

**Câu hỏi trắc nghiệm**

Lưu ý: chỉ chọn một đáp án duy nhất.

**Câu 1.** Cách nào sau đây trả về một *list* chứa 10 số nguyên không âm đầu tiên?

1. **list('0123456789')**
2. **[i for i in range(10)]**
3. **[i for i in range(1, 11)]**
4. **[i for i in range(1, 9)] + [9]**

**Câu 2.** Điều kiện nào có thể dùng để kiểm tra *arr* là một list rỗng (*arr* mang giá trị **[]**)?

1. **arr == []**
2. **arr == False**
3. **arr is None**
4. **arr == None**

**Câu 3.** Cho đoạn code sau:

**arr = [1, 2, 3]**

**arr\_1 = arr[:]**

**arr\_2 = arr.copy()**

**arr[0] = 2021**

Hãy chọn phát biểu đúng về giá trị của *arr\_1* và *arr\_2* sau khi thực thi đoạn code trên.

1. Giá trị của arr\_1 và arr\_2 đều là **[1, 2, 3]**
2. Giá trị của arr\_1 là **[1, 2, 3]** và arr\_2 là **[2021, 2, 3]**
3. Giá trị của arr\_1 là **[2021, 2, 3]** và arr\_2 là **[1, 2, 3]**
4. Giá trị của arr\_1 và arr\_2 đều là **[2021, 2, 3]**

**Câu 4.** Cho list **arr = [4, 6, 2, 7, 8, 1, 4, 8, 1, 6]**. Cách nào sau đây trả về giá trị bé nhất trong *arr*?

1. **max(arr.reverse())**
2. **arr.sort()[0]**
3. **arr.sort(reverse=True)**

**arr[-1]**

1. B và C

**Câu 5.** Trường hợp nào ta buộc phải sử dụng *tuple* để lưu một danh sách các phần tử thay vì *list*?

1. Lưu các phần tử khác kiểu dữ liệu trong cùng một danh sách
2. Tách một danh sách thành hai danh sách có kích thước nhỏ hơn
3. Lưu các phần tử trong một *dictionary*, với mỗi *key* là một số nguyên và mỗi *value* là một danh sách.
4. Lưu các phần tử trong một *set*, với mỗi phần tử là một danh sách.

**Câu 6.** Cho biến **name = 'Mindx'**. Hãy chọn cách hợp lệ để thay đổi giá trị của biến *name* thành **'MindX'**.

1. **name[-1] = 'X'**
2. **name[-1].upper()**
3. **name.find('x') = 'X'**
4. **name = name[:-1] + name[-1].upper()**

**Câu 7.** Đặc tính *immutable* của cấu trúc dữ liệu *string* có hệ quả gì?

1. Ta không thể thay đổi giá trị của một biến với kiểu dữ liệu string.
2. Ta chỉ có thể thay đổi giá trị của một biến string bằng cách tạo một string mới và gán cho biến đó.
3. Ta có thể thay đổi giá trị của một string trong bộ nhớ, nhưng giá trị của biến trỏ đến string đó không thay đổi.
4. Ta chỉ có thể thay đổi giá trị của một string trong bộ nhớ bằng cách sử dụng các hàm cho sẵn trên string như *upper()*, *replace()*.

**Câu 8.** Cấu trúc dữ liệu *set* hiệu quả hơn *list* trong việc:

1. Kiểm tra một phần tử có nằm trong tập hợp hay không
2. Sắp xếp một tập hợp với kích thước lớn
3. Lưu các phần tử thuộc kiểu dữ liệu *string*
4. Lưu một tập hợp chứa nhiều phần tử có cùng giá trị

**Câu 9.** Cấu trúc dữ liệu nào không thể được dùng để làm *key* trong *dictionary*?

1. set
2. list
3. dictionary
4. Tất cả các cấu trúc dữ liệu trên

**Câu 10.** Cho thông tin về quốc gia của các thành phố lớn như sau:

* Hanoi: Vietnam
* Tokyo: Japan
* Moscow: Russia
* ...

Giả sử dữ liệu bao gồm 1000 thành phố và quốc gia. Hãy chọn cách lưu trữ hiệu quả nhất để truy vấn tên quốc gia dựa vào thành phố.

1. **data = [('Hanoi', 'Vietnam'), ('Tokyo', 'Japan'), ...]**
2. **data = {'Hanoi': 'Vietnam', 'Tokyo': 'Japan', ...]**
3. **data\_cities = {'Hanoi', 'Tokyo', ...}**

**data\_counties = {'Vietnam', 'Japan', ...}**

1. **data\_cities = ['Hanoi', 'Tokyo', ...]**

**data\_counties = ['Vietnam', 'Japan', ...]**

# Đáp án

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B | A | A | C | D | D | B | A | D | B |