

HƯỚNG DẪN CHI TIẾT

1. Viết chương trình nhập nội dung tập hợp hoa quả lên màn hình và lưu vào một tập hợp có tên là hoaqua. Việc nhập liệu kết thúc khi nhập vào ký tự 'q'.

Thực hiện các yêu cầu:

- In các loại quả trong tập hợp đã nhập ra màn hình.
- Kiểm tra xem một loại quả được nhập từ bàn phím có thuộc thuộc tập hợp hoaqua không?

Hướng dẫn:

```
1  hoaqua=set()
2  while True:
3      kt= input("Nhập tên loại quả, bấm 'q' để thoát: ")
4      if kt=='q':
5          break
6      hoaqua.add(kt)
7      print("Tập hợp quả đã nhập bao gồm có :",hoaqua)
8      print('Kiểm tra một loại quả có thuộc vào tập hợp hoa quả không.')
9      check_item = input("nhập tên quả: ")
10
11     if check_item in hoaqua:
12         print(check_item + " nằm trong tập hợp ",hoaqua)
13     else:
14         print(check_item + " không thuộc tập hợp ",hoaqua)
```

Kết quả chạy chương trình:

```
Nhập tên loại quả, bấm 'q' để thoát: Táo
Nhập tên loại quả, bấm 'q' để thoát: Cam
Nhập tên loại quả, bấm 'q' để thoát: Ổi
Nhập tên loại quả, bấm 'q' để thoát: Chuối
Nhập tên loại quả, bấm 'q' để thoát: Lê
Nhập tên loại quả, bấm 'q' để thoát: q
Tập hợp quả đã nhập bao gồm có : {'Ổi', 'Cam', 'Táo', 'Lê', 'Chuối'}
Kiểm tra một loại quả có thuộc vào tập hợp hoa quả không.
nhập tên quả: Xoài
Xoài không thuộc tập hợp {'Ổi', 'Cam', 'Táo', 'Lê', 'Chuối'}
```

2. Viết chương trình thực hiện các việc sau đối với Set:

- Thêm một phần tử
- Xóa bỏ một phần tử bằng remove();
- Xóa bỏ một phần tử bằng discard();

- d. Xóa bỏ một phần tử ngẫu nhiên
- e. Thêm nhiều phần tử vào Set.

Hướng dẫn:

Trong phần xóa phần tử chúng ta sử dụng cả remove và discard, có một sự khác biệt giữa hai hàm này đó là:

- Hàm remove(): Nếu phần tử cần xóa không tồn tại trong Set, hàm remove() sẽ gọi một ngoại lệ KeyError.
- Hàm discard(): Nếu phần tử cần xóa không tồn tại trong Set, hàm discard() sẽ không gọi một ngoại lệ và sẽ không làm bất kỳ thay đổi nào trong Set.
- Nếu chắc chắn rằng phần tử cần xóa tồn tại trong Set, thì có thể sử dụng remove(). Nếu không chắc chắn phần tử cần xóa tồn tại hay không, nên sử dụng discard() để tránh một ngoại lệ.

```
1 # Initialize a set
2 fruits = set(["apple", "banana", "cherry"])
3
4 # a. Add an element to the set
5 fruits.add("orange")
6 print("Sau khi thêm phần tử vào tập hợp:", fruits)
7
8 # b. Remove an element using remove()
9 fruits.remove("banana")
10 print('Tập hợp sau khi dùng hàm remove xóa phần tử
11 "banana": ', fruits)
12
13 # c. Remove an element using discard()
14 fruits.discard("cherry")
15 print("Sau khi dùng hàm discard xóa phần tử 'cherry':",
16 fruits)
17
18 # d. Remove a random element
19 random_fruit = fruits.pop()
20 print("Xóa ngẫu nhiên phần tử:", random_fruit)
21 print("Tập hợp hoa quả còn lại:", fruits)
22 # e. Add multiple elements to the set
```

```
23 fruits.update(["kiwi", "mango", "grapes"])
24 print("Sau khi thêm nhiều phần tử vào tập hợp:", fruits)
```

Kết quả thực hiện chương trình:

Sau khi thêm phần tử vào tập hợp: {'apple', 'banana', 'orange', 'cherry'}

Tập hợp sau khi dùng hàm remove xóa phần tử "banana": {'apple', 'orange', 'cherry'}

Sau khi dùng hàm discard xóa phần tử 'cherry': {'apple', 'orange'}

Xóa ngẫu nhiên phần tử: apple

Tập hợp hoa quả còn lại: {'orange'}

Sau khi thêm nhiều phần tử vào tập hợp: {'kiwi', 'grapes', 'mango', 'orange'}

3. Cho 2 tập hợp $A = \{1,2,3,4\}$; $B=\{3,4,5,6\}$

Thực hiện các yêu cầu sau:

- Viết chương trình kiểm tra tập hợp A có là con của tập hợp B không?
- Kiểm tra 2 tập hợp có rời nhau không?
- Viết code thực hiện các phép hợp, giao, hiệu, hiệu đối xứng.

Hướng dẫn:

```
1 A = {1, 2, 3, 4}
2 B = {5, 6, 7, 8}
3
4 if A.isdisjoint(B):
5     print("2 tập hợp rời nhau!")
6 else:
7     print("2 tập hợp không rời nhau")
8
9 # Phép hợp
10 print("Phép hợp: ", A.union(B))
11
12 # Phép giao
13 print("Phép giao: ", A.intersection(B))
14
15 # Phép hiệu
16 print("Phép hiệu: ", A.difference(B))
17
18 # Phép hiệu đối xứng
19 print("Phép hiệu đối xứng: ", A.symmetric_difference(B))
```

Kết quả thực hiện chương trình:

2 tập hợp rời nhau!

Phép hợp: {1, 2, 3, 4, 5, 6, 7, 8}

Phép giao: set()

Phép hiệu: {1, 2, 3, 4}

Phép hiệu đối xứng: {1, 2, 3, 4, 5, 6, 7, 8}

4. Viết chương trình tạo tập hợp A bao gồm 5 phần tử ngẫu nhiên lấy từ dãy 0,1,2,3,...,9.

Hướng dẫn:

```
1 import random
2
3 # Tạo tập hợp A với 5 phần tử ngẫu nhiên
4 A = set()
5 while len(A) < 5:
6     A.add(random.randint(0, 9))
7
8 print(A)
```

Kết quả thực hiện chương trình:

{0, 5, 6, 7, 8}

5. Cho tập hợp my_Set bao gồm các phần tử chỉ là số hoặc xâu ký tự. Chẳng hạn:

my_set={1,2, "Táo", 3.14, 'Pi'}

Viết chương trình tách tập hợp my_set thành hai tập con N (chỉ chứa số nguyên hoặc thập phân) và S(chỉ chứa xâu ký tự).

Hướng dẫn:

Trong bài tập này chúng ta sử dụng hàm isinstance() để kiểm tra một đối tượng có phải là một instance của một lớp (hoặc subclass) nào đó hay không. Nó trả về giá trị True nếu đối tượng đó là một instance của lớp được chỉ định hoặc giá trị False nếu không phải.

Dòng lệnh số 6:

```
if isinstance(i, (int, float)):
```

trong code đang kiểm tra xem biến i có phải là một kiểu số nguyên (int) hoặc là kiểu số thực (float) hay không. Nếu biến i là một số nguyên hoặc một số thực, thì câu lệnh if sẽ trả về giá trị True và thực hiện các câu lệnh trong khối if. Nếu biến i không phải là một số nguyên hoặc số thực, thì câu lệnh if sẽ trả về giá trị False và không thực hiện các câu lệnh trong khối if.

```
1 my_set = {1, 2, "apple", "banana", 3.14, "Pi"}
2 N = set()
3 S = set()
```

```

4
5 for i in my_set:
6     if isinstance(i, (int, float)):
7         N.add(i)
8     else:
9         S.add(i)
10
11 print("N:", N)
12 print("S:", S)

```

Kết quả thực hiện chương trình:

N: {1, 2, 3.14}

S: {'apple', 'Pi', 'banana'}

6. Viết chương trình tạo tập hợp A chỉ bao gồm các số nguyên tố nhỏ hơn N với N được nhập từ bàn phím.

```

1 A = set()
2 N = int(input("Nhập số N: "))
3 for i in range(2, N):
4     is_prime = True
5     for j in range(2, i):
6         if i % j == 0:
7             is_prime = False
8             break
9     if is_prime:
10        A.add(i)
11 print("Tập hợp A:", A)

```

Kết quả thực hiện chương trình:

Nhập số N: 100

Tập hợp A: {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}

7. Viết chương trình sinh ngẫu nhiên 2 tập hợp số tự nhiên A, B sau đó liệt kê ra màn hình các phần tử chung của A, B.

```

1 import random
2
3 A = set([random.randint(0,100) for i in range(10)])
4 B = set([random.randint(0,100) for i in range(10)])

```

```

5
6 print("A = ", A)
7 print("B = ", B)
8 print("Các phần tử chung của A, B là: ", A & B)

```

Chú ý: Trong câu lệnh ở dòng 3,4 ở code trên chúng ta đã sử dụng lệnh List Comprehension trong Python. Nó tạo ra một danh sách A, bao gồm 10 số nguyên tự nhiên được sinh ngẫu nhiên trong khoảng từ 0 đến 100.

Cụ thể, trong câu lệnh trên, **random.randint(0, 100)** sẽ trả về một số nguyên ngẫu nhiên trong khoảng từ 0 đến 100. **range(10)** sẽ tạo ra một dãy từ 0 đến 9. Cuối cùng, câu lệnh List Comprehension sẽ dùng các số từ range(10) làm chỉ số và sử dụng *random.randint(0,100)* để tạo ra 10 số nguyên ngẫu nhiên, sau đó gán giá trị này cho A.

Kết quả thực hiện chương trình:

```

A = {0, 9, 75, 11, 16, 49, 82, 23, 59, 94}
B = {100, 6, 40, 41, 10, 44, 82, 54, 30, 62}
Các phần tử chung của A, B là: {82}

```

8. Viết chương trình sinh các tập hợp A,B, C như sau:

A- là tập hợp 100 số ngẫu nhiên trong khoảng [1,1000]

B- là tập hợp 20 số ngẫu nhiên lấy từ tập hợp A.

C- là tập hợp 10 số ngẫu nhiên lấy từ B

```

1 import random
2 A = set(random.choices(range(1, 1001),k=100))
3 B=set()
4 C=set()
5
6 temp1=list(A)
7 while len(B) < 20:
8     b=random.choice(temp1)
9     B.add(b)
10 temp2=list(B)
11 while len(C)<10:
12     c=random.choice(temp2)
13     C.add(c)
14 print("Tập hợp A=",A)
15 print("Tập hợp B=",B)
16 print("Tập hợp C=",C)

```

Chú ý: Trong đoạn code trên dòng lệnh số 2, sử dụng hàm `choices()` trong thư viện `random`, đoạn mã `"A = set(random.choices(range(1, 1001), k=100))"` tạo một tập hợp A bằng cách chọn ngẫu nhiên 100 phần tử từ dãy số từ 1 đến 1000 bằng hàm `random.choices`. Hàm `random.choices` sẽ trả về một danh sách các số ngẫu nhiên được chọn từ dãy số được chỉ định. Cuối cùng, danh sách đó được chuyển sang một tập hợp bằng cách sử dụng hàm `set()`.

Kết quả thực hiện chương trình.

Tập hợp A = {9, 13, 536, 537, 544, 547, 39, 53, 566, 55, 567, 60, 577, 80, 593, 594, 89, 93, 608, 107, 112, 114, 117, 632, 641, 131, 140, 653, 657, 146, 670, 163, 675, 174, 688, 705, 217, 219, 734, 739, 755, 246, 261, 777, 278, 798, 801, 295, 808, 309, 836, 849, 856, 349, 864, 883, 887, 891, 894, 385, 386, 899, 388, 898, 396, 400, 404, 405, 411, 416, 928, 421, 939, 437, 455, 972, 463, 464, 977, 976, 978, 989, 478, 483, 484, 998, 999, 486, 495, 496}

Tập hợp B = {385, 641, 131, 404, 536, 411, 416, 544, 174, 705, 577, 836, 978, 217, 349, 478, 864, 495, 117, 894}

Tập hợp C = {544, 705, 641, 131, 174, 495, 536, 217, 411, 349}

9. Viết chương trình sinh các tập hợp A, B có ý nghĩa như sau:

A- là tập hợp 10 số tự nhiên ngẫu nhiên lấy từ khoảng [2,1000]

B- Tập tất cả các số là ước số nguyên tố của các số lấy từ A. Tức là một số p thuộc B khi và chỉ khi: p là số nguyên tố và p là ước số của một số nằm trong A. (Chú ý: ước số ở đây có thể bao gồm cả 1 và chính nó)

Hướng dẫn:

```
1 import random
2 import math
3
4 # Tạo tập hợp A với 10 phần tử ngẫu nhiên trong khoảng từ 2 đến 1001
5 A = set(random.choices(range(2, 1001), k=10))
6
7 # Tạo tập hợp B
8 B = set()
9
10 # Duyệt qua từng phần tử trong tập hợp A
11 for number in A:
12     for i in range(2, number + 1):
13         if number % i == 0: #nếu i là ước của number
14             is_prime = True #kiểm tra xem i có là số nguyên tố?
15             # Duyệt từ 2 đến căn bậc 2 của number
16             for j in range(2, int(math.sqrt(i)) + 1):
17                 # Nếu tìm thấy ước chung, i không phải là số nguyên tố
18                 if i % j == 0:
19                     is_prime = False
```

```

15         break
16         # Nếu i là số nguyên tố, thêm vào B
17         if is_prime:
18             B.add(i)
19 # In tập hợp A và tập hợp B
20 print("Tập hợp A: ", A)
21 print("Tập hợp B: ", B)

```

Kết quả thực hiện chương trình:

Lần 1:

Tập hợp A: {258, 2, 228, 836, 461, 464, 788, 533, 437, 959}

Tập hợp B: {2, 3, 197, 7, 41, 137, 43, 11, 461, 13, 19, 23, 29}

Lần 2:

Tập hợp A: {962, 997, 902, 270, 463, 847, 818, 88, 190}

Tập hợp B: {2, 3, 997, 37, 5, 7, 41, 11, 13, 463, 19, 409}

Lần 3:

Tập hợp A: {928, 225, 994, 8, 233, 907, 49, 530, 377, 351}

Tập hợp B: {2, 3, 5, 7, 71, 233, 907, 13, 53, 29}

10. Viết chương trình nhận phần thưởng của một lớp học:

Một lớp học có k sinh viên được đánh mã số sinh viên theo thứ tự từ 1 đến k. Nhà trường phát phần thưởng cho sinh viên thành 2 đợt.

Đợt 1 gồm m sinh viên có mã số từ I_1, I_2, \dots, I_m .

Đợt 2 gồm n em có số thứ tự II_1, II_2, \dots, II_n

Trong mỗi đợt phát thưởng mỗi em trong danh sách sẽ được nhận phần thưởng là giấy khen và 500,000 đồng.

Thực hiện yêu cầu:

Viết chương trình nhập k và danh sách các sinh viên được nhận thưởng đợt 1, đợt 2. Đưa ra danh sách các sinh viên được thưởng 1000,000 đồng, 500,000đ và các em không nhận được tiền thưởng.

Hướng dẫn:

Xem danh sách sinh viên được thưởng đợt 1 là tập hợp A. Danh sách các sinh viên được thưởng đợt 2 là tập hợp B. C là tập gồm k số 1, 2, ..., n.

$$A = \{ I_1, I_2, \dots, I_m \}$$

$$B = \{ II_1, II_2, \dots, II_n \}$$

Như vậy những sinh viên được thưởng 1,000,000đ (2 đợt) chính là các sinh viên có mã số(số thứ tự) thuộc về cả tập hợp A và tập hợp B, tức là thuộc A & B.

Những sinh viên được thưởng một lần (nhận 500,000đ) là những bạn có mã số hay số thứ tự chỉ nằm ở một trong 2 tập hợp A hoặc B mà không thuộc cả 2. Tức là thuộc tập hợp $A \Delta B$ (hiệu đối xứng của A và B).

Những sinh viên không được thưởng sẽ thuộc tập hợp các phần tử trong lớp (thuộc C) nhưng không nằm trong A và không nằm trong B tức là tập hợp $C - (A \cup B)$.

Dưới đây là code của chương trình

```
1  # Khởi tạo 2 set để lưu danh sách sinh viên nhận thưởng đợt 1 và đợt 2
2  A = set()    #Tập hợp các sinh viên nhận thưởng đợt 1
3  B = set()    #Tập hợp các sinh viên nhận thưởng đợt 2
4  C = set()    #Tập hợp gồm k số sinh viên trong lớp
5  # Nhập số lượng sinh viên trong lớp
6  k = int(input("Nhập số lượng sinh viên trong lớp: "))
7  C = {i for i in range(1,k+1)}
8
9  # Nhập số lượng sinh viên nhận thưởng đợt 1
10 m = int(input("Nhập số lượng sinh viên nhận thưởng đợt 1: "))
11 for i in range(m):
12     sv_number = int(input("Nhập mã số sinh viên nhận thưởng đợt 1: "))
13     # Thêm sinh viên vào set
14     A.add(sv_number)
15
16 # Nhập số lượng sinh viên nhận thưởng đợt 2
17 n = int(input("Nhập số lượng sinh viên nhận thưởng đợt 2: "))
18 for i in range(n):
19     sv_number = int(input("Nhập mã sinh viên nhận thưởng đợt 2: "))
20     # Thêm sinh viên vào set
21     B.add(sv_number)
22
23 # Tạo set để lưu danh sách sinh viên nhận đầy đủ 2 đợt thưởng
24 set1 = A.intersection(B) #A & B
25 # Tạo set để lưu danh sách sinh viên nhận 1 đợt thưởng
26 set2 = A^B               #Hiệu đối xứng/symetric difference
27 #Tạo tập hợp những sinh viên không được thưởng
28 set0 = C - (A|B)
29 # In ra danh sách sinh viên nhận đầy đủ 2 đợt thưởng
```

```

30 print("Các sinh viên được nhận đầy đủ 2 đợt thưởng : ", set1)
31 print("Các sinh viên được thưởng 1 đợt: ", set2)
32 print("Danh sách sinh viên không được thưởng: ", set0)

```

Kết quả thực hiện chương trình:

Nhập số lượng sinh viên trong lớp: 10
 Nhập số lượng sinh viên nhận thưởng đợt 1: 3
 Nhập SBD sinh viên nhận thưởng đợt 1: 2
 Nhập SBD sinh viên nhận thưởng đợt 1: 3
 Nhập SBD sinh viên nhận thưởng đợt 1: 7
 Nhập số lượng sinh viên nhận thưởng đợt 2: 4
 Nhập SBD sinh viên nhận thưởng đợt 2: 1
 Nhập SBD sinh viên nhận thưởng đợt 2: 2
 Nhập SBD sinh viên nhận thưởng đợt 2: 3
 Nhập SBD sinh viên nhận thưởng đợt 2: 6
 Danh sách sinh viên nhận đầy đủ 2 đợt thưởng: {2, 3}
 Danh sách sinh viên nhận 1 đợt thưởng: {1, 6, 7}
 Danh sách sinh viên không được thưởng: {4, 5, 8, 9, 10}

11. Danh sách đại diện.

Nhập danh sách A chứa n số nguyên. Tạo danh sách B chỉ gồm các ký tự đại diện trong A, tức là nếu phần tử nào xuất hiện nhiều lần trong A thì chỉ lấy 1 phần tử đưa vào B. Các phần tử trong danh sách B được sắp xếp tăng dần.

Hướng dẫn:

```

1  # Nhập danh sách A chứa n số nguyên
2  #A = list(map(int, input("Nhập danh sách A: ").split()))
3  n=int(input('Nhập số phần tử của danh sách:'))
4  print('Nhập danh sách:')
5  A=[int(input()) for i in range(n)]
6
7  # Tạo danh sách B chỉ gồm các ký tự đại diện trong A
8  B = list(set(A))
9
10 # Sắp xếp các phần tử trong B tăng dần
11 for i in range(len(B)):
12     for j in range(i+1, len(B)):
13         if B[i] > B[j]:

```

```

14         B[i], B[j] = B[j], B[i]
15
16 # In danh sách B
17 print("Danh sách B: ", end="")
18 print(*B)          #ký tự '*' cho phép in ra các phần tử của B
19                   # mà không nằm trong cặp '{}'

```

Kết quả thực hiện chương trình;

Nhập số phần tử của danh sách:6

Nhập danh sách:

1
5
9
1
8
3

Danh sách B: 1 3 5 8 9

Bài tập về từ điển.

12. Viết chương trình sử dụng dictionary lưu các thông tin về Tivi bao gồm (Hãng, model, năm sản xuất) in nội dung thông tin ra màn hình.

Hướng dẫn:

```

1  tv_1 = {"Hãng": "Samsung", "Model": "UE43TU7000", "Năm sản xuất": 2020}
2  tv_2 = {"Hãng": "LG", "Model": "OLED55CX", "Năm sản xuất": 2019}
3  tv_3 = {"Hãng": "Sony", "Model": "XH95", "Năm sản xuất": 2021}
4
5  tivi = [tv_1, tv_2, tv_3]
6
7  for tv in tivi:
8      print("Thông tin tivi:")
9      print("Hãng:", tv["Hãng"])
10     print("Model:", tv["Model"])
11     print("Năm sản xuất:", tv["Năm sản xuất"])
12     print("-" * 30)

```

Kết quả thực hiện chương trình:

Thông tin tivi:

Hãng: Samsung

Model: UE43TU7000

Năm sản xuất: 2020

Thông tin tivi:

Hãng: LG

Model: OLED55CX

Năm sản xuất: 2019

Thông tin tivi:

Hãng: Sony

Model: XH95

Năm sản xuất: 2021

13. Viết chương trình kiểm tra một key có tồn tại trong từ điển không?

Hướng dẫn:

```
1 # Tạo một từ điển chứa các thông tin về Tivi
2 television_dict = {"Hãng": "Samsung", "Model": "UE49NU7100", "Năm sản
3 xuất": 2021}
4
5 # Key cần kiểm tra
6 key = input("Nhập từ khóa(key) cần kiểm tra: ")
7
8 # Sử dụng toán tử in để kiểm tra key có tồn tại trong từ điển hay
9 không
10 if key in television_dict:
11     # Nếu tồn tại, in ra thông báo và giá trị tương ứng
12     print("Key '{}' tồn tại trong từ điển với giá trị là
13 '{}'.format(key, television_dict[key]))
14 else:
15     # Nếu không tồn tại, in ra thông báo
16     print("Key '{}' không tồn tại trong từ điển".format(key))
```

Kết quả thực hiện chương trình:

Lần 1:

Nhập từ khóa(key) cần kiểm tra: Hãng

Key 'Hãng' tồn tại trong từ điển với giá trị là 'Samsung'

Lần 2:

Nhập từ khóa(key) cần kiểm tra: Hãng

Từ khóa(Key) 'Hãng' tồn tại trong từ điển với giá trị là 'Samsung'

Lần 3:

Nhập từ khóa(key) cần kiểm tra: Model

Từ khóa(Key) 'Model' tồn tại trong từ điển với giá trị là 'UE49NU7100'

Lần 4:

Nhập từ khóa(key) cần kiểm tra: Năm sản xuất

Từ khóa(Key) 'Năm sản xuất' tồn tại trong từ điển với giá trị là '2021'

Lần 5:

Nhập từ khóa(key) cần kiểm tra: Sony

Khóa(Key) 'Sony' không tồn tại trong từ điển

14. Giả sử đã có một dữ liệu từ điển thông tin học sinh bao gồm cặp <key>:<value> là <Họ tên>:<Điểm số>. Chẳng hạn bộ dữ liệu gốc là:

Dic = {"Nguyễn Văn A":9,0, "Lê Thị B":8.5}

Viết một chương trình có chức năng cập nhật điểm số sinh viên vào bộ dữ liệu đã có trên.

Chương trình được thực hiện như sau:

Hai tham số được lần lượt nhập vào:

a. Nhập họ tên sinh viên:

b. Nhập điểm số:

Các thông tin trên sẽ được cập nhật vào bộ dữ liệu gốc, hoặc tạo mới một bản ghi, hoặc cập nhật chỉnh sửa điểm của sinh viên đã có trong danh sách. Chương trình sẽ đưa ra thông báo như sau:

Đã tạo mới thêm dữ liệu họ tên và điểm một sinh viên.

Hoặc

Đã cập nhật lại thông tin điểm của sinh viên này.

Hướng dẫn.

```
1 dic = {"Nguyễn Văn A": 9.0, "Lê Thị B": 8.5}
2 old=dic.copy()
3 name = input("Nhập họ tên sinh viên: ")
4 diem = float(input("Nhập điểm số: "))
5
6 if name in dic:
7     dic[name] = diem
8     print("Đã cập nhật lại thông tin điểm của sinh viên này.")
```

```

9  else:
10     dic[name] = diem
11     print("Đã tạo mới thêm dữ liệu họ tên và điểm một sinh viên.")
12
13 print('Dữ liệu ban đầu:\n',old)
14 print('Dữ liệu sau khi cập nhật hoặc thêm mới: \n',dic)

```

Kết quả thực hiện chương trình

Lần 1:

Nhập họ tên sinh viên: Trần Văn C

Nhập điểm số: 7.0

Đã tạo mới thêm dữ liệu họ tên và điểm một sinh viên.

Dữ liệu ban đầu:

{'Nguyễn Văn A': 9.0, 'Lê Thị B': 8.5}

Dữ liệu sau khi cập nhật hoặc thêm mới:

{'Nguyễn Văn A': 9.0, 'Lê Thị B': 8.5, 'Trần Văn C': 7.0}

Lần 2:

Nhập họ tên sinh viên: Nguyễn Văn A

Nhập điểm số: 6.5

Đã cập nhật lại thông tin điểm của sinh viên Nguyễn Văn A

Dữ liệu ban đầu:

{'Nguyễn Văn A': 9.0, 'Lê Thị B': 8.5}

Dữ liệu sau khi cập nhật hoặc thêm mới:

{'Nguyễn Văn A': 6.5, 'Lê Thị B': 8.5}

15. Một từ điển D được định nghĩa bao gồm Khóa là mã sinh viên trong lớp, Value là cặp 3 giá trị tên, điểm HP1, điểm HP2. Ví dụ về một bộ dữ liệu từ điển như vậy:

D={"K1601":["Hà", 7, 9.5], "K1602":["Bình", 10, 8], "K1603":["Hoa", 6.0, 9]}

Viết chương trình nhập bộ dữ liệu từ điển trên như sau:

- Đầu tiên yêu cầu người dùng nhập mã sinh viên.
- Tiếp theo yêu cầu nhập, tên, điểm HP1 và điểm HP2 trên một dòng bởi 1 chuỗi ký tự, và 2 số, cách nhau bởi dấu phẩy.
- Thông báo loại 1: "Đã hoàn thành nhập điểm cho sinh viên mới.
- Thông báo loại 2: "Đã hoàn thành cập nhật lại nhập điểm cho sinh viên tron danh sách.

Hướng dẫn:

```

1  # Tạo từ điển D
2  D = {"K1601": ["Hà", 7, 9.5], "K1602": ["Bình", 10, 8], "K1603": ["Hoa", 6.0, 9]}

```

```

3 old=D.copy()          #Lưu dữ liệu ban đầu
4 # Nhập mã sinh viên
5 student_id = input("Nhập mã sinh viên: ")
6
7 # Nhập tên, điểm HP1, điểm HP2
8 student_info = input("Nhập tên, điểm HP1, điểm HP2 của sinh viên (ngăn
9 cách nhau bởi dấu phẩy): ")
10 student_info = student_info.split(",")
11 student_info = [student_info[0], float(student_info[1]),
12 float(student_info[2])]
13
14 # Thêm hoặc cập nhật thông tin sinh viên trong từ điển D
15 if student_id in D:
16     D[student_id] = student_info
17     print("Đã hoàn thành cập nhật lại nhập điểm cho sinh viên trong danh sách.")
18 else:
19     D[student_id] = student_info
20     print("Đã hoàn thành nhập điểm cho sinh viên mới.")
21
22     print('Dữ liệu ban đầu:\n',old)
23 print('Dữ liệu sau khi cập nhật hoặc thêm mới: \n',D)

```

Kết quả thực hiện chương trình:

Nhập mã sinh viên: KH1604

Nhập tên, điểm HP1, điểm HP2 của sinh viên (ngăn cách nhau bởi dấu phẩy): Tuấn,
5.0,8.5

Đã hoàn thành nhập điểm cho sinh viên mới.

Dữ liệu ban đầu:

```
{'K1601': ['Hà', 7, 9.5], 'K1602': ['Bình', 10, 8], 'K1603': ['Hoa', 6.0, 9]}
```

Dữ liệu sau khi cập nhật hoặc thêm mới:

```
{'K1601': ['Hà', 7, 9.5], 'K1602': ['Bình', 10, 8], 'K1603': ['Hoa', 6.0, 9], 'KH1604': ['Tuấn', 5.0, 8.5]}
```

16. Cho trước bộ dữ liệu điểm các học phần 1, học phần 2 và học phần 3 của các sinh viên K16.

Ví dụ: D={"K1601":["Hà", 7,9.5], "K1602":["Bình",10,8], "K1603":["Hoa, 6.0,9]}

Viết chương trình tính điểm trung bình của K16 và in ra màn hình nội dung như sau:

Điểm trung bình:

K1601

Hà: 8.25

K1602

Bình: 9.0

K1603

Hoa: 7.5

Hướng dẫn:

```
1 # bộ dữ liệu điểm các học phần 1, 2 và 3 của các sinh viên K16
2 D = {"K1601":["Hà", 7, 9.5], "K1602":["Bình",10, 8], "K1603":["Hoa", 6.0, 9]}
3
4 # in ra tiêu đề
5 print("Điểm trung bình học kỳ 1:")
6 print("-"*15)
7 #duyet qua từng phần tử trong bộ dữ liệu
8 for key, value in D.items():
9     #tính điểm trung bình
10     average = (value[1] + value[2]) / 2
11
12     #in ra mã sinh viên và điểm trung bình
13     print(key)
14     print(value[0] + ": " + str(average))
15     print(".*20)
```

Kết quả thực hiện chương trình:

Điểm trung bình học kỳ 1:

K1601

Hà: 8.25

.....

K1602

Bình: 9.0

.....

K1603

Hoa: 7.5

.....

17. Cho trước 3 bộ dữ liệu từ điển các học phần của một lớp sinh viên KHDL K16. Cả 3 bộ dữ liệu này đều có khuôn dạng giống nhau với bộ <key-value> với key là mã sinh viên gồm 2 chữ số, value là một danh sách bao gồm: tên sinh viên, điểm số.

Giả sử chúng ta có bộ dữ liệu điểm 3 học phần Đại số tuyến tính viết tắt là DS, Giải tích viết tắt là GT và Tin cơ sở viết tắt là TN như sau:

DS={"04":["Hà", 7], "02":["Bình",8], "03":["Hoa", 9], "05":["Sơn",6], "08":["Hùng",10]}

GT={"04":["Hà", 9], "06":["Ngọc",8], "09":["Vân", 7], "05":["Sơn",6], "08":["Hùng",10]}

TN={"04":["Hà", 7], "02":["Bình",8], "01":["An", 9], "05":["Sơn",6], "08":["Hùng",10]}

Yêu cầu là tạo ra một bộ dữ liệu từ điển mới đặt tên là DTB sẽ tính điểm trung bình cho cả 3 học phần trên.

Tuy nhiên do dữ liệu từng môn có thể chưa đầy đủ nên danh sách tên học sinh của các học phần trên có thể còn bị thiếu. Vì vậy, yêu cầu chương trình chỉ tính điểm trung bình cho các sinh viên có đủ điểm cả 3 học phần. Chương trình sẽ bỏ qua không tính điểm trung bình các sinh viên thiếu điểm của một trong 3 học phần trên.

Hướng dẫn:

```
1 DS={"04":["Hà", 7], "02":["Bình",8], "03":["Hoa", 9], "05":["Sơn",6], "08":["Hùng",10]}
2 GT={"04":["Hà", 7], "06":["Ngọc",8], "09":["Vân", 9], "05":["Sơn",6], "08":["Hùng",10]}
3 TN={"04":["Hà", 7], "02":["Bình",8], "01":["An", 9], "05":["Sơn",6], "08":["Hùng",10]}
4
5 DTB = {}
6 print("Các sinh viên sau có điểm trung bình là:")
7 for key in DS.keys():
8     if key in GT.keys() and key in TN.keys():
9         print(key,end=" ")
10        print(DS[key][0], end=" ")
11        avg = (DS[key][1] + GT[key][1] + TN[key][1]) / 3
12        print(round(avg,2))
13        DTB[key] = [DS[key][0], avg]
14 print('Bộ dữ liệu điểm trung bình DTB là:')
15 print('DTB=',DTB)
```

Chú ý:

Trong code trên, ở dòng lệnh số 10 viết '**DS[key][0]**' có ý nghĩa là lấy giá trị đầu tiên của thành phần giá trị tương ứng với key trong từ điển DS.

Ví dụ: `DS["04"][0]` sẽ trả về giá trị 'Hà', vì 'Hà' là giá trị đầu tiên trong thành phần giá trị tương ứng với key "04" trong từ điển DS.

Dòng lệnh số 13, mã lệnh '`DTB[key]=[DS[key][0],avg]`' có nghĩa là tạo mới một mục trong từ điển DTB với key là key của mục tương tự trong từ điển DS. Giá trị của mục mới sẽ là một danh sách gồm 2 phần tử, gồm phần tử đầu tiên của giá trị tương tự của mục tương tự trong từ điển DS, và phần tử thứ hai là biến avg.

Ví dụ, nếu key là "04", thì mục tương tự trong từ điển DS sẽ là ["Hà", 7], và mục mới trong từ điển DTB sẽ là {"04": ["Hà", avg]}.

Kết quả thực hiện chương trình:

Các sinh viên sau có điểm trung bình là:

04 Hà 7.0

05 Sơn 7.0

08 Hùng 9.0

Bộ dữ liệu điểm trung bình DTB là:

DTB= {'04': ['Hà', 7.0], '05': ['Sơn', 7.0], '08': ['Hùng', 9.0]}

18. Cho trước một đoạn mã văn bản tiếng Anh:

Str="

”

Viết chương trình đếm xem đoạn văn bản trên có bao nhiêu từ và tần xuất xuất hiện các từ trong chuỗi Str.

19. Viết chương trình nhập lần lượt thông tin sinh viên trong lớp theo các thông số sau:

Nhập mã sinh viên 04 ký tự là chữ số

Nhập họ tên sinh viên

Nhập điểm học phần 1 (HP1) là số thập phân với 1 chữ số sau dấu phẩy.

Nhập điểm học phần 2 (HP2) là số thập phân với 1 chữ số sau dấu phẩy.

Việc nhập liệu kết thúc khi nhập vào một xâu rỗng.

Sau đó in ra màn hình họ tên học sinh, điểm HP1, HP2 theo mẫu sau:

Mã SV	Họ tên	Điểm HP1	Điểm HP 2
0002	Nguyễn Văn B	8.5	9.3
0001	Lê Thị A	7.8	7.6
0003	Trần Văn C	10	8.5

Yêu cầu thể hiện ra như sau:

- Mã sinh viên được căn trái, chiếm độ rộng 08 ký tự
- Họ tên sinh viên được căn trái, chiếm độ rộng 30 ký tự
- Điểm HP1, chiếm độ rộng 05 ký tự, căn phải

- Điểm HP2, chiếm độ rộng 05 ký tự, căn phải.

Hướng dẫn

```
1 # Thông tin sinh viên
2 sinh_vien = []
3
4 # Bắt đầu nhập liệu
5 while True:
6     # Nhập mã sinh viên (04 ký tự)
7     ma_sinh_vien = input("Nhập mã sinh viên (04 ký tự): ")
8
9     # Nếu mã sinh viên rỗng, kết thúc nhập liệu
10    if ma_sinh_vien == "":
11        break
12
13    # Nhập họ tên sinh viên
14    ho_ten = input("Nhập họ tên sinh viên: ")
15
16    # Nhập điểm HP1
17    diem_hp1 = float(input("Nhập điểm HP1: "))
18
19    # Nhập điểm HP2
20    diem_hp2 = float(input("Nhập điểm HP2: "))
21
22    # Nhập điểm HP3
23    diem_hp3 = float(input("Nhập điểm HP3: "))
24
25    # Thêm thông tin sinh viên vào danh sách
26    sinh_vien.append({
27        "ma_sinh_vien": ma_sinh_vien,
28        "ho_ten": ho_ten,
29        "diem_hp1": diem_hp1,
30        "diem_hp2": diem_hp2,
31        "diem_hp3": diem_hp3
32    })
33
34 # In thông tin sinh viên ra màn hình
35 print()
36 print("-----THÔNG TIN SINH VIÊN-----")
37 print("-"*60)
38 print("Mã SV".ljust(8), "Họ tên".ljust(20), "Điểm HP1".rjust(5), "Điểm HP2".rjust(5))
39 print("-"*60)
40 for sv in sinh_vien:
41     print(sv["ma_sinh_vien"].ljust(8),sv["ho_ten"].ljust(20),\
```

```
str(sv["diem_hp1"]).rjust(5), str(sv["diem_hp2"]).rjust(5))
```

Kết quả thực hiện chương trình.

Nhập mã sinh viên (04 ký tự): 0002
Nhập họ tên sinh viên: Nguyễn Văn B
Nhập điểm HP1: 8.5
Nhập điểm HP2: 9.3
Nhập mã sinh viên (04 ký tự): 0001
Nhập họ tên sinh viên: Lê Thị A
Nhập điểm HP1: 7.8
Nhập điểm HP2: 7.6
Nhập mã sinh viên (04 ký tự): 0003
Nhập họ tên sinh viên: Trần Văn C
Nhập điểm HP1: 10
Nhập điểm HP2: 8.5
Nhập mã sinh viên (04 ký tự):

-----THÔNG TIN SINH VIÊN-----

Mã SV	Họ tên	Điểm HP1	Điểm HP2
0002	Nguyễn Văn B	8.5	9.3
0001	Lê Thị A	7.8	7.6
0003	Trần Văn C	10.0	8.5