

NỘI DUNG ÔN TẬP GIỮA HỌC KỲ 2

Năm học 2025-2026

Môn: TIN HỌC - Khối: 10

- Thời gian làm bài: 45 phút
- Trắc nghiệm 70% + 30% tự luận
 - + Trắc nghiệm dạng thức 1 (chọn 1 phương án): 03 điểm - 12 câu
 - + Trắc nghiệm dạng thức 2 (chọn đúng/sai): 04 điểm - 4 câu
 - + Tự luận: 03 điểm

Chủ đề 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

• Bài 20: CÂU LỆNH LẶP FOR

1. Lệnh lặp for:

- Là lệnh lặp với số lần biết trước.
- Số lần lặp được xác định bởi vùng giá trị của lệnh range()
- Lệnh range(n) trả lại vùng giá trị gồm n số từ 0 đến n-1. Vd: range(9) trả lại vùng giá trị từ 0 đến 8
- Cú pháp lệnh for:

for <i> in range(n):
<khởi lệnh>

 - + for, in là từ khóa
 - + i là biến điều khiển vòng lặp, lần lượt nhận các giá trị trong vùng range(n)
 - + khởi lệnh là hoạt động lặp

2. Lệnh range:

- Lệnh tạo vùng giá trị, trả lại vùng giá trị gồm các số nguyên liên tiếp
- Cú pháp: range(start,stop,step)
Trong đó: start là giá trị bắt đầu
 stop là giá trị dừng
 step là bước nhảy (giá trị tăng hoặc giảm giữa mỗi lần lặp)
- Lệnh range có 1 tham số: range(stop)
 - + Giá trị start: mặc định là 0
 - + Bước nhảy step: mặc định là 1
- Lệnh range có 2 tham số: range(start,stop)
 - + Bước nhảy step: mặc định là 1
 - + Nếu giá trị start>stop cho vùng rỗng
- Lệnh range có 3 tham số: range(start,stop,step)
 - + Nếu start<stop thì giá trị step >0 (lặp tiến)
 - + Nếu start>stop thì giá trị step <0 (lặp lùi)

• Bài 21: CÂU LỆNH LẶP WHILE

1. Lệnh lặp while:

- Là lệnh lặp với số lần không biết trước.
- Số lần lặp của lệnh phụ thuộc vào điều kiện của lệnh
- Cú pháp lệnh:

**while <điều kiện>:
<khối lệnh>**

+ while là từ khóa

+ điều kiện là biểu thức logic

+ khối lệnh được thực hiện khi điều kiện đúng, khi điều kiện sai dừng lặp (kết thúc lệnh while)

2. Cấu trúc lập trình:

- Gồm ba cấu trúc lập trình cơ bản của các ngôn ngữ lập trình bậc cao: cấu trúc tuần tự, cấu trúc rẽ nhánh, cấu trúc lặp.
- Cấu trúc tuần tự: Khối gồm các lệnh được thực hiện theo trình tự từ trên xuống dưới. Thể hiện bằng các câu lệnh như gán, nhập/xuất...
- Cấu trúc rẽ nhánh: Khối các câu lệnh chỉ được thực hiện tùy thuộc vào điều kiện nào đó là đúng hay sai (biểu thức logic) Thể hiện bằng câu lệnh rẽ nhánh if
- Cấu trúc lặp: Khối các câu lệnh được thực hiện lặp đi lặp lại tùy theo điều kiện nào đó vẫn còn đúng hay sai. Thể hiện bằng câu lệnh lặp for, while

• Bài 22: KIỂU DỮ LIỆU DANH SÁCH (list)

1. Kiểu dữ liệu danh sách (list):

- Là tập hợp các phần tử có giá trị, các phần tử có các kiểu dữ liệu khác nhau.
- Truy cập các phần tử thông qua chỉ số (vị trí của phần tử trong danh sách).
- Chỉ số của danh sách bắt đầu từ 0 đến len() - 1 (len() là độ dài của danh sách) danh sách rỗng là danh sách có độ dài bằng 0.
- Cú pháp lệnh tạo một danh sách:

<tên list> = [<V1>, <V2>, ..., <Vn>]

Ví dụ: >>>A = [1, 124, 11.85, "chao"]

+ Các phần tử nằm trong cặp dấu []

+ Giữa các phần tử cách nhau bởi dấu “,”

+ Để truy cập đến một phần tử A[i] (A là tên ds, i là chỉ số)

2. Duyệt các phần tử của danh sách:

- Duyệt lần lượt các phần tử của danh sách bằng lệnh for kết hợp với vùng giá trị range()

A=[1,13,5,9]

for i in range(len(A)):

print(A[i], end=" ")

3. Một số lệnh làm việc với danh sách:

- Python có một số lệnh dành riêng (phương thức) cho dữ liệu kiểu danh sách.

Cú pháp: <danh sách>.<phương thức>

- len(A) trả lại giá trị số nguyên là độ dài của danh sách (số lượng các phần tử của danh sách)
- A=[] tạo danh sách rỗng
- [1,2,3] + [4,5,6,7] ghép danh sách (cộng danh sách)
- del A[i] xóa một phần tử của danh sách tại vị trí i
- A[i]=giá trị thay đổi giá trị của phần tử A[i]
- A.append(giá trị) thêm phần tử vào cuối danh sách

• **Bài 23: MỘT SỐ LỆNH LÀM VIỆC VỚI KIỂU DỮ LIỆU DANH SÁCH**

1. Duyệt danh sách với toán tử in:

- Dùng để kiểm tra một phần tử có nằm trong danh sách không. Kết quả trả lại là True hoặc False.
- Có thể duyệt nhanh từ phần tử của danh sách bằng toán tử in và lệnh for mà không cần sử dụng lệnh range().

Ví dụ: >>>A = [1, 124, 13, 9]

>>> 13 in A cho kết quả True

>>> 12 in A cho kết quả False

>>> for k in A:

print(k, end=" ")

cho kết quả: 1 124 13 9

2. Một số lệnh làm việc với danh sách:

- A.clear() xóa toàn bộ danh sách A -> Ds sau xóa là Ds rỗng
- A.remove(x) xóa phần tử có giá trị x tìm thấy đầu tiên của danh sách A; nếu không tìm thấy báo lỗi
- A.insert(k,x) chèn phần tử có giá trị x vào vị trí k của danh sách A

• **Bài 24: XÂU KÝ TỰ (str)**

1. Xâu ký tự (str):

- Là dãy các ký tự Unicode.
 - Có thể coi xâu là danh sách các ký tự nhưng không thay đổi từng ký tự của xâu (khác với danh sách)
 - Truy cập từng ký tự của xâu qua chỉ số
- Ví dụ: S="Chao cac ban"
- Hoặc S=input('Nhap xau: ')

2. Lệnh duyệt ký tự của xâu:

- Duyệt theo chỉ số

```
>>> for i in range(len(s)):
        print(s[i], end=" ")
```
- Duyệt theo ký tự của xâu

```
>>> for i in s:
        print(i, end=" ")
```

• **Bài 25: MỘT SỐ LỆNH LÀM VIỆC VỚI XÂU KÝ TỰ**

1. Xâu con và lệnh tìm vị trí xâu con:

- Để tìm một xâu trong một xâu khác có thể dùng toán tử in hoặc lệnh find().
- Lệnh find() trả về giá trị là vị trí của xâu con trong xâu mẹ
Cú pháp:

- Cú pháp đơn: **<xâu mẹ>.find(<xâu con>)**

Tìm vị trí đầu tiên của xâu con trong xâu mẹ và trả về vị trí đó, nếu không tìm thấy thì trả về -1

- Cú pháp đầy đủ: **<xâu mẹ>.find(<xâu con>,start)**

Tìm vị trí xâu con trong xâu mẹ bắt đầu từ vị trí start

Ví dụ:

```
>>> 'ab' in '123acbcabd'      True
>>> 'cd' in '123acbcabd'      False
>>> '123acbcabd'.find('ac')    Cho giá trị là 3
>>> '123acbcabd'.find('ad')    Cho giá trị là -1
>>> '123acbcabd'.find('ac',5)  Cho giá trị là -1
```

2. Một số lệnh thường dùng với xâu ký tự:

- **<xâu>.split(ký tự dùng để tách)** **tách một xâu thành danh sách**
+ () : mặc định là dấu cách
+ (','): dùng ký tự “,” để phân tách

Ví dụ:

`'Chao cac ban'.split(' ') -> ['Chao','cac','ban']`

`'ky,nguyen,vuon,minh'.split(',') -> ['ky','nguyen','vuon','minh']`

- **“ký tự nối”.join(danh sách)** **nối các phần tử của danh sách thành một xâu**

Ví dụ:

`' '.join(['nguyen','tran','thu','an']) -> 'nguyen tran thu an'`