{S}, \text{a}, \text{m}\}$ (ขอ秱เครื่องหมาย ‘ เพื่อให้อ่านง่าย) และผู้เล่นจะต้องสู้กับศัตรูโดยการพิมพ์สายอักษรความยาวอย่างน้อย 2 ตัวอักษรที่ตรงกับ “ส่วนย่อ” ค่าพลังของศัตรู โดยกำหนดให้ส่วนย่อของค่าพลังคือสายอักษรที่ประกอบด้วยตัวอักษร ที่ตำแหน่ง a ถึงตำแหน่ง b ของ power สำหรับค่า a และ b ใด ๆ ที่ $0 \leq a \leq b < \text{power.size()}$ เมื่อผู้เล่นพิมพ์สายอักษรดังกล่าวถูกต้องแล้ว ค่าพลังของศัตรูจะลดลงไปโดยเกิดจากการลบส่วนย่อดังกล่าวทั้งหมดที่ปรากฏอยู่ใน power ออกไป

ตัวอย่างต่อไปนี้แสดงถึงสิ่งที่เกิดขึ้นเมื่อผู้เล่นพิมพ์สายอักษรมา 1 ตัว

power	คำที่ผู้เล่นพิมพ์	power หลังจากพิจารณาคำที่พิมพ์	คำอธิบาย
$\{\text{l}, \text{a}, \text{m}, \text{S}, \text{a}, \text{m}\}$	am	$\{\text{l}, \text{S}\}$	เอา a, m ที่เจอทั้งหมดออกจากเวคเตอร์
$\{\text{l}, \text{a}, \text{m}, \text{S}, \text{a}, \text{m}\}$	sa	$\{\text{l}, \text{a}, \text{m}, \text{S}, \text{a}, \text{m}\}$	ไม่ตรงกัน (อักษรตัวใหญ่ตัวเล็กถือว่าไม่เหมือนกัน)
$\{\text{l}, \text{a}, \text{m}, \text{S}, \text{a}, \text{m}\}$	lam	$\{\text{S}, \text{a}, \text{m}\}$	ลบ $\text{l}, \text{a}, \text{m}$ ออกจากเวคเตอร์
$\{\text{l}, \text{a}, \text{m}, \text{S}, \text{a}, \text{m}\}$	Sam	$\{\text{l}, \text{a}, \text{m}\}$	ลบ $\text{S}, \text{a}, \text{m}$ ออกจากเวคเตอร์

{a, a, a, M, M, M}	aM	{a, a, M, M}	คล aM ออกไปครึ่งเดียว (ถึงแม้คลบแล้วจะมี aM เหลืออยู่ แต่ aM ที่เหลืออยู่นั้นไม่ได้เป็น aM ที่อยู่ติดกันในตอนแรก)
--------------------	----	--------------	---

จงเขียนฟังก์ชันที่คำนวณค่าพลังของศัตรูที่เหลืออยู่เมื่อผู้เล่นพิมพ์สายอักขระเข้ามา

```
void attack(vector<char> &power, string input){
```

```
    int n = power.size();
    vector<char> tmp(n);
    string now = "";
    for(int i=0 ; i!=n ; it++)
    {
        tmp.push_back(power[i]);
        if (input[now.size()] != power[i]){
            now = "";
            continue;
        }
        now += power[i];
        if(now == input){
            for(int i=0 ; i!=input.size() ; it++) tmp.pop_back();
            now = "";
        }
    }
    power = tmp;
}
```

8. (10 คะแนน) สมชายเปิดบริษัทให้บริการติวหนังสือ สมชายมีพนักงาน n คน การให้บริการแต่ละครั้งนั้น สมชายจะส่งพนักงานออกไป 1 คนเท่านั้น โดยจะไปติวหนังสือเป็นจำนวนวันเท่ากับที่ลูกค้าต้องการพอดี เมื่อติวเสร็จ พนักงานคนดังกล่าวสามารถไปรับงานตัวได้ในวันถัดไปทันที มีงานเข้ามาเป็นจำนวน m งาน และเพื่อความเป็นระเบียบ สมชายจะทำงานแต่ละงานตามลำดับ กล่าวคือ ถ้ามีพนักงานว่างเมื่อไร สมชายจะส่งพนักงานไปทำงานที่ยังไม่มีคนไปทำ ที่มาก่อนที่สุด

งานแต่ละงานประกอบด้วยข้อมูล 3 ส่วนคือ “ชื่อลูกค้า” “ชื่องาน” และ “จำนวนวัน” ลูกค้าแต่ละคนอาจจะมีงานมาหลายงาน ก็ได้ แต่รับประทานว่า “ชื่องาน” จะไม่ซ้ำกันเลย งานแต่ละชิ้น จะมาในรูปแบบ $\text{pair}<\text{string}, \text{pair}<\text{string}, \text{int}>>$ โดย string ตัวแรกคือ ชื่อลูกค้า และ string ตัวที่สองคือชื่องาน ส่วน int นั้นระบุจำนวนวัน

เราต้องการทราบว่า งานแต่ละงานทำเสร็จตามลำดับใด และ ลูกค้าแต่ละคนได้รับบริการรวมทั้งหมดเป็นจำนวนวันกี่วัน เพื่อความสะดวก กำหนดให้วันแรกที่เริ่มเปิดให้บริการคือวันที่ 0 และ วันที่ i ในระบบสามารถบุได้ด้วย int ทั้งหมด

จงเขียนฟังก์ชัน $\text{processTask}(n, \&\text{task}, \&\text{finish_task}, \&\text{client_total})$ เพื่อคำนวณคำตอบดังกล่าว โดยกำหนดให้

- tasks เป็น $\text{vector}<\text{pair}<\text{string}, \text{pair}<\text{string}, \text{int}>>$ ที่ระบุงานที่เข้ามาที่บริษัทตามลำดับ ให้ $\text{task}[0]$ คืองานแรกสุด

- `finish_task` เป็น `vector<string>` โดย `finish_task` ต้องเก็บชื่องานที่เสร็จตามลำดับ (นับจากวันที่งานเสร็จ และให้งานที่เสร็จแรกอยู่ที่ `finish_task[0]` หากมีงานมากกว่าหนึ่งงานเสร็จพร้อมกันในวันเดียวกัน งานที่เสร็จพร้อมกันเหล่านั้นจะเรียงตามอัลฟิเด้)
- `client_total` เป็น `map<string,int>` โดย `client_total[name]` จะต้องเป็นจำนวนวันรวมที่ลูกค้าชื่อ `name` มาใช้บริการ
- ให้ `finish_task` และ `client_total` เมื่อเรียกใช้ฟังก์ชันจะไม่มีข้อมูลใด ๆ อยู่ภายในเลย ฟังก์ชันนี้จะต้องเติมข้อมูลเข้าไปให้ถูกต้อง

```

void processTasks(int n,
    const std::vector<std::pair<std::string, std::pair<std::string, int>>>& tasks, //tasks.size() คือ m
    std::vector<std::string>& finish_task,
    std::map<std::string, int>& client_total) {

#define PIS pair<int,string>
priority_queue<PIS ,vector<PIS> ,greater<PIS>> pq;
for(int i=0; i<=n; i++){
    auto task = tasks[i];
    int time = task.second.second;
    string taskname = task.second.first;
    string name = task.first;
    client_total[name] += time;
    pq.push(make_pair(time,taskname));
}
int idx = n;
while (!pq.empty()){
    auto finish = pq.top(); pq.pop();
    int doneTime = finish.first;
    string taskname = finish.second;
    finish_task.push_back(taskname);
    if(idx!=m){
        auto task = tasks[idx+1];
        int time = task.second.second;
        string taskname = task.second.first;
        string name = task.first;
        client_total[name] += time;
        pq.push(make_pair(time,taskname));
    }
}
}
}

```

9. (10 คะแนน) Social network แห่งหนึ่งมีผู้ใช้ n คน โดยผู้ใช้แต่ละคนสามารถระบุได้ด้วยหมายเลขตั้งแต่ 1 ถึง n ในแต่ละวันนั้น เราจะวัดจำนวนครั้งที่แต่ละคน “สร้าง content” ในระบบ และ “กดถูกใจ” ให้กับ content ที่สร้างโดยคนอื่น ฝ่ายพัฒนาระบบของระบบมีความสามารถอยู่สองความสามารถ คือ 1) ต้องการหาว่า มีใครบ้างมีการ “กดถูกใจ” รวมตั้งแต่ตอนเริ่มใช้งานเป็นจำนวนครั้งตั้งแต่ \min_like ถึง \max_like รวมทั้วทั้ย และ 2) ต้องการหาว่า มีใครบ้างมีการ “สร้าง content” รวมตั้งแต่ตอนเริ่มใช้งานเป็น จำนวนมากที่สุดที่น้อยกว่า threshold

จะเขียนคลาส `SocialNetwork` ซึ่งต้องมีฟังก์ชันต่อไปนี้เป็นอย่างน้อย (นิสิตสามารถเพิ่มเติมฟังก์ชันอื่น ๆ ได้)

- `void init(int n)` ฟังก์ชันนี้จะถูกเรียกเป็นฟังก์ชันแรกและเรียกเพียงครั้งเดียวเพื่อบอกว่าระบบมีผู้ใช้งาน n คน
- `void add_content(int user_id, int content)` ซึ่งเป็นฟังก์ชันที่ระบบจะเรียกใช้เพื่อบอกว่าผู้ใช้หมายเลข `user_id` มีการสร้าง content เพิ่มขึ้นเป็นจำนวน `content` ขึ้น รับประทานว่าค่า `content` เป็นจำนวนเต็มบวก และ $1 \leq user_id \leq n$
- `void add_like(int user_id, int like)` ซึ่งเป็นฟังก์ชันที่ระบบจะเรียกใช้เพื่อบอกว่าผู้ใช้หมายเลข `user_id` มีการกดถูกใจเพิ่มขึ้น เป็นจำนวน `like` ครั้ง รับประทานว่าค่า `like` เป็นจำนวนเต็มบวก และ $1 \leq user_id \leq n$
- `vector<pair<int,int>> like_in_range(int min_like, int max_like)` เป็นฟังก์ชันที่เรียกเพื่อตอบคำถาม 1) โดยจะต้องคืนเป็น `vector` ของ `pair` ของ {หมายเลขผู้ใช้, จำนวนการกดถูกใจ} เรียงตามจำนวนการกดถูกใจน้อยไปมาก (ถ้าจำนวนถูกใจเท่ากัน จะเรียงตามลำดับได้ก็ได้) รับประทานว่า $\min_like \leq \max_like$
- `pair<int,vector<int>> content_below(int threshold)` เป็นฟังก์ชันที่เรียกเพื่อตอบคำถาม 2) โดยจะต้องคืนเป็น `pair` ของ จำนวน content และ `vector` ของ หมายเลขของผู้ใช้ที่สร้าง content ตรงกับเงื่อนไข (เรียงตามลำดับได้ก็ได้)

จงตอบคำถามต่อไปนี้

1.1. จงระบุชื่อและประเภทของ Data Member ที่ใช้ในคลาสนี้ พร้อมทั้งระบุวัตถุประสงค์ของ Data Member ดังกล่าว

```
set<pair<int, int>> liked, content; //กับมีส่วนของปัจจัย , int like = จำนวน like/content, int size = ขนาด content
map<int, int> l, c; //กับมีส่วนของ key = จำนวน, value = จำนวน like/content
```

1.2. จงเขียนคลาสดังกล่าว (ให้ถือว่าสามารถใช้ Data Structure และฟังก์ชันต่าง ๆ ของ C++ ได้โดยไม่จำกัด)

```
#define PII pair<int,int>
Class SocialNetwork{
private:
    int n;
    set<PII, greater<PII>> content;
    set<PII> liked;
    map<int,int> l,c;
public:
    void init(int a): n(a) {}
    void add_content(int user_id, int content){
        PII tmp = make_pair(c[user_id],user_id); //ตั้งค่า
        content.erase(tmp); //ลบออก
        c[user_id] += content; //บันทึก
        tmp.first = l[user_id];
        content.insert(tmp); //ตั้งค่า
    }
    void add_like (int user_id, int like){
        PII tmp = make_pair(l[user_id],user_id); //ตั้งค่า
        liked.erase(tmp); //ลบออก
        l[user_id] += like; //บันทึก
        tmp.first = l[user_id];
        liked.insert(tmp); //ตั้งค่า
    }
    vector<PII> like_in_range(int min_like,int max_like){
        auto start = liked.lower_bound (make_pair(min_like, 0));
        auto stop = liked.upper_bound(make_pair(max_like, nt));
        vector<PII> ans;
        for(auto it = start ; it != stop ; it++){
            int like = it->first;
            int id = it->second;
            ans.push_back (make_pair(id,like));
        }
        return ans;
    }
    pair<int,vector<int>> content_below (int threshold) {
        auto it = content.upper_bound (make_pair(threshold, 0));
        pair<int,vector<int>> ans;
        ans.first = it->first;
        ans.second.push_back(it->second);
        while((it+1)->first == ans.first) ans.second.push_back((it+1)->second);
        return ans;
    }
};
```

Reference

Common (All classes support these two capacity functions)

Capacity	<code>size_t size(); // return the number of items in the structure</code> <code>bool empty(); // return true only when size() == 0</code>
----------	---

Container Class (All classes in this category support these two iterator functions.)

Iterator	<code>iterator begin(); // an iterator referring to the first element</code> <code>iterator end(); // an iterator referring to the past-the-end element</code>
----------	---

vector<T> และ list<T>

Element Access สำหรับ vector	<code>T& operator[] (size_t n);</code> <code>T& at(int dx);</code>
Modifier ที่ใช้ได้ทั้ง list และ vector	<code>void push_back(const T& val);</code> <code>void pop_back();</code> <code>iterator insert(iterator position, const T& val);</code> <u><code>iterator insert(iterator position, InputIterator first, InputIterator last);</code></u> <code>iterator erase(iterator position);</code> <u><code>iterator erase(iterator first, iterator last);</code></u> <code>void clear();</code> <code>void resize(size_t n);</code>
Modifier ที่ใช้ได้เฉพาะ list	<code>void push_front(const T& val);</code> <code>void pop_front();</code> <code>void remove(const T& val);</code>

set<T>

Operation	<code>iterator find (const T& val);</code> <code>size_t count (const T& val);</code>
Modifier	<code>pair<iterator,bool> insert (const T& val);</code> <code>void insert (InputIterator first, InputIterator last);</code> <code>iterator erase (iterator position);</code> <u><code>iterator erase (iterator first, iterator last);</code></u> <code>size_t erase (const T& val);</code>

map<KeyT, MappedT>

Element Access	<code>MappedT& operator[] (const KeyT& k);</code>
Operation	<code>iterator find (const KeyT& k);</code> <code>size_t count (const KeyT& k);</code>
Modifier	<code>pair<iterator,bool> insert (const pair<KeyT,MappedT>& val);</code> <code>void insert (InputIterator first, InputIterator last);</code> <code>iterator erase (iterator position);</code> <u><code>iterator erase (iterator first, iterator last);</code></u> <code>size_t erase (const KeyT& k);</code>

Container Adapter

These three data structures support the same data modifiers but each has different strategy. These data structures do not support iterator.

Modifier	<code>void push (const T& val); // add the element</code> <code>void pop(); // remove the element</code>
----------	---

	queue<T>	<code>stack<T> and priority_queue<T, ContainerT = vector<T>, CompareT = less<T> ></code>
Element Access	<code>T front();</code> <code>T back();</code>	<code>T top();</code>

Useful functions

```

iterator find(iterator first, iterator last, const T& val);
void sort(iterator first, iterator last, Compare comp);
void lower_bound(iterator first, iterator last, const T& val);
void upper_bound(iterator first, iterator last, const T& val);
pair<T1,T2> make_pair (T1 x, T2 y);

```