

Lab3 Solutions

Love Letter

สังเกตว่าถ้ามีสระ ตัวอักษรที่เข้ารหัสแล้วจะมีรูปแบบเดียวกันเสมอ (apa, epe, ...) ดังนั้นสามารถวนลูปเพื่อ print ตัวอักษรตามปกติได้เลย โดยเพิ่มเงื่อนไขว่าถ้าพบสระให้ข้ามการ print ไป 2 ตัวอักษร

Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    string str, vowel = "aeiou";
    getline(cin, str);

    for(int i=0; i<str.length(); i++)
    {
        cout << str[i];
        for(int j=0; j<vowel.length(); j++) if(str[i]==vowel[j]) i+=2;
    }
}
```

time complexity : $O(n)$

Exam

ประกาศ struct *student* เพื่อเก็บข้อมูลนักเรียนแต่ละคน ทำให้เข้าถึงข้อมูลและคำนวณคะแนนรวม
หรือค่าเฉลี่ยได้สะดวกขึ้น

Example Code

```
#include <bits/stdc++.h>
using namespace std;

typedef struct student{
    int num;
    int point[3];
    float sum;
} Student;

int main()
{
    float average[3]={0}, sum=0;
    int n;
    cin >> n;

    Student students[n];
    for(int i=0;i<n;i++)
    {
        students[i].num = i+1;
        students[i].sum = 0;
        for(int j=0;j<3;j++)
        {
            cin >> students[i].point[j];
            students[i].sum += students[i].point[j];
            average[j] += students[i].point[j];
        }
        sum += students[i].sum;
    }
}
```

```
for(int i=0;i<n;i++)
{
    bool check = true;
    int countpass=0;
    for(int j=0;j<3;j++)
    {
        countpass += (students[i].point[j] >= average[j]/n);
        if(students[i].point[j]==0) check = false;
    }
    if(check && students[i].sum >= sum/n && countpass>=2)
        cout << students[i].num << " ";
}
}
```

time complexity : $O(n)$

Encrypt

ใช้เลขแถวแรกเป็นค่าเริ่มต้น เก็บไว้ใน array *temp* จากนั้นพิจารณาตัวเลขแต่ละตัวในแถวที่ 2 ถึงแถวที่ *n* สมมติให้ *i* แทนเลขแถว และ *j* แทนลำดับเลขใน *temp* ถ้าหากค่า *temp* ภายในตำแหน่งที่ *j* เท่ากับ *k* ให้เปลี่ยนเป็นเลขใน *arr* แถวที่ *i* ตำแหน่งที่ *k* แทน ($temp[j] = arr[i][temp[j]]$)

ปล.ในโค้ดใช้ $temp[j] - 1$ เพราะตำแหน่งใน array เป็นได้จาก 0 ถึง $n - 1$ ไม่ใช่ 1 ถึง n

Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int m,n;
    cin >> m >> n;
    int arr[m][n],temp[n];
    for(int i=0;i<m;i++) for(int j=0;j<n;j++) cin >> arr[i][j];

    for(int i=0;i<n;i++) temp[i] = arr[0][i]; // init temp
    for(int i=1;i<m;i++) for(int j=0;j<n;j++) temp[j] = arr[i][temp[j]-1]; //decrypt

    for(int i=0;i<n;i++) cout << temp[i] << " ";
}
```

time complexity : $O(n^2)$