資結期末專題

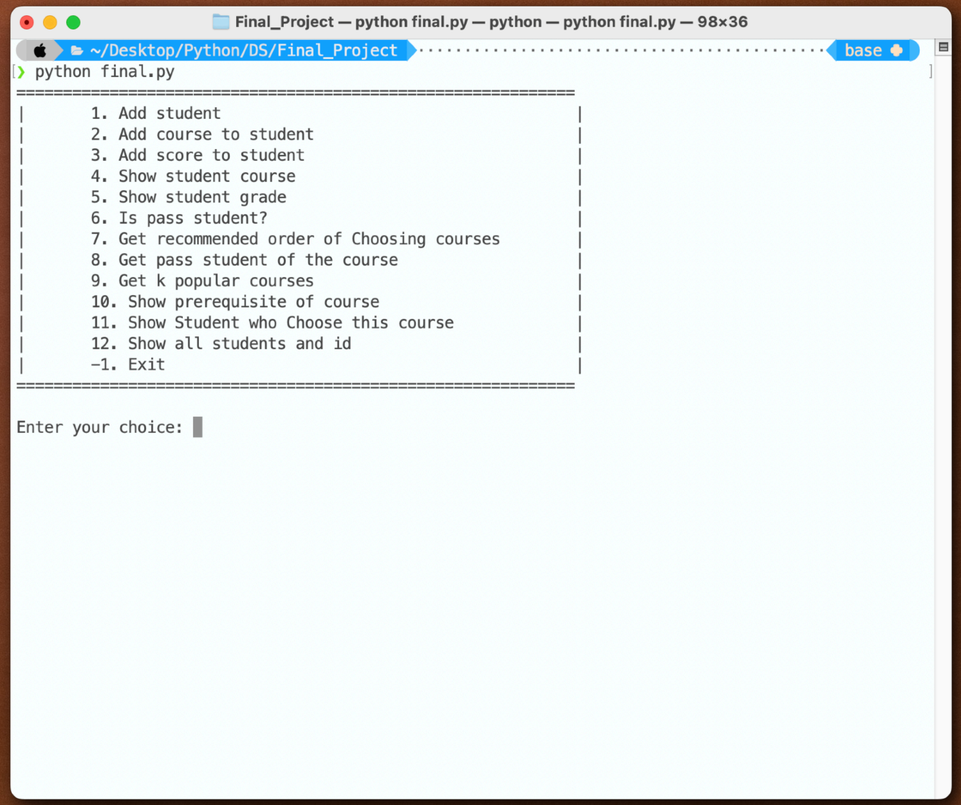
資管二 4111029007 史岱澂

學生課程、成績管理系統

操作選單

```
con = "y";
if __name__ == "__main__":
    choice = 0;
    while (con == "y" or con == "Y"):
        print("=" * 60);
        print("| \t1. Add student                                    |");
        print("| \t2. Add course to student                          |");
        print("| \t3. Add score to student                           |");
        print("| \t4. Show student course                            |");
        print("| \t5. Show student grade                             |");
        print("| \t6. Is pass student?                               |");
        print("| \t7. Get recommended order of Choosing courses      |");
        print("| \t8. Get pass student of the course                 |");
        print("| \t9. Get k popular courses                          |");
        print("| \t10. Show prerequisite of course                   |");
        print("| \t11. Show Student who Choose this course           |");
        print("| \t12. Show all students and id                      |");
        print("| \t-1. Exit                                          |");
        print("=" * 60);
        print();
        print("Enter your choice: ", end = "");
        try:
            choice = int(input());
        except ValueError:
            print("Please enter a number between 1 ~ 12 or -1.\n");
            continue;
        print();
```

Total: 12個選項（1 ~ 12), -1代表結束



執行畫面

學生課程、成績管理系統

1. Add Student

```
if (choice == 1):
    print("Enter student name: ", end = "");
    name = input();
    print("Enter student id: ", end = "");
    try: id = int(input());
    except ValueError:
        print("Please enter a number.\n");
        continue;
    add_student(id, name);
    print();
```

```
def add_student (id, name):
    global students_data, Len;
    if (id in all_id or name in all_name):
        print(f"Student is already exists.");
        return;
    all_id.append(id);
    all_name.append(name);
    students_data.append(Student(id, name));
    idx[name] = Len;
    idx[id] = Len;
    Len += 1;
```

```
class Student:
    def  __init__(self, id, name) -> None:
        self.id = id;
        self.name = name;
        self.course = DoublyLinkedList(); # what student choose
        self.grades = AVLTree(); # grades of student
```

輸入: 名字 & 學號

Len: 記錄該Student的資料是位於哪個index

All_name & All_id → 避免重複

所選課程 → DoublyLinkedList
分數 → AVLTree

學生課程、成績管理系統

2. Add Course to Student

```
elif (choice == 2):
    print("Enter student name: ", end = "");
    name = input();
    print("Enter course name: ", end = "");
    course = input();
    add_course_by_name(name, course);
    print();
```

```
def add_course_by_name (name, course):
    global students_data, idx;
    if (course not in course_data):
        print(f"Course {course} does not exist.");
        return;
    if (name in idx):
        if (students_data[idx[name]].add_course(course)):
            course_data[course].add_student(name);
            add_student_to_course(course_data, name, course);
    else: print(f"Student {name} does not exist.");
```

```
def add_student_to_course (course_dict, student, course):
    if (course in course_dict):
        # course_dict[course].add_student(student);
        if (popularity.search(course)):
            # print(f"{course} find");
            popularity.modify(course, course_dict[course].people);
        else: popularity.insert(course_dict[course].people, course);
    else: print(f"Course {course} does not exist.");
```

輸入：學生名字 & 課程名稱

每個課程有誰選課 → Queue

方便計算熱門課程 → MaxHeap

選課時會判斷有沒有被擋修，否則不能選

```
Enter your choice: 2

Enter student name: apple
Enter course name: Statistic 2
Student apple does not have the prerequisites of Statistic 2.
```

學生課程、成績管理系統

3. Add Score to Student

```
elif (choice == 3):
    print("Enter student name: ", end = "");
    name = input();
    print("Enter course name: ", end = "");
    course = input();
    print("Enter score: ", end = "");
    try: score = int(input());
    except ValueError:
        print("Please enter a number between 0 ~ 100.\n");
        continue;
    if (score > 100 or score < 0):
        print("Please enter a number between 0 ~ 100.");
        # continue;
    else: add_score_by_name(name, course, score);
    print();
```

```
def add_score_by_name (name, course, score):
    global students_data, idx;
    if (course not in course_data):
        print(f"Course {course} does not exist.");
        return;
    if (name in idx):
        if (not students_data[idx[name]].course.Search(course)):
            print(f"Student {name} does not choose {course}.");
            return;
        if (students_data[idx[name]].grades.search_course(students_data[idx[name]].grades.root, course)):
            print(f"Student {name} already has score of {course}.");
            return;
        students_data[idx[name]].add_grade(course, score);
        insert_subject_score(course, score, name);
    else: print(f"Student {name} does not exist.");
```

```
Enter your choice: 3

Enter student name: apple
Enter course name: Financial Accounting
Enter score: 90

continue? [y/n]
```

輸入：學生名字 & 課程名稱 & 分數
學生要新增該課程成績，該課程要新增學生成績

檢查是否有選該課程、該課程是否已經有成績

學生課程、成績管理系統

4. Show Student Course

```
elif (choice == 4):
    print("Enter student name: ", end = "");
    name = input();
    show_student_course_by_name(name);
    print();
```

```
Enter your choice: 4

Enter student name: apple
Student apple choose: Calculus,Linear Algebra,Financial Accounting

continue? [y/n]
```

```
def show_student_course_by_name (name):
    global students_data, idx;
    if (name in idx):
        print(f"Student {name} choose: ", end = "");
        students_data[idx[name]].course.Print();
    else: print(f"Student {name} does not exist.");
```

輸入：學生名字
印出Doubly Linked List內容

5. Show Student Grade

```
elif (choice == 5):
    print("Enter student name: ", end = "");
    name = input();
    show_student_grade_by_name(name);
    print();
```

```
Enter your choice: 5

Enter student name: apple
Student apple grades:
Financial Accounting: 90
Calculus: 100
Linear Algebra: 100


continue? [y/n]
```

```
def show_student_grade_by_name (name):
    global students_data, idx;
    if (name in idx):
        print(f"Student {name} grades: ", end = "\n");
        students_data[idx[name]].grades.inorder(students_data[idx[name]].grades.root);
        print();
    else: print(f"Student {name} does not exist.");
```

輸入：學生名字
Inorder來印出AVL_Tree的data

6. Is Student Pass?

```
elif (choice == 6):
    print("Enter student name: ", end = "");
    name = input();
    ret = isPass(name);
    if (ret is None):
        print(f"Student {name} does not exist.");
    else:
        print(f"{name} pass:", end = "");
        if (len(ret) == 0): print(" None");
        else:
            for i in ret: print(f" {i}", end = ",\n"[i == ret[-1]]);
    print();
```

```
def isPass (name):
    global students_data, idx;
    ret = [];
    def inorder (p):
        if (p != None):
            inorder(p.left);
            if (p.score ≥ 60): ret.append(p.course);
            inorder(p.right);

    if (name in idx):
        tmp = students_data[idx[name]].grades.root;
        inorder(tmp);
        return ret;
    return None;
```

```
Enter your choice: 6

Enter student name: apple
apple pass: Financial Accounting, Calculus, Linear Algebra

continue? [y/n]
```

```
Enter your choice: 6

Enter student name: aa
aa pass: None

continue? [y/n]
```

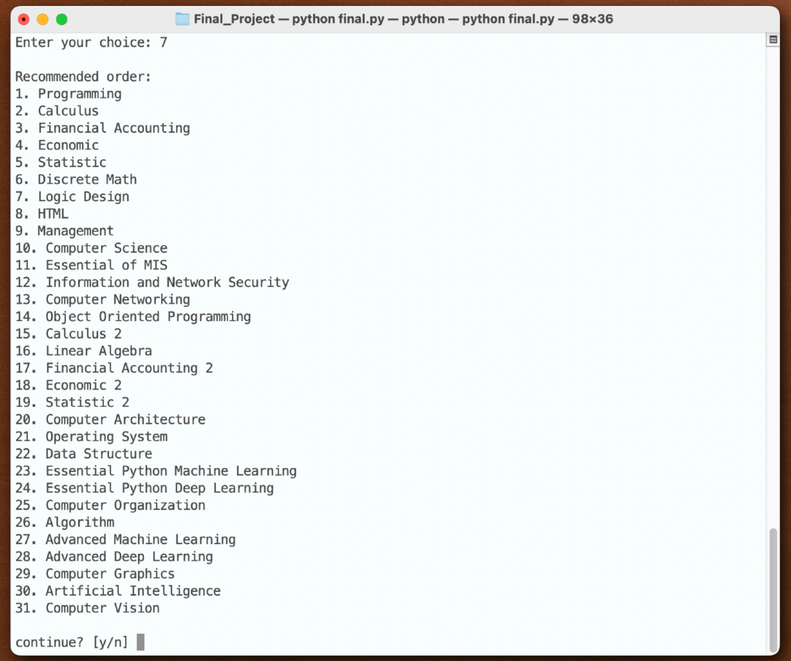
輸入：學生名字
查看學生有哪些科目是及格的
什麼都沒有 → 顯示空白

學生課程、成績管理系統

7. Get Recommended Order of Choosing Course

```
order = get_recommanded_order();
```

```
elif (choice == 7):
    print("Recommended order: ");
    for i in range(len(order)):
        print(f"{i + 1}. {order[i]}");
    print();
```



course

Subject	Prerequisite	Recommended_Grade
Programming	None	1
Calculus	None	1
Financial Accounting	None	1
Economic	None	1
Statistic	None	1
Financial Accounting 2	Financial Accounting	2
Economic 2	Economic	2
Statistic 2	Statistic	2
Calculus 2	Calculus	2
Object Oriented Programming	Programming	1
Data Structure	Object Oriented Programming	2
Algorithm	Data Structure	3
Discrete Math	None	1
Operating System	Computer Science	2
Linear Algebra	Calculus	1
Logic Design	None	1
HTML	None	1
Management	None	1
Essential Python Machine Learning	Object Oriented Programming	3
Essential Python Deep Learning	Object Oriented Programming	3
Computer Science	None	1
Essential of MIS	None	1
Information and Network Security	None	2
Computer Networking	None	2
Computer Architecture	Logic Design	2
Computer Organization	Computer Architecture	3
Computer Graphics	Algorithm	3
Computer Vision	Computer Graphics	4
Artificial Intelligence	Algorithm	3
Advanced Machine Learning	Essential Python Machine Learning	4
Advanced Deep Learning	Essential Python Deep Learning	4

```
def get_recommanded_order ():
    gp = {};
    In = {};
    grade = [[] for _ in range(5)];
    for i in range(len(Subjects)): gp[Subjects[i]] = [];
    for i in range(len(Subjects)):
        if (Prerequisite[i] != "None"):
            gp[Prerequisite[i]].append(Subjects[i]);
            if (Subjects[i] in In): In[Subjects[i]] += 1;
            else: In[Subjects[i]] = 1;
        else:
            # print(Recommended_grade[i]);
            grade[Recommended_grade[i]].append(Subjects[i]);

    q = Queue();
    for i in range(len(Subjects)):
        if (Subjects[i] not in In): q.push(Subjects[i]);

    order = [];
    nw = 1;
    rec = dict(zip(Subjects, Recommended_grade));

    while (not q.empty()):
        u = q.front();
        if (rec[u] > nw):
            nw = rec[u];
            for i in grade[nw]: order.append(i);
        if (u not in order): order.append(u);
        q.pop();
        for i in range(len(gp[u])):
            v = gp[u][i];
            In[v] -= 1;
            if (In[v] == 0): q.push(v);

    return order;
```

Topological Sort

8. Get Pass Student of the Course

```
elif (choice == 8):
    print("Enter course name: ", end = "");
    course = input();
    ret = get_pass_student(course);
    if (ret is not None and len(ret) != 0):
        print(f"Student who pass {course}: ", end = "");
        for i in ret: print(i, end = ",\n"[i == ret[-1]]);
    print();
```

```
def get_pass_student (course):
    if (course in course_data):
        subject_score[course], ret = passed_student(subject_score[course]);
        if (len(ret) == 0): print(f"No student passed {course} or No one choose this class.");
        return ret;
    else:
        print(f"Course {course} does not exist.");
        return None;
```

```
Enter your choice: 8

Enter course name: Calculus
Student who pass Calculus: fish,apple

continue? [y/n]
```

```
def passed_student (p, k = 60):
    a, b = split(p, k);
    ret = [];
    def inorder (p):
        if (p != None):
            inorder(p.lch);
            ret.append(p.name);
            inorder(p.rch);
    inorder(b);
    p = merge(a, b);
    return p, ret;
```

輸入：課程名稱
利用Treap將學生成績切成<60, ⩾60兩組，並且選⩾60的那組來return

Treap: 弱平衡二元搜尋樹
節點間符合BST的特性，且用Heap的特性來維持樹的平衡
⇒ Tree + Heap的資料結構

9. Get K Popular Courses

```
elif (choice == 9):
    print("Enter k: ", end = "");
    try: k = int(input());
    except ValueError:
        print("Please enter a number.\n");
        continue;
    if (k <= 0): print("Please enter a number greater than 0.");
    ret = get_k_popular(k);
    if (ret is not None and len(ret) != 0):
        a = 1;
        for i in ret:
            print(f"{a}. {i.name}, Number of people: {i.val}");
            a += 1;
    print();
```

```
def get_k_popular (k): return popularity.get_first_k_value(k);
```

```
def get_first_k_value (self, k):
    if (k > self.size):
        print("k cannot be larger than number of courses been chosen.");
        return;
    tmp = [];
    for i in range(k): tmp.append(self.pop());
    ret = tmp[0::];
    for i in range(k): self.insert(tmp[i].val, tmp[i].name);
    return ret;
```

```
Enter your choice: 9

Enter k: 3
1. Programming, Number of people: 3
2. Calculus, Number of people: 3
3. Linear Algebra, Number of people: 2

continue? [y/n]
```

輸入：k

在MaxHeap中尋找前 k 個data並return

10. Show Prerequisite of the Course

```
elif (choice == 10):
    print("Enter course name: ", end = "");
    course = input();
    if (course not in course_data):
        print(f"Course {course} does not exist.\n");
        continue;
    print(f"Prerequisite of {course}: ", end = "");
    print(course_data[course].prerequisites);
    print();
```

```
Enter your choice: 10

Enter course name: Algorithm
Prerequisite of Algorithm: Data Structure

continue? [y/n]
```

```
Enter your choice: 10

Enter course name: Calculus
Prerequisite of Calculus: None

continue? [y/n]
```

輸入：課程名稱
印出該課程的先修科目
沒有 → None

11. Show Student Who choose this Course

```
Enter your choice: 11

Enter course name: Calculus
Student who choose the course: apple,fish

continue? [y/n]
```

```
Enter your choice: 11

Enter course name: Algorithm
Student who choose the course: None

continue? [y/n]
```

```
elif (choice == 11):
    print("Enter course name: ", end = "");
    course = input();
    show_student_who_choose_course(course);
    print();
```

```
def show_student_who_choose_course (course):
    if (course in course_data):
        print("Student who choose the course: ", end = "");
        course_data[course].students.Print();
    else: print(f"Course {course} does not exist.");
```

輸入：課程名稱
課程先搶先贏 → Queue
印出Queue的順序
沒有 → None

12. Show All Students & ID

```
elif (choice == 12):
    print("All students: ", end = "\n");
    students_data = get_student_data();
    a = 1;
    for i in students_data:
        print(f"{a}. {i.name} {i.id}");
        a += 1;
    print();
```

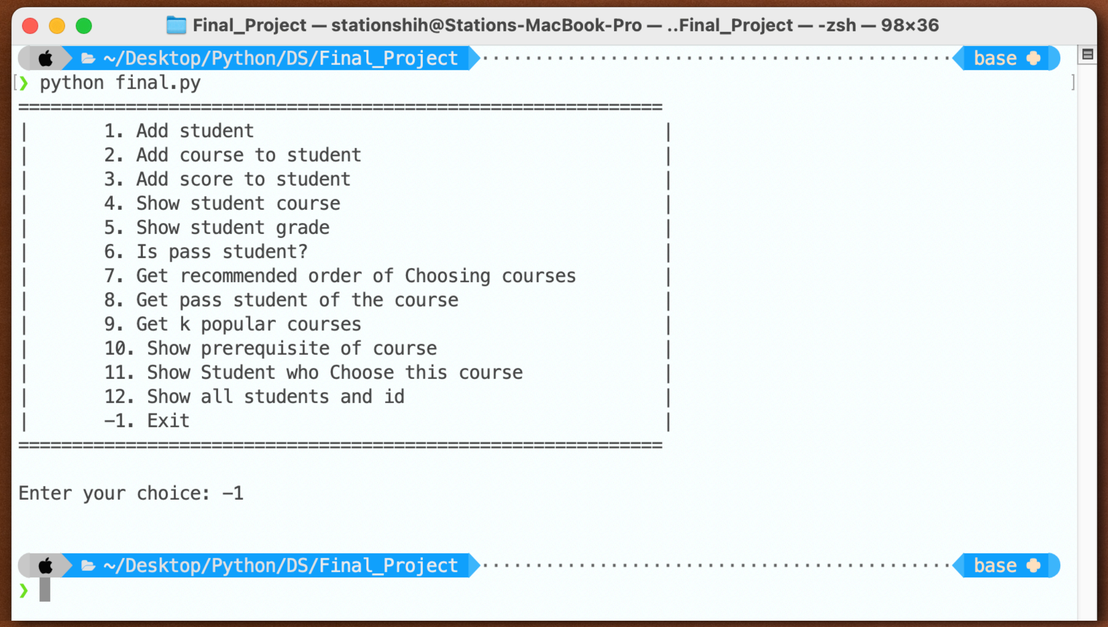
```
def get_student_data (): return students_data;
```

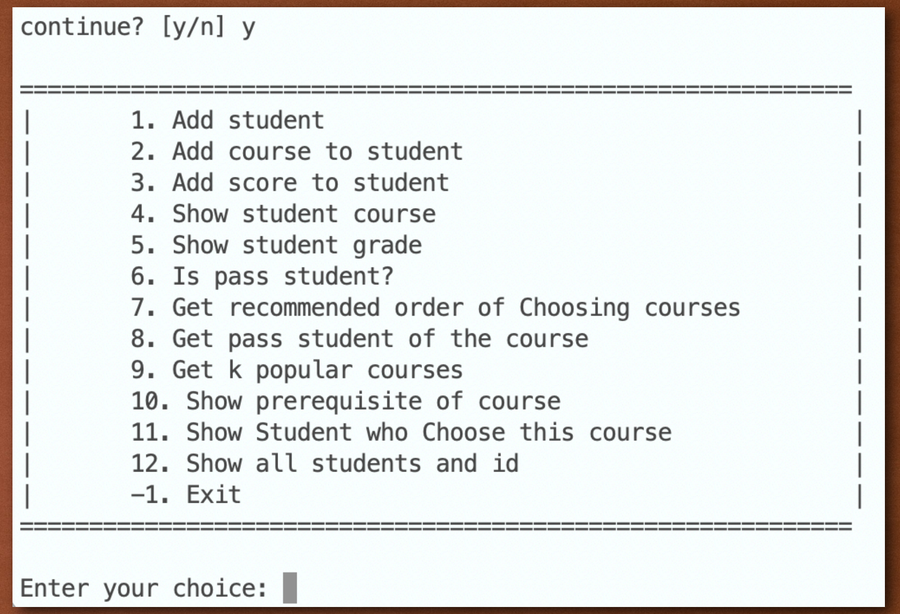
```
Enter your choice: 12

All students:
1. apple 4111029007
2. banana 4111029003
3. cat 4111029012
4. dog 4111029013
5. egg 4111029014
6. fish 4111029028
7. grape 4111029030
8. aa 123

continue? [y/n]
```

印出Student Data（按新增順序）
Student在Data中的Index → 用映射表來存

學生課程、成績管理系統

-1. Exit / Continue



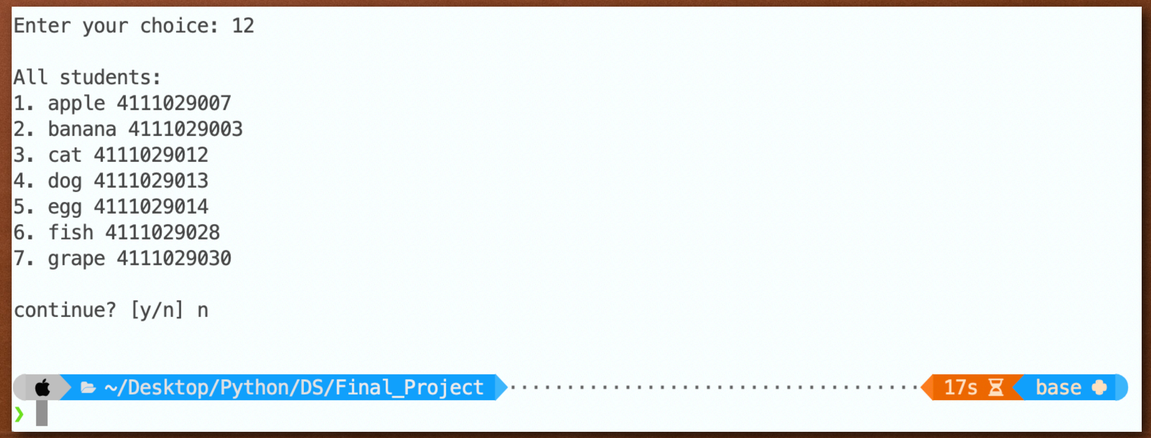


-1 → 結束選單

Continue:

Y or y → 繼續選單

Else → 結束選單



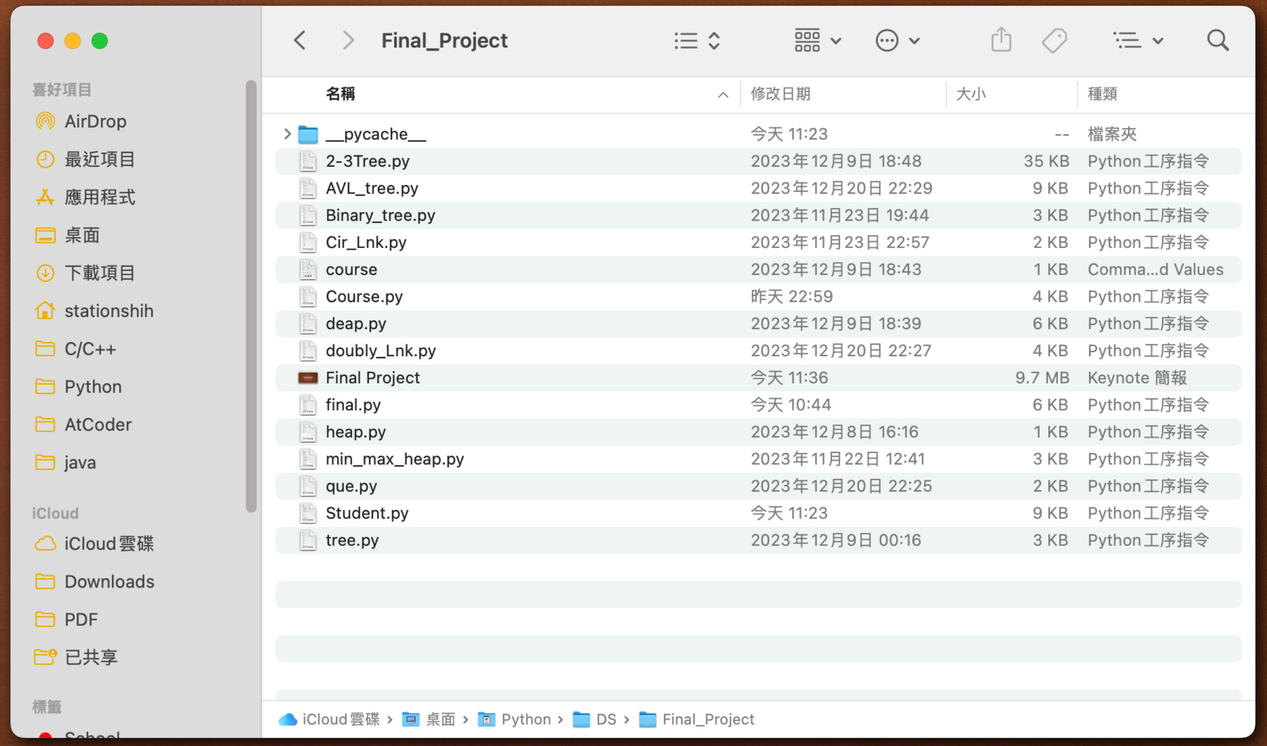
學生課程、成績管理系統

Others

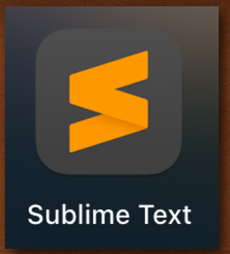
執行時在terminal輸入:
python final.py

```
def Insert_Default_Data ():
    add_student(4111029007, "apple");
    add_student(4111029003, "banana");
    add_student(4111029012, "cat");
    add_student(4111029013, "dog");
    add_student(4111029014, "egg");
    add_student(4111029028, "fish");
    add_student(4111029030, "grape");

    # print(Subjects);

    add_course_by_name("apple", "Calculus");
    add_course_by_name("apple", "Linear Algebra");
    add_course_by_name("banana", "Financial Accounting");
    add_course_by_name("banana", "Economic");
    add_course_by_name("cat", "Financial Accounting");
    add_course_by_name("cat", "Programming");
    add_course_by_name("dog", "Programming");
    add_course_by_name("dog", "Essential of MIS");
    add_course_by_id(4111029014, "Management");
    add_course_by_id(4111029014, "Essential of MIS");
    add_course_by_id(4111029028, "Calculus");
    add_course_by_id(4111029028, "Linear Algebra");
    add_course_by_id(4111029030, "Computer Science");
    add_course_by_id(4111029030, "Programming");
```

```
    add_score_by_name("apple", "Calculus", 100);
    add_score_by_name("apple", "Linear Algebra", 100);
    add_score_by_name("banana", "Financial Accounting", 60);
    add_score_by_name("banana", "Economic", 60);
    add_score_by_name("cat", "Financial Accounting", 40);
    add_score_by_name("cat", "Programming", 40);
    add_score_by_name("dog", "Programming", 80);
    add_score_by_name("dog", "Essential of MIS", 80);
    add_score_by_id(4111029014, "Management", 60);
    add_score_by_id(4111029014, "Essential of MIS", 60);
    add_score_by_id(4111029028, "Calculus", 80);
    add_score_by_id(4111029028, "Linear Algebra", 30);
    add_score_by_id(4111029030, "Computer Science", 100);
    add_score_by_id(4111029030, "Programming", 70);
```




```
courses = pd.read_csv("course.csv");

Subjects = courses["Subject"].tolist();
Prerequisite = courses["Prerequisite"].tolist();
Recommended_grade = courses["Recommended_Grade"].tolist();
```



謝謝！