

5	W6A5	Nhập một list số nguyên, In ra dictionary {'positives': ..., 'negatives': ..., 'zeros': ...} (đếm số >0, <0, =0)	<pre> 1 -2 0 3 -4 0 0 0 0 0 -1 -2 -3 5 1 -1 1 -1 </pre>	<pre> {'positives': 2, 'negatives': 2, 'zeros': 3} {'positives': 0, 'negatives': 0, 'zeros': 3} {'positives': 0, 'negatives': 3, 'zeros': 3} {'positives': 1, 'negatives': 0, 'zeros': 3} {'positives': 2, 'negatives': 2, 'zeros': 3} </pre>
6	W6A6	Viết chương trình nhập vào một danh sách các số nguyên, sau đó in ra tổng các số trong list đó.	<pre> 1 2 3 4 0 0 0 10 -1 -2 3 100 200 300 </pre>	<pre> 10 0 10 0 600 </pre>
7	W6A7	Viết chương trình nhập vào một tuple, in ra phần tử đầu, phần tử cuối, và tuple đảo ngược	<pre> 5 8 2 7 1 4 1 10 20 3 6 9 a b c </pre>	<pre> 5 4 ('4', '1', '7', '2', '8', '5') 1 1 ('1',) 10 20 ('20', '10') 3 9 ('9', '6', '3') a c ('c', 'b', 'a') </pre>
8	W6A8	Viết chương trình nhập vào một list các chuỗi, sau đó đếm số lần xuất hiện của từng chuỗi và lưu vào dictionary.	<pre> apple banana apple a b a a c cat cat mouse dog dog cat bird x y z x y z 1 2 2 1 3 1 </pre>	<pre> {'apple': 2, 'banana': 1} {'a': 3, 'b': 1, 'c': 1} {'cat': 3, 'mouse': 1, 'dog': 2, 'bird': 1} {'x': 2, 'y': 2, 'z': 1} {'1': 3, '2': 2, '3': 1} </pre>

9	W6A9	Viết chương trình nhập vào hai dictionary. Sau đó gộp lại thành một dictionary mới trong đó nếu một key xuất hiện trong cả hai dict, giá trị sẽ được cộng lại, nếu chỉ có trong một dict thì giữ nguyên. In ra dictionary mới theo thứ tự từ điển tăng dần của key.	{'a': 1} {'a': 2} {'a': 1, 'b': 2} {'c': 3, 'b': 4} {} {'x': 1} {'x': 5} {} {'k': 1} {'m': 2, 'n': 3, 'k': 4}	{'a': 3} {'a': 1, 'b': 6, 'c': 3} {'x': 1} {'x': 5} {'k': 5, 'm': 2, 'n': 3}
10	W6A10	Viết chương trình nhập vào một danh sách các số nguyên, và số nguyên dương k. In ra list gồm các cặp số khác nhau có tổng bằng k. Mỗi cặp là một tuple, hai số trong tuple có thể giống nhau, thứ tự xuất hiện trong danh sách của số thứ nhất nhỏ hơn thứ tự của số thứ hai. Các tuple trong list được sắp xếp theo thứ tự tăng dần của phần tử đầu tiên trong tuple.	1 2 3 4 5 6 1 1 1 1 1 2 10 5 -5 0 5 1 2 3 2 1 4 1 2 3 4 10	[(1, 5), (2, 4)] [(1, 1)] [(5, 0), (10, -5)] [(1, 3), (2, 2), (3, 1)] []

11	W6A11	<p>[Tách số chẵn và lẻ từ tuple] Nhập vào một tuple chứa các số nguyên. Hãy tạo ra hai tuple mới: một chứa tất cả các số chẵn và một chứa tất cả các số lẻ từ tuple ban đầu, giữ nguyên thứ tự tương đối. In ra hai tuple này trên hai dòng riêng biệt.</p>	<p>1 10 2 9 3 8 4 7 5 6</p> <p>2 4 6 8 10 12</p> <p>1 3 5 7 9 11</p> <p>0 -1 -2 -3 -4 5</p> <p>99</p>	<p>(10, 2, 8, 4, 6) (1, 9, 3, 7, 5)</p> <p>(2, 4, 6, 8, 10, 12) ()</p> <p>() (1, 3, 5, 7, 9, 11)</p> <p>(0, -2, -4) (-1, -3, 5)</p> <p>() (99,)</p>
12	W6A12	<p>[Tìm phần tử xuất hiện nhiều nhất] Nhập vào một list các số nguyên. Hãy tìm số xuất hiện nhiều lần nhất trong list. Nếu có nhiều số cùng có số lần xuất hiện nhiều nhất, hãy in ra số nhỏ nhất trong số đó.</p>	<p>1 2 3 2 4 2 5</p> <p>5 5 5 1 1 1 8 9</p> <p>10 20 5 30 15</p> <p>-1 -2 -3 -1 -3 -4</p> <p>100</p>	<p>2</p> <p>1</p> <p>5</p> <p>-3</p> <p>100</p>
13	W6A13	<p>[Đảo ngược key và value của dictionary] Nhập vào một dictionary có value là duy nhất (không trùng nhau). Hãy tạo một dictionary mới bằng cách đảo ngược key và value của dictionary ban đầu. In ra dictionary mới.</p>	<p>a 1 b 2 c 3</p> <p>1 10 2 20</p> <p>k1 v1 k2 v2</p> <p>id 99</p> <p>code A-1 user B_2</p>	<p>{'1': 'a', '2': 'b', '3': 'c'}</p> <p>{'10': '1', '20': '2'}</p> <p>{'v1': 'k1', 'v2': 'k2'}</p> <p>{'99': 'id'}</p> <p>{'A-1': 'code', 'B_2': 'user'}</p>

14	W6A14	<p>[Giao của hai danh sách] Nhập vào hai dòng, mỗi dòng là một danh sách các số nguyên. Hãy tìm các phần tử chung (giao) của hai danh sách này và in ra một list mới chứa các phần tử chung đó, không trùng lặp.</p>	1 2 3 4 4 5 4 5 6 7 5 8 1 2 3 4 5 6 10 20 30 30 20 10 20 30 -1 -2 0 5 8 0 9 -1 10 -1 1 2 3 4 5	[4, 5] [] [10, 20, 30] [-1, 0] []
15	W6A15	<p>[Lọc dictionary theo giá trị] Nhập vào một dictionary có key là chuỗi và value là số nguyên, và một số nguyên k ở dòng tiếp theo. Hãy tạo một dictionary mới chỉ chứa các cặp key-value từ dictionary ban đầu mà có value lớn hơn k. In ra dictionary mới.</p>	math 95 science 88 english 92 history 90 apple 10 banana 20 orange 15 5 a 10 b 20 c 30 100 red 50 green 50 blue 60 50 x -5 y -10 z -2 -8	{'math': 95, 'english': 92} {'apple': 10, 'banana': 20, 'orange': 15} {} {'blue': 60} {'x': -5, 'z': -2}
16	W6A16	Nhập vào mảng n hàng, m cột số nguyên. In lại mảng ra màn hình như ma trận hai chiều, định dạng mỗi phần tử 4 vị trí căn phải		
17	W6A17	Nhập vào mảng vuông 2 chiều n hàng, n cột, in ra đường chéo chính và đường chéo phụ của mảng		
18	W6A18	Nhập vào mảng n hàng, m cột số nguyên và số nguyên k (n, m, k nhập trên cùng một dòng, cách nhau dấu cách). Tính và in ra tổng các phần tử trên cột k (hàng k)		

19	W6A19	Viết hàm đưa vào 1 list số nguyên, tìm và trả về vị trí có giá trị lớn nhất trong list	1 2 3 4 5 -6 7 8 19 34 25 67 89 100 1000 2 45 67 12 90 23 34 8 90 67 24 -45 -78 -29 -78 -199 -199	9 6 5 3 1
20	W6A20	Viết hàm đưa vào một list số nguyên và một số nguyên dương k. Hãy tìm và trả về vị trí của phần tử đầu tiên có giá trị k trong list, nếu không có thì trả về -1	1 2 3 4 5 -6 7 8 19 1 2 3 4 5 5 34 25 67 89 100 1000 34 25 67 89 100 1000 89 2 45 67 12 90 23 23 34 8 90 67 24 34 -45 -78 -29 -78 -199 -199 -29	5 4 6 1 3
21	W6A21	Nhập một dòng gồm các chuỗi (tách bởi dấu cách). Tạo dictionary ánh xạ chuỗi -> tuple các vị trí (0-based) mà chuỗi đó xuất hiện	a b a c b a x cat dog cat mouse 1 1 1 a b c d	{'a': (0, 2, 5), 'b': (1, 4), 'c': (3,)} {'x': (0,)} {'cat': (0, 2), 'dog': (1,), 'mouse': (3,)} {'1': (0, 1, 2)}
22	W6A22	Viết hàm đưa vào 1 list số nguyên, tìm và trả về vị trí có giá trị lớn nhất trong list	1 2 3 4 5 -6 7 8 19 34 25 67 89 100 1000 2 45 67 12 90 23 34 8 90 67 24 -45 -78 -29 -78 -199 -199	9 6 5 3 1