



BÀI TẬP CHUYÊN ĐỀ

PYTHON CƠ BẢN

BÀI 1: Tổng quan Lập trình Python



Mục tiêu chính:

- Làm quen với môi trường làm việc với Python
- Tạo và thực thi ứng dụng đơn giản

1.1. Xuất hình "HELLO"

- ✓ **Yêu cầu:** Viết chương trình xuất ra shell chữ HELLO như sau:

```

**      **      ***** **      **      *****
**      **      **      **      **      **      **
***** ***** **      **      **      **
**      **      **      **      **      **      **
**      **      ***** ***** ***** *****
    
```

- ✓ **Thuật giải**

- In ra màn hình những dấu "*" và sắp xếp thành chữ HELLO

- ✓ **Hướng dẫn**

- Tạo project có tên là **Python_co_ban**, trong đó tạo package **Bai1**
- Trong package Bai1, tạo module tên là **hello.py**
- Trong module **hello**
 - Sử dụng lệnh `print("Nội dung");` xuất nội dung ra màn hình shell

1.2. Tính toán đơn giản

- ✓ **Yêu cầu:** Viết chương trình tính toán đơn giản, kết quả xuất ra shell như sau

```

x = 10 , y = 5
Tổng x+y = 15
Hiệu x-y = 5
Tích x*y = 50
Thương x/y = 2.0
    
```

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- x, y: với giá trị cho sẵn (y khác 0)

- **Xuất:**

- Kết quả của: `x+y`, `x-y`, `x*y`, `x/y`

- ✓ **Hướng dẫn**

- Trong package Bai1, tạo module tên là **ting_toan_don_gian.py**
- Khai báo và gán giá trị cho x, y, ví dụ như:
 - `x, y = 10, 5`



- Tính và in kết quả:
 - `print('Tổng x+y =', x + y)`
 - Làm tương tự cho `-`, `*`, `/`

BÀI 2: Biến, hằng và các kiểu dữ liệu



Mục tiêu chính: Cung cấp cho HV kiến thức và kỹ năng làm việc với:

- Định danh (Identifier)
- Từ khóa (Keyword)
- Biến (Variable)
- Các kiểu dữ liệu: Integer, Boolean, Float, String

2.1. Tính tiền hàng

- ✓ **Yêu cầu: Xây dựng chương trình tính tiền hàng theo công thức: Thành tiền = Số lượng * Đơn giá**

- Sử dụng shell. Số lượng và đơn giá được nhập vào từ bàn phím. Viết chương trình tính tiền hàng, sau đó hiển thị kết quả.

```
Nhập số lượng:
100
Nhập đơn giá:
5000
Thành tiền = 100 * 5000 = 500000
```

- ✓ **Hướng dẫn sử dụng:**
 - Nhập số lượng và đơn giá. Chương trình sẽ tính và in ra thành tiền

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- Số lượng
- Đơn giá

- **Xuất:**

- Thành tiền

- **Qui tắc xử lý:**

- Thành tiền = Số lượng x Đơn giá

- ✓ **Hướng dẫn**

- Trong project Python_co_ban, tạo package **Bai2**
- Trong package Bai2, tạo module có tên **ting_tien_hang.py**
 - Thực hiện tính toán và in kết quả như hình trên

2.2. Đổi nhiệt độ từ độ C sang độ F

- ✓ **Yêu cầu: Viết chương trình đổi nhiệt độ từ độ C sang độ F**

- Sử dụng shell

- Công thức tính: $\text{Độ F} = 9/5 * \text{Độ C} + 32$

Nhập độ C:

27

27.00 độ C = 80.60 độ F

✓ **Hướng dẫn sử dụng:**

- Nhập vào độ C => chương trình sẽ hiển thị Độ F

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Độ C

▪ **Xuất:**

- Độ F

▪ **Qui tắc xử lý :**

- $\text{Độ F} = 9/5 * \text{Độ C} + 32$

✓ **Hướng dẫn**

- Trong package Bai2, tạo module **doi_nhiet_do.py**
 - Thực hiện tính toán và xuất kết quả như hình trên

2.3. Tính diện tích và chu vi hình tròn

✓ **Yêu cầu: Viết chương trình tính diện tích và chu vi hình tròn**

- Sử dụng shell.
- Bán kính của hình tròn được nhập vào từ bàn phím.
- Viết chương trình tính diện tích và chu vi của hình tròn, sau đó hiển thị kết quả.

Nhập bán kính:

10

$p = 2 * 3.14 * 10 = 62.80$

$s = 3.14 * 10 * 10 = 314.00$

✓ **Hướng dẫn sử dụng:**

- Nhập vào bán kính r => Chương trình sẽ hiển thị diện tích và chu vi hình tròn

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Bán kính

▪ **Xuất:**

- Chu vi
- Diện tích

▪ **Qui tắc xử lý :**

- Chu vi = $2 * \text{Bán kính} * \text{PI}$
- Diện tích = $\text{PI} * \text{Bán kính} * \text{Bán kính}$

✓ **Thuật giải**

- Khai báo biến r nhận giá trị được nhập
- Khai báo biến p (chu vi)
- Khai báo biến s (diện tích)
- Xử lý tính chu vi ($2 * \text{Bán kính} * \text{PI}$)
- Xử lý tính diện tích ($\text{PI} * \text{Bán kính} * \text{Bán kính}$)
- Kết xuất dữ liệu. Kết quả chu vi và diện tích là số thực, để xuất giá trị có 2 số thập phân ta phải định dạng chuỗi xuất ra

✓ **Hướng dẫn**

- Trong package Bai2, tạo module có tên là **tinhh_p_s_hinh_tron.py**
 - Khai báo hằng số $\text{PI} = 3.14$
 - Định dạng chuỗi: `%.2f`

2.4. Xử lý chuỗi

✓ **Yêu cầu: Viết chương xử lý chuỗi**

- Sử dụng shell.
- Chuỗi s1, s2, s3 được nhập vào từ bàn phím
- Chỉ mục index được nhập từ bàn phím
- Cho biết chiều dài của chuỗi s1 và s2, s3
- Tạo chuỗi con s4 từ chuỗi s1 với nội dung từ index đến hết chuỗi
- Lặp lại chuỗi s2: 2 lần

```
Nhập chuỗi s1:
hello
Nhập chuỗi s2:
Python
Nhập chuỗi s3:
programming language
Nhập index:
2
Chiều dài chuỗi s1 = 5
Chiều dài chuỗi s2 = 6
Chiều dài chuỗi s3 = 20
Chuỗi s4 = llo
Chuỗi s2 lặp lại 2 lần = PythonPython
```

✓ **Hướng dẫn sử dụng:**

- Nhập vào chuỗi s1, s2, s3, v. Chương trình sẽ hiển thị kết quả như yêu cầu trên

✓ **Tóm tắt yêu cầu**

- **Nhập:**



- s1, s2, s3
- index

▪ **Xuất:**

- Chiều dài chuỗi s1, s2, s3
- Chuỗi con s4 được tạo ra từ chuỗi s với nội dung từ index đến hết chuỗi
- Lặp lại chuỗi s2: 2 lần

✓ **Hướng dẫn**

- Trong package Bai2, tạo module **xu_ly_chuoi.py**
 - Sử dụng các phương thức về chuỗi đã được hướng dẫn để xử lý chuỗi và xuất ra shell như hình trên

2.5. Tính tiền lãi gửi tiết kiệm

✓ **Yêu cầu: Viết chương trình tính tiền lãi gửi tiết kiệm**

- Sử dụng shell
- Lãi suất một năm, số tiền gửi và số tháng gửi được nhập vào từ bàn phím. Viết chương trình tính tiền lãi và tính tổng số tiền nhận được sau khi hết thời hạn gửi tiền:

Tiền lãi = (Số tiền gửi * Số tháng) * (Lãi suất năm/12)

Tổng số tiền = Số tiền gửi + Tiền lãi

- Sau đó hiển thị kết quả.
- Ví dụ: Lãi suất năm = 7.6, số tiền gửi = 10.000.000 VNĐ, số tháng gửi = 6 tháng
 - Tiền lãi = (10000000 * 6) * (7.6/100/12) = 380000 VNĐ
 - Tiền vốn + lãi = 10000000 + 380000 = 10380000 VNĐ

```
Lãi suất 1 năm (%):
7.6
Số tiền gửi:
10000000
Số tháng gửi:
6
Tiền lãi = 380000.0
Tiền vốn + lãi = 10000000 + 380000 = 10380000
```

✓ **Hướng dẫn sử dụng:**

- Nhập vào lãi suất một năm, số tiền gửi, số tháng gửi => chương trình sẽ hiển thị Tiền lãi và Tiền vốn + lãi.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Lãi suất một năm
- Số tiền gửi
- Số tháng gửi



- **Xuất:**

- Tiền lãi
- Tiền vốn + lãi

- **Qui tắc xử lý :**

- Tiền lãi = (Số tiền gửi * Số tháng) * (Lãi suất năm/12)
- Tiền vốn + lãi = Số tiền gửi + Tiền lãi

- ✓ **Hướng dẫn**

- Trong package Bai2, tạo module **tingh_tien_lai.py**
 - Thực hiện tính toán và xuất kết quả như hình trên

BÀI 3: Toán tử



Mục tiêu chính: Cung cấp cho HV kiến thức và kỹ năng:

- Sử dụng các toán tử trong Python

giá trị biểu thức

✓ Yêu cầu: Xây dựng chương trình tính và in ra tổng của biểu thức S.

- Sử dụng shell.
- Nhập vào một số nguyên x. Tính và in ra $S = 1 + x + x^3/3 + x^5/5$

Nhập x :

6

$S = 1 + x + x^3/3 + x^5/5 = 1634.2$

✓ Hướng dẫn sử dụng:

- Nhập vào x => Chương trình sẽ hiển thị tổng của biểu thức S

✓ Tóm tắt yêu cầu

▪ Nhập:

- x

▪ Xuất:

- S

▪ Quy tắc xử lý :

- $S = 1 + x + x^3/3 + x^5/5$

✓ Hướng dẫn

- Trong project Python_co_ban, tạo package **Bai3**
- Trong package Bai3, tạo module có tên là **tinh_bieu_thuc.py**

3.2. Tính kết quả 2

✓ Yêu cầu: Hãy cho biết kết quả xuất ra của đoạn chương trình sau:

```
result = 1 + 2
print('result =', result)
original_result = result
result = result - 1
print('result =', result)
original_result = result
result = result * 2
original_result = result
print('result =', result)
result = result ** 3
original_result = result
print('result =', result)
```

```
result = result + 8
original_result = result
print('result =', result)
result = result % 7
original_result = result
print('result =', result)
result = result // 2
original_result = result
print('result =', result)
```

Sau khi tính kết quả, hãy viết và chạy đoạn chương trình này để kiểm tra lại.

3.3. Tính kết quả 3

✓ **Yêu cầu:** Hãy cho biết kết quả xuất ra của đoạn chương trình sau:

```
result = 5
print('result =', result)
result -= 1
print('result =', result)
result += 3
print('result =', result)
result = - result
print('result =', result)
result = True
print('not result =', not result)
```

Sau khi tính kết quả, hãy viết và chạy đoạn chương trình này để kiểm tra lại.

3.4. Tính kết quả 4

✓ **Yêu cầu:** Hãy cho biết kết quả xuất ra của đoạn chương trình sau:

```
x = 10
y = 4
print('x = %d, y = %d'%(x,y))
equivalence = x==y
print('x==y is', equivalence)
equivalence = x!=y
print('x!=y is', equivalence)
equivalence = x>y
print('x>y is', equivalence)
x = 8
y = 9
print('x = %d, y = %d'%(x,y))
equivalence = x>=y
print('x>=y is', equivalence)
equivalence = x<y
print('x<y is', equivalence)
equivalence = x<=y
print('x<=y is', equivalence)
```

Sau khi tính kết quả, hãy viết và chạy đoạn chương trình này để kiểm tra lại.

3.5. Tính kết quả 5



✓ **Yêu cầu:** Hãy cho biết kết quả xuất ra của đoạn chương trình sau:

```
x = 15
y = 12
print('Binary of x =', bin(x), ', Binary of y =', bin(y))
print('x & y =', bin(x & y))
print('x | y =', bin(x | y))
print('x ^ y =', bin(x ^ y))
print('~x =', bin(~x))
print('x << 2 =', bin(x << 2))
print('y >> 2 =', bin(y >> 2))
```

Sau khi tính kết quả, hãy viết và chạy đoạn chương trình này để kiểm tra lại.

3.6. Tính kết quả 6

✓ **Yêu cầu:** Hãy cho biết kết quả xuất ra của đoạn chương trình sau:

```
x = True
y = False
print('x and y :', x and y)
print('x or y :', x or y)
print('not x :', not x)
print('x is y :', x is y)
print('x is not y :', x is not y)
```

Sau khi tính kết quả, hãy viết và chạy đoạn chương trình này để kiểm tra lại.

BÀI 4: Cấu trúc điều kiện



Mục tiêu chính: Cung cấp cho học viên kiến thức và kỹ năng sử dụng:

- Cấu trúc if
- Cấu trúc if ... elif ... else

4.1. Tìm số lớn nhất – số nhỏ nhất

✓ **Yêu cầu: Viết chương trình tìm số lớn nhất và số nhỏ nhất**

- Sử dụng shell.
- a, b, c, d là 4 số được nhập vào từ bàn phím. Viết chương trình tìm số lớn nhất và nhỏ nhất của 4 số

```
Nhập a, b, c, d:
24, 12, 3, 11
a, b, c, d = 24 12 3 11
Max = 24
Min = 3
```

✓ **Hướng dẫn sử dụng:**

- Nhập vào bốn số: a, b, c, d => Hiển thị số lớn nhất và số nhỏ nhất trong bốn số

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- a, b, c, d

▪ **Xuất:**

- Số lớn nhất
- Số nhỏ nhất

✓ **Hướng dẫn**

- Trong project Python_co_ban, tạo package **Bai4**
- Trong package Bai4, tạo module có tên là **tim_min_max.py**

4.2. Tìm |x|

✓ **Yêu cầu: Viết chương trình tìm giá trị tuyệt đối của một số**

- Sử dụng shell.
- Nhập vào một số nguyên x. In ra giá trị tuyệt đối của x

```
Nhập x:
-10
|-10| = 10
```

✓ **Hướng dẫn sử dụng:**

- Nhập $x \Rightarrow$ Hiển thị $|x|$

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- x

▪ **Xuất:**

- $|x|$

✓ **Hướng dẫn**

- Trong package Bai4, tạo module có tên là **tim_gttdd_x.py**

4.3. Tính cước taxi

✓ **Yêu cầu: Viết chương trình tính cước taxi theo biểu phí cơ bản như sau:**

- **Dành cho xe 4 chỗ**

Giá mở cửa 11.000 đồng/ 0.8 km
 Trong phạm vi 30km 15.300 đồng/km
 Từ km thứ 31 trở đi 12.100 đồng/km

- **Dành cho xe 7 chỗ**

Giá mở cửa 12.000 đồng/ 0.8 km
 Trong phạm vi 30km 16.100 đồng/km
 Từ km thứ 31 trở đi 13.800 đồng/km

- **Tiền chờ:** 5 phút đầu miễn phí, từ phút thứ sáu trở đi là 750đ/phút

```
Loại xe (chỉ nhập 4 hoặc 7):
4
Số km di chuyển:
32
Thời gian chờ (làm tròn theo phút):
6
Tiền chờ = (6 - 5) * 750đ = 750
Tiền di chuyển = 485160.0
Tiền cước = 485160 + 750 = 485910
```

Ví dụ: Xe 4 chỗ

```
Loại xe (chỉ nhập 4 hoặc 7):
7
Số km di chuyển:
10
Thời gian chờ (làm tròn theo phút):
10
Tiền chờ = (10 - 5) * 750đ = 3750
Tiền di chuyển = 160120.0
Tiền cước = 160120 + 3750 = 163870
```

Ví dụ: Xe 7 chỗ

✓ **Hướng dẫn sử dụng:**

- Nhập vào số km, loại xe (4 hoặc 7), thời gian chờ => Hiển thị tiền chờ, tiền di chuyển, tiền cước

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Loại xe
- Số km
- Thời gian chờ

▪ **Xuất:**

- Tiền chờ
- Tiền di chuyển
- Tiền cước (= tiền chờ + tiền di chuyển)

▪ **Qui tắc xử lý:**

- Học viên tự phân tích quy tắc xử lý dựa theo bảng giá trên

✓ **Hướng dẫn**

- Trong package Bai4, tạo module có tên là **tinhh_cuoc_taxi.py**

4.4. Tính tiền điện

✓ **Yêu cầu: Viết chương trình tính tiền điện dành cho hộ gia đình**

- Tính tiền điện dành cho hộ gia đình phải trả khi biết số Kwh tiêu thụ và cách thức tính theo quy định như sau:

STT	Mức sử dụng của một hộ trong tháng	Giá bán điện (đồng/kWh)
1	Cho 50kWh (cho hộ nghèo và thu nhập thấp)	1.484
2	Cho kWh từ 51 - 100	1.533
3	Cho kWh từ 101 - 200	1.786
4	Cho kWh từ 201 - 300	2.242
5	Cho kWh từ 301 – 400	2.503
6	Cho kWh từ 401 trở lên	2.587

(Biểu giá áp dụng từ ngày 16/03/2015 theo <https://evnspc.vn/>)

Số kw tiêu thụ:

300

Tiền điện phải trả= 553650

✓ **Hướng dẫn sử dụng:**

- Nhập số Kwh tiêu thụ => Hiển thị tổng số tiền điện phải trả.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Số Kwh

▪ **Xuất:**

- Tổng số tiền

▪ **Quy tắc xử lý :**

- Học viên tự phân tích quy tắc xử lý dựa theo bảng giá trên

✓ **Hướng dẫn**

- Trong package Bai4, tạo module có tên là **tingh_tien_dien.py**

4.5. Tính tiền thuê phòng của resort

✓ **Yêu cầu: Viết chương trình tính tiền điện**

- Tính tiền thuê phòng của resort theo bảng giá sau:

Mã loại	Tên loại phòng	Giá 1 đêm	Ở từ 2 - 3 đêm	Ở từ 4 đêm
1	Standard	1,260,000	Giảm 25% so với giá 1 đêm	Giảm 30% so với giá 1 đêm
2	Superior Garden View	1,550,000		
3	Superior Ocean View	1,830,000		
4	Garden View Bungalow	1,830,000		
5	Pool View Bungalow	2,120,000		
6	Family Room	2,120,000		
7	Beach Front Bungalow	2,540,000		
8	VIP sea view	4,800,000		

Nhập loại phòng (từ 1 đến 8):

6

Nhập số đêm:

4

Thành tiền = 5936000.0 vnd

✓ **Hướng dẫn sử dụng:**

- Nhập loại phòng (1 – 8), số đêm ở => Hiển thị tổng số tiền phòng phải trả.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**



- Loại phòng
- Số đêm

- **Xuất:**

- Số tiền

- **Qui tắc xử lý :**

- Học viên tự phân tích quy tắc xử lý dựa theo bảng giá trên

- ✓ **Hướng dẫn**

- Trong package Bai4, tạo module có tên là **tingh_tien_thue_phong.py**

BÀI 5: Cấu trúc lặp



Mục tiêu chính: Cung cấp cho học viên kiến thức và kỹ năng sử dụng:

- Cấu trúc lặp while
- Cấu trúc lặp for

5.1. Count down

✓ Yêu cầu: Xây dựng chương trình Count down.

- Sử dụng shell
- Nhập vào một số nguyên n. Thực hiện việc count down như hình sau:

```
Input number:
10
10
9
8
7
6
5
4
3
2
1
Start!!!
```

✓ Hướng dẫn sử dụng:

- Nhập vào n => Hiển thị các giá trị count down

✓ Tóm tắt yêu cầu

▪ Nhập:

- n

▪ Xuất:

- Count down

✓ Hướng dẫn

- Trong project Python_co_ban, tạo package **Bai5**
- Trong package Bai5, tạo module có tên là **count_down.py**
 - Sử dụng cấu trúc lặp

5.2. Tính S

✓ Yêu cầu: Xây dựng chương trình tính và in ra kết quả của biểu thức.



- Sử dụng shell
- Nhập vào một số nguyên n và một số thực x . Tính và in ra kết quả sau: $S = (x^2 + 1)^n$

Nhập n:

4

Nhập x:

5

$S = (x*x + 1)^n = 456976$

✓ **Hướng dẫn sử dụng:**

- Nhập vào n, x . => Hiển thị giá trị của biểu thức

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- n
- x

▪ **Xuất:**

- S

✓ **Hướng dẫn**

- Trong package Bai5, tạo module có tên là **tiny_S.py**
 - Sử dụng cấu trúc lặp

5.3. Tính A

✓ **Yêu cầu: Xây dựng chương trình tính và in ra kết quả của biểu thức.**

- Sử dụng shell
- Nhập vào một số nguyên n và một số thực x . Tính và in ra kết quả của biểu thức sau: $A = (x^2 + x + 1)^n + (x^2 - x + 1)^n$

Nhập n:

3

Nhập x:

2

$A = (x^2 + x + 1)^n + (x^2 - x + 1)^n = 370$

✓ **Hướng dẫn sử dụng:**

- Nhập vào n, x . => Hiển thị giá trị của biểu thức

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- n
- x

- **Xuất:**

- A

- ✓ **Hướng dẫn**

- Trong package Bai5, tạo module có tên là **tin_h_A.py**
 - Sử dụng cấu trúc lặp

5.4. Kiểm tra số nguyên tố

- ✓ **Yêu cầu: Xây dựng chương trình kiểm tra số nguyên tố**

- Sử dụng shell
- Nhập vào một số x kiểm tra xem x có phải là số nguyên tố hay không. (Số nguyên tố là số chỉ chia hết cho 1 và chính nó)

```
Nhập x:
7
7 là số nguyên tố
```

```
Nhập x:
10
10 không là số nguyên tố
```

- ✓ **Hướng dẫn sử dụng:**

- Nhập vào x => Hiển thị kết quả kiểm tra x

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- x

- **Xuất:**

- Là số NT/ không là số NT

- ✓ **Hướng dẫn**

- Trong package Bai5, tạo module có tên là **kt_so_nguyen_to.py**
 - Sử dụng cấu trúc lặp

5.5. Tính giá trị biểu thức

- ✓ **Yêu cầu: Xây dựng chương trình tính và in ra kết quả của các biểu thức.**

- Sử dụng shell
- Nhập vào một số nguyên n, tính các biểu thức sau đây:
 - A = tổng các số lẻ nhỏ hơn hay bằng n
 - B = tổng các số chẵn nhỏ hơn hay bằng n
 - C = tích các số từ 1 đến n



- D = tích các số chia hết cho 3 nhỏ hơn hay bằng n
- E = tổng các số nguyên tố nhỏ hơn hay bằng n

Nhập n:

4

$$A = 1 + 3 + \dots = 4$$

$$B = 2 + 4 + \dots = 6$$

$$C = 1 * 2 * 3 * 4 * \dots = 24$$

$$D = 3 * \dots = 3$$

$$E = 2 + 3 + \dots = 5$$

✓ **Hướng dẫn sử dụng:**

- Nhập vào n. => Hiển thị giá trị của biểu thức A, B, C, D, E

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- n

▪ **Xuất:**

- A, B, C, D, E

✓ **Hướng dẫn**

- Trong package Bai5, tạo module có tên là **tinht_GTBT.py**
 - Sử dụng vòng lặp

BÀI 6: Numbers – Strings - Datetimes



Mục tiêu chính: Cung cấp cho HV kiến thức và kỹ năng sử dụng:

- Các hàm thư viện Numbers – Strings - Datetimes

6.1. Sử dụng hàm max, min

- ✓ **Yêu cầu:** Viết lại bài tìm giá trị lớn nhất, nhỏ nhất bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong project Python_co_ban, tạo package **Bai6**
 - Trong package Bai6, tạo module có tên là **tim_min_max_2.py**
 - Sử dụng hàm thư viện Numbers

6.2. Sử dụng hàm abs

- ✓ **Yêu cầu:** Viết lại bài tìm $|x|$ bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **tim_gttd_x_2.py**
 - Sử dụng hàm thư viện Numbers

6.3. Sử dụng hàm pow

- ✓ **Yêu cầu:** Viết lại bài tính $S = (x^2 + 1)^n$ bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **tingh_S_2.py**
 - Sử dụng hàm thư viện Numbers

6.4. Sử dụng hàm pow

- ✓ **Yêu cầu:** Viết lại bài tính : $A = (x^2 + x + 1)^n + (x^2 - x + 1)^n$ bằng cách sử dụng hàm thư viện.
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **tingh_A_2.py**
 - Sử dụng hàm thư viện Numbers

6.5. Giải phương trình bậc 2

- ✓ **Yêu cầu:** Xây dựng chương trình giải phương trình bậc 2: $ax^2 + bx + c = 0$
 - Sử dụng shell
 - Nhập vào a, b, c => Giải phương trình và in kết quả



Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

2

Nhập b:

-7

Nhập c:

3

Phương trình bậc 2:

Phương trình có hai nghiệm phân biệt $x_1 = 3.0$, $x_2 = 0.5$

Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

6

Nhập b:

1

Nhập c:

5

Phương trình bậc 2:

Phương trình vô nghiệm

Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

1

Nhập b:

-8

Nhập c:

16

Phương trình bậc 2:

Phương trình có nghiệm kép $x_1 = x_2 = 4.0$

Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Nhập a:

0

Nhập b:

2

Nhập c:

4

Phương trình bậc 2 suy biến thành phương trình bậc 1:

Nghiệm = -2.0

✓ **Hướng dẫn sử dụng:**

- Nhập vào a, b, c => Hiển thị nghiệm của phương trình

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- a, b, c

▪ **Xuất:**

- Nghiệm x_1, x_2

▪ **Qui tắc xử lý :**

- Nếu $a = 0$: giải phương trình bậc I $bx+c = 0$
 - Nếu $b = 0, c \neq 0$: phương trình vô nghiệm
 - Nếu $b = 0, c = 0$: phương trình vô số nghiệm
 - Nếu $b \neq 0$: phương trình có nghiệm $x = -c/b$
- Nếu $a \neq 0$:
 - Tính $\Delta = b^2 - 4ac$
 - Nếu $\Delta < 0$ thì phương trình vô nghiệm.
 - Nếu $\Delta = 0$ thì phương trình (2) có nghiệm kép : $x_1 = x_2 = x_0 = \frac{-b}{2a}$
 - Nếu $\Delta > 0$ thì phương trình (2) có 2 nghiệm phân biệt : $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$

✓ **Hướng dẫn**

- Trong package Bai6, tạo module có tên là **giai_PT_bac_2.py**
 - Áp dụng quy tắc xử lý trên
 - Sử dụng hàm thư viện Numbers

6.6. Sử dụng hàm Strings

✓ **Yêu cầu: Xây dựng chương trình xử lý chuỗi**

- Sử dụng shell
- Nhập vào chuỗi s, chuỗi s_sub, s_find, s_replace.
 - In chuỗi s
 - Loại bỏ khoảng trắng ở đầu và cuối chuỗi
 - In chuỗi với ký tự đầu chuỗi viết hoa
 - Đếm và in ra số lần chuỗi con s_sub xuất hiện trong chuỗi s
 - Tìm kiếm s_find trong s và thay thế bằng s_replace, in chuỗi sau khi tìm kiếm và thay thế

```

Nhập chuỗi s:
a b c d e f duck
Nhập chuỗi con s_sub:
d
Nhập chuỗi tìm s_find:
duck
Nhập chuỗi thay thế s_replace:
dog
Chuỗi s sau khi loại bỏ khoảng trắng đầu và cuối chuỗi: a b c d e f duck
Chuỗi viết hoa ký tự đầu: A b c d e f duck
Số lần s_sub xuất hiện trong s: 2
Chuỗi s sau khi tìm kiếm và thay thế: A b c d e f dog
    
```

✓ **Hướng dẫn sử dụng:**

- Nhập vào s, s_sub, s_find, s_replace => Hiển thị kết quả như hình trên

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- s, s_sub, s_find, s_replace

▪ **Xuất:**

- s

✓ **Hướng dẫn**

- Trong package Bai6, tạo module có tên là **xu_ly_chuoi.py**
 - Sử dụng hàm thư viện Strings

6.7. Sử dụng hàm Datetimes

✓ **Yêu cầu: Xây dựng chương trình xử lý ngày, tháng, năm**

- Sử dụng shell
- Nhập vào ngày, tháng, năm (hợp lệ)
 - Xuất ngày theo định dạng ngày – tháng - năm
 - Cho biết năm được nhập vào có phải là năm nhuận hay không?
 - Cho biết ngày/tháng/năm nhập vào là thứ mấy?
 - Cho biết tháng nhập vào có bao nhiêu ngày?



```

Nhập ngày:
18
Nhập tháng:
2
Nhập năm:
2017
Ngày tháng năm vừa nhập: 18 - 02 - 2017
Năm 2017 không là năm nhuận
18 - 02 - 2017 là Thứ Bảy
Số ngày trong tháng 2 là: 28
    
```

- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào ngày, tháng, năm => Hiển thị kết quả như hình trên
- ✓ **Tóm tắt yêu cầu**
 - **Nhập:**
 - Ngày, tháng, năm
 - **Xuất:**
 - Thông tin theo yêu cầu
- ✓ **Hướng dẫn**
 - Trong package Bai6, tạo module có tên là **xu_ly_thoi_gian.py**
 - Sử dụng hàm thư viện Datetimes

BÀI 7: List – Tuple – Set – Dictionary



Mục tiêu chính: Cung cấp cho học viên kiến thức và kỹ năng sử dụng:

- List
- Tuple
- Set
- Dictionary

7.1. List of animal

✓ Yêu cầu: Tìm thú trong vườn thú

- Sử dụng shell
- Tạo ra một list có các con thú.
- Nhập vào một con thú cần tìm

=> Chương trình in ra danh sách các con thú, số lượng các con thú và kết quả tìm kiếm như hình dưới:

```
List of animals: ['ant', 'bear', 'cat', 'dog', 'elephant', 'fish', 'goat', 'hippo']
Number of animals: 8
I want to find:
bear
There is bear in list of animals
```

▪ Nhập:

- Nhập list, con thú cần tìm

▪ Xuất:

- List
- Số lượng thú
- Kết quả tìm kiếm

✓ Hướng dẫn

- Trong project Python_co_ban, tạo package **Bai7**
- Trong package Bai7, tạo module có tên là **list_of_animals.py**

7.2. List numbers 1

✓ Yêu cầu: Viết chương trình xử lý list như sau

- Tạo list
- Cho phép người dùng lần lượt nhập các phần tử số cho list cho đến khi không muốn nhập nữa
- Nhập vào một số x
=> Chương trình sẽ trả lời những câu hỏi sau:
 - Tính tổng các phần tử trong list
 - x có xuất hiện trong list hay không không? Nếu có thì cho biết x xuất hiện bao nhiêu lần?

- x có lớn hơn tất cả các số trong list không?
- Nếu không thì x nhỏ hơn những số nào trong list? (In ra tất cả các số lớn hơn x)

```

Nhập giá trị: 10
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 5
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: -2
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 23
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 5
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 6
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 7
Tiếp tục nhập giá trị? 1: Có, 0: không 0
Nhập giá trị cần tìm x: 5
List: [10, 5, -2, 23, 5, 6, 7]
Tổng các giá trị trong list: 54
5 xuất hiện 2 lần trong list
5 không lớn hơn tất cả các số trong list
Các số lớn hơn 5 trong list: [10, 23, 6, 7]
    
```

✓ Tóm tắt yêu cầu

▪ Nhập:

- Giá trị của các phần tử trong list
- x

▪ Xuất:

- List
- Tổng list
- Kết quả tìm x trong list
- Các số lớn hơn x trong list

✓ Hướng dẫn

- Trong package Bai7, tạo module có tên là **list_numbers_1.py**

7.3. List numbers 2

✓ Yêu cầu: Viết chương trình thực xử lý list như sau

- Tạo list
- Nhập số phần tử trong list
- Cho phép người dùng lần lượt nhập các phần tử cho list cho đến khi không muốn nhập nữa
=> Chương trình sẽ thực hiện những công việc sau:
 - Tìm và in ra tất cả các số nguyên tố có trong list
 - Tính trung bình cộng của các phần tử âm/ phần tử dương trong list
 - Tìm giá trị lớn nhất/ nhỏ nhất trong list

- Sắp xếp list theo giá trị tăng dần

```
Nhập giá trị: 1
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: -3
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 2
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 8
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: -4
Tiếp tục nhập giá trị? 1: Có, 0: không 1
Nhập giá trị: 7
Tiếp tục nhập giá trị? 1: Có, 0: không 0
List: [1, -3, 2, 8, -4, 7]
Các số nguyên tố trong list: [2, 7]
Các phần tử âm trong list: [-3, -4]
Trung bình cộng các phần tử âm: -3.50
Các phần tử dương trong list: [1, 2, 8, 7]
Trung bình cộng các phần tử dương: 4.50
Giá trị max trong list 8
Giá trị min trong list -4
List sắp tăng dần: [-4, -3, 1, 2, 7, 8]
```

✓ Tóm tắt yêu cầu

▪ Nhập:

- Giá trị của các phần tử trong list

▪ Xuất:

- List
- Các số nguyên tố trong list
- Trung bình cộng âm/ dương trong list
- Max/ min trong list
- List tăng dần

✓ Hướng dẫn

- Trong package Bai7, tạo module có tên là **list_number_2.py**

7.4. Tuple strings

✓ Yêu cầu: Viết chương trình thực hiện việc xử lý tuple như sau:

- Tạo 1 tuple có 10 phần tử chuỗi bất kỳ.
- Nhập index dương ($0 \leq \text{index} < 10$), index âm ($-1 \geq \text{index} \geq -9$)
- Nhập chuỗi cần tìm s_find
=> Chương trình sẽ thực hiện những công việc sau:



- In tuple
- In giá trị của phần tử trong tuple có index dương và index âm đã nhập
- Tìm và đếm số lần xuất hiện của s_find trong tuple

```
Tuple: ('red', 'green', 'yellow', 'blue', 'black', 'white', 'pink', 'orange', 'red', 'blue')
Nhập số từ 0 đến 9: 2
Nhập số từ -1 đến -9: -2
Nhập chuỗi cần tìm:
blue
Tuple[ 2 ]= yellow
Tuple[ -2 ]= red
blue xuất hiện trong tuple 2 lần
```

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Tuple, index dương, index âm
- s_find

▪ **Xuất:**

- Theo yêu cầu trên

✓ **Hướng dẫn**

- Trong package Bai7, tạo module có tên là **tuple_strings.py**

7.5. Tuple numbers

✓ **Yêu cầu: Viết chương trình thực hiện việc xử lý trên tuple như sau:**

- Tạo 1 tuple a chứa 4 số nguyên dương đầu tiên
- Tạo 1 tuple b chứa 4 số nguyên dương tiếp theo
- Tạo 1 tuple c là sự kết hợp của các phần tử trong tuple a và b
- Tạo 1 tuple d từ tuple c với các phần tử được sắp xếp
- In phần tử thứ 3 của d
- In 3 phần tử cuối cùng của d

```

Tuple a: (3, 1, 2, 4)
Tuple b: (5, 7, 6, 8)
Tuple c: (3, 1, 2, 4, 5, 7, 6, 8)
Tuple d: (1, 2, 3, 4, 5, 6, 7, 8)
Tuple[3]= 4
3 phần tử cuối cùng của tuple d (6, 7, 8)
```

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Tuple a, b

▪ **Xuất:**

- Tuple c, d

- Phần tử thứ 3 của d
- 3 phần tử cuối cùng của d

✓ **Hướng dẫn**

- Trong package Bai7, tạo module có tên là **tuple_numbers.py**

7.6. Set numbers

✓ **Yêu cầu: Viết chương trình thực hiện việc xử lý trên set như sau:**

- Khai báo và khởi tạo set1, set2
 - Cho phép người dùng lần lượt nhập các phần tử số cho set1 cho đến khi không muốn nhập nữa
 - Cho phép người dùng lần lượt nhập các phần tử số cho set2 cho đến khi không muốn nhập nữa
- => Chương trình sẽ thực hiện những công việc sau::
- In set1, set2
 - Cho biết mỗi set có bao nhiêu phần tử, tổng giá trị các phần tử của mỗi set
 - Tìm giá trị lớn nhất, nhỏ nhất của mỗi set
 - Lấy ra một phần tử ở set1 và in ra phần tử này
 - Thực hiện set union của set1 và set2 và in kết quả
 - Thực hiện set intersection của set1 và set2 và in kết quả
 - Thực hiện set difference của set1 với set2 và in kết quả
 - Thực hiện set symmetric difference của set1 với set2 và in kết quả
 - Sắp xếp set1 tăng dần và set2 giảm dần

```

Nhập giá trị cho element trong set 1: 5
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không 1
Nhập giá trị cho element trong set 1: 2
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không 1
Nhập giá trị cho element trong set 1: 7
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không 1
Nhập giá trị cho element trong set 1: 4
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không 0
Nhập giá trị cho element trong set 2: 7
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không 1
Nhập giá trị cho element trong set 2: 8
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không 1
Nhập giá trị cho element trong set 2: 10
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không 1
Nhập giá trị cho element trong set 2: 6
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không 0
Set 1: {2, 4, 5, 7}
Set 2: {8, 10, 6, 7}
Chiều dài Set 1: 4
Chiều dài Set 2: 4
Tổng Set 1: 18
Tổng Set 2: 31
Max Set 1, Min Set 1: 7 , 2
Max Set 2, Min set 2: 10 , 6
Pop Set 1: 2
Set 1 sau khi pop: {4, 5, 7}
Set 1 union Set 2: {4, 5, 6, 7, 8, 10}
Set 1 intersection Set 2: {7}
Set 1 difference Set 2: {4, 5}
Set 1, Set 2 symmetric differnce: {4, 5, 6, 8, 10}
Set 1 tăng dần: [4, 5, 7]
Set 2 giảm dần: [10, 8, 7, 6]
    
```

✓ Tóm tắt yêu cầu

- **Nhập:**
 - set1, set2
- **Xuất:**
 - Theo các yêu cầu liệt kê ở trên

✓ Hướng dẫn

- Trong package Bai7, tạo module có tên là **set_numbers.py**

7.7. Danh bạ điện thoại

- ✓ **Yêu cầu:** Viết chương trình thực hiện việc xử lý danh bạ điện thoại như sau:

- Tạo một danh bạ kiểu dictionary để lưu trữ danh bạ điện thoại với các cặp key-value, ví dụ như:

Name	Telephone number
Johnny	0989741258
Katherine	0903852147
Misu	0913753951
Jack	0933753654
...	...

- Nhập tên cần tìm search_name
- Nhập tên, số điện thoại
=> Chương trình sẽ thực hiện những công việc sau:
 - Tìm search_name trong danh bạ. Nếu không tìm thấy thì in thông tin tên – số điện thoại. Nếu không tìm thấy thì thông báo là không tìm thấy.
 - Thêm một liên hệ mới với thông tin: tên – số điện thoại đã nhập
 - In danh bạ



```

Bạn muốn làm gì? 1: Xem danh bạ; 2: Tìm kiếm, 3: Thêm mới      1
Danh bạ điện thoại:
Tên      Số điện thoại
Jack --  0933753654
Misu --  0913753951
Johnny -- 0989741258
Katherine -- 0903852147
Tiếp tục lựa chọn? 1: Có; 0: Không      1
Bạn muốn làm gì? 1: Xem danh bạ; 2: Tìm kiếm, 3: Thêm mới      2
Nhập tên cần tìm:
Misu
Misu có số điện thoại là: 0913753951
Tiếp tục lựa chọn? 1: Có; 0: Không      1
Bạn muốn làm gì? 1: Xem danh bạ; 2: Tìm kiếm, 3: Thêm mới      3
Nhập tên:
Louisa
Nhập số điện thoại:
0913852258
Danh bạ điện thoại:
Tên      Số điện thoại
Jack --  0933753654
Misu --  0913753951
Johnny -- 0989741258
Louisa -- 0913852258
Katherine -- 0903852147
Tiếp tục lựa chọn? 1: Có; 0: Không      0
    
```

✓ Tóm tắt yêu cầu

▪ Nhập:

- Danh bạ
- search_name
- Liên hệ mới

▪ Xuất:

- Kết quả tìm kiếm
- Danh bạ

✓ Hướng dẫn

- Trong package Bai7, tạo module có tên là **danh_ba.py**

7.8. Từ điển

✓ Yêu cầu: Viết chương trình thực hiện việc xử lý từ điển Anh – Việt như sau:

- Tạo một từ điển
=> Chương trình sẽ thực hiện những công việc sau:

- Thêm từ vào từ điển (key: từ tiếng Anh, value: nghĩa tiếng Việt)
- Hiển thị từ điển, cho biết trong từ điển hiện tại có bao nhiêu từ
- Tìm kiếm từ tiếng Anh => nếu tìm thấy thì hiển thị key và value. Nếu không tìm thấy thì thông báo không tìm thấy
- Xóa một từ trong từ điển, dựa trên key cung cấp

Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ 1

Dictionary:

Từ Anh Nghĩa Việt

cat con mèo

dog con chó

ant con kiến

bear con gấu

Tiếp tục lựa chọn? 1: Có; 0: Không 1

Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ 2

Nhập từ cần tra: book

Không tìm thấy từ trong từ điển

Tiếp tục lựa chọn? 1: Có; 0: Không 1

Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ 3

Nhập từ Anh: book

Nhập nghĩa Việt: quyển sách

Dictionary:

Từ Anh Nghĩa Việt

cat con mèo

dog con chó

book quyển sách

ant con kiến

bear con gấu

Tiếp tục lựa chọn? 1: Có; 0: Không 1

Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ 4

Nhập từ cần xóa: ant

Bạn có thật sự muốn xóa hay không? 1: Xóa, 0: Không 1

Đã xóa từ trong từ điển

Dictionary:

Từ Anh Nghĩa Việt

cat con mèo

dog con chó

book quyển sách

bear con gấu

Tiếp tục lựa chọn? 1: Có; 0: Không 0

✓ Tóm tắt yêu cầu

▪ Nhập:

- Từ cần tìm/ Từ cần thêm/ Từ cần xóa

▪ Xuất:

- Kết quả như hình trên

✓ Hướng dẫn

- Trong package Bai7, tạo module có tên là tu_dien.py

BÀI 8: Phương thức/hàm



Mục tiêu chính: Cung cấp cho học viên kiến thức và kỹ năng:

- Xây dựng và gọi sử dụng phương thức/hàm

8.1. Tính năm âm lịch

- ✓ **Yêu cầu: Viết chương trình tính năm âm lịch từ năm dương lịch.**

Nhập năm:

2017

Năm 2017 âm lịch là năm Đinh Dậu

- ✓ **Hướng dẫn sử dụng:**

- Khi người dùng nhập năm sinh => Hiển thị năm âm lịch.

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- Năm sinh

- **Xuất:**

- Năm âm lịch

- **Qui tắc xử lý :**

- Can là lấy năm chia lấy dư cho 10

0	1	2	3	4	5	6	7	8	9
Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

- Chi là lấy năm chia lấy dư cho 12

0	1	2	3	4	5	6	7	8	9	10	11
Thân	Dậu	Tuất	Hợi	Tý	Sửu	Dần	Mão	Thìn	Tỵ	Ngọ	mùi

- ✓ **Hướng dẫn**

- Trong project Python_co_ban, tạo package **Bai8**

- Trong package Bai8, tạo module có tên là **nam_am_lich.py**

- Viết phương thức **tingh_can(nam)** có kết quả trả về là **chuỗi can**, tingh_chi(nam) có kết quả trả về là **chuỗi chi**
- Sử dụng cấu trúc if...elif...else để giải quyết bài toán theo quy tắc xử lý trên.

8.2. Tính chỉ số BMI



✓ **Yêu cầu: Viết chương trình tính chỉ số BMI**

– Cách tính BMI:

- BMI = Cân nặng / (Chiều cao * Chiều cao)
- Bảng đánh giá BMI:
 - Gầy: <18.5
 - Bình thường: 18.5 – 24.99
 - Thừa cân: >=25

```
Nhập cân nặng (kg):
52
Nhập chiều cao (m):
1.6
Chỉ số BMI của bạn: 20.31
Kết quả: Bạn bình thường
```

✓ **Hướng dẫn sử dụng:**

- Khi người dùng nhập chiều cao, cân nặng => hiển thị chỉ số BMI kèm theo đánh giá.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Chiều cao (m)
- Cân nặng (kg)

▪ **Xuất:**

- Chỉ số BMI của cơ thể
- Kết luận

✓ **Hướng dẫn**

- Trong package Bai8, tạo module có tên là **tingh_bmi.py**
 - Xây dựng phương thức tính **tingh_bmi(can_nang, chieu_cao)** có giá trị trả về là BMI như quy tắc xử lý trên
 - Xây dựng phương thức **danh_gia_bmi(bmi)** có giá trị trả về là lời đánh giá theo quy tắc xử lý trên

8.3. Phương thức - hàm Bài 5

✓ **Yêu cầu: Hãy viết lại các bài tập 5.1, 5.2, 5.3 bằng cách xây dựng phương thức/ hàm**

✓ **Hướng dẫn**

- Bài 5.1: Xây dựng phương thức **tingh_S(n, x)**: với n và x là tham số truyền vào, phương thức có giá trị trả về là $S = (x^2 + 1)^n$
- Bài 5.2: Xây dựng phương thức **tingh_A(n, x)**: với n và x là tham số truyền vào, phương thức có giá trị trả về là $A = (x^2 + x + 1)^n + (x^2 - x + 1)^n$



- Bài 5.3: Xây dựng phương thức **kiem_tra_so_nguyen_to(x)**: x là tham số truyền vào, phương thức có giá trị trả về là True nếu x là số nguyên tố, có giá trị trả về là False nếu x không là số nguyên tố

8.4. Phương thức - hàm bài 7

- ✓ **Yêu cầu: Hãy viết lại các bài 7.2, 7.4, 7.6 trong bài 7 bằng cách xây dựng phương thức/hàm**
- ✓ **Hướng dẫn**
 - Bài 7.2:
 - Xây dựng phương thức **them_vao_list(list_original)**: dùng để nhập thêm các phần tử vào list. Kết quả trả về là list_original sau khi đã thêm các phần tử
 - Xây dựng phương thức **ting_tong_list(list_original)**: dùng để tính tổng các phần tử trong list. Kết quả trả về là tổng của list
 - Bài 7.4:
 - Xây dựng phương thức **tim_dem_slxh(tuple_original, x)**: dùng để tìm và đếm số lần xuất hiện của x trong tuple. Kết quả trả về là số lần xuất hiện (0: không xuất hiện)
 - Bài 7.6:
 - Xây dựng phương thức **in_dictionary(dictionary)**: dùng để in dictionary theo định dạng mỗi item (key : value) hiển thị trên một dòng.
 - Xây dựng phương thức **tim_kiem_dictionary(dictionary, key_search)**: dùng để tìm key_search trong từ điển. Kết quả trả về là chuỗi key : value nếu tìm thấy, 'Không tìm thấy keysearch' nếu không tìm thấy
 - Xây dựng phương thức **tim_kiem_dictionary(dictionary, key_insert, value_insert)**: dùng để thêm key : value mới vào dictionary. Kết quả trả về là dictionary sau khi đã thêm

8.5. Lambda

- ✓ **Yêu cầu: hãy sử dụng biểu thức Lambda để tính**
 - Diện tích, chu vi hình tròn với tham số là r (bán kính)
 - Diện tích và chu vi hình chữ nhật với tham số là a, b (chiều dài và chiều rộng)
- ✓ **Hướng dẫn sử dụng:**
 - Khi người dùng nhập r, a, b => hiển thị diện tích và chu vi hình tròn, hình chữ nhật
- ✓ **Tóm tắt yêu cầu**
 - **Nhập:**
 - r, a, b
 - **Xuất:**
 - S, P hình tròn
 - S, P hình chữ nhật
- ✓ **Hướng dẫn**

- Trong package Bai8, tạo module có tên là **su_dung_lambda.py**
 - Tính diện tích hình tròn :
`s_tron = lambda r: math.pi * math.pow(r, 2)`
 - Làm tương tự cho các yêu cầu còn lại
 - Gọi sử dụng các lambda đã viết

8.6. Map, filter, reduce

- ✓ **Yêu cầu: Hãy áp dụng các built-in function map(), reduce(), filter() cho những bài tập list, tuple của bài List – Tuple - Dictionary**

- Tính tổng các phần tử trong list
- List các số lớn hơn x
- List các số nguyên tố
- List các phần tử âm
- List các phần tử dương
- ...

BÀI 9: Module



Mục tiêu chính: Giúp cho học viên

- *Hiểu được khái niệm về module*
- *Biết cách Cách import module vào module*
- *Biết Cách tổ chức package*

9.1. Module 1

✓ Yêu cầu: Tổ chức và sử dụng module

- Hãy đưa các phương thức hàm trong bài 5 vào một module có tên là **ham_bai_5.py**.
- Import module **ham_bai_5** vào các bài tập của bài 5.

✓ Hướng dẫn

- Trong project Python_co_ban, tạo package **Bai9**
- Trong package Bai9
 - Tạo module có tên là **ham_bai_5.py**
- Import module **ham_bai_5** vào các bài tập của bài 5 => gọi hàm => xem kết quả

9.2. Module 2

✓ Yêu cầu: Tổ chức và sử dụng module

- Hãy đưa các phương thức hàm dành cho List trong bài 7 vào một module có tên là **ham_list_7.py**.
- Hãy đưa các phương thức hàm dành cho Tuple trong bài 7 vào một module có tên là **ham_tuple_7.py**.
- Hãy đưa các phương thức hàm dành cho Dictionary trong bài 7 vào một module có tên là **ham_dictionary_7.py**.
- Hãy đưa các phương thức hàm dành cho Dictionary trong bài 7 vào một module có tên là **ham_set_7.py**.
- Gọi sử dụng các phương thức này vào các bài tập của bài 7

✓ Hướng dẫn

- Tạo package Bai9,
- Trong Bai9
 - Tạo module có tên là **ham_list_7.py**.
 - Tạo module có tên là **ham_tuple_7.py**.
 - Tạo module có tên là **ham_dictionary_7.py**.
 - Tạo module có tên là **ham_set_7.py**.
- Import module là **ham_list_7, ham_tuple_7, ham_dictionary_7, ham_set_7** vào các bài tập của bài 7 => gọi hàm => xem kết quả

BÀI 10: Xử lý ngoại lệ



Mục tiêu chính:

- Xử lý ngoại lệ trong việc xây dựng các xử lý

Giá trị biểu thức

- ✓ **Yêu cầu:** Xây dựng chương trình tính và in ra kết quả của biểu thức.

$$A = \sqrt{\frac{5x - y}{2x + 7y}}$$

- Bổ sung ngoại lệ cho phần xử lý

```
Nhập x:
2
Nhập y:
7
A = 0.23791547571544325

Nhập x:
7
Nhập y:
-2
Error: divisor must be <>0
A = None
```

- ✓ **Hướng dẫn sử dụng:**

- Nhập vào x,y. => Hiển thị giá trị của biểu thức A

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- x, y

- **Xuất:**

- A

- ✓ **Hướng dẫn**

- Trong project Python_co_ban, tạo package **Bai10**
- Trong package Bai10, tạo module có tên là **tinh_toan_GTBT.py**

- Xây dựng phương thức **def tinhGTBT(x, y)** với x,y là tham số truyền vào. Phương thức có giá trị trả về là giá trị của biểu thức.

(Bổ sung ngoại lệ cho phần xử lý)

10.2. Xử lý ngoại lệ cho các bài tập

- ✓ **Yêu cầu: Bổ sung xử lý ngoại lệ (nếu cần) cho xử lý của các bài tập trong bài "Cấu trúc điều kiện", "Cấu trúc lặp", "List – tuple – dictionary", "Phương thức"**

- Ví dụ: Thêm try ... except... cho bài tính |x|

```
try:
    x = int(input('Nhập x:\n'))
    abs_x = x
    if abs_x < 0:
        abs_x = - abs_x
        print('|%d| = %d'%(x, abs_x))
except ValueError as err:
    print('Error:', err)
```

BÀI 11: Files I/O



Mục tiêu chính:

- Mở file, đọc file, đóng file text/ csv
- Mở file, ghi file, đóng file text/csv
- Thao tác trên tập tin
- Thao tác trên thư mục

11.1. Mở - đọc - đóng file

- ✓ **Yêu cầu: Viết chương trình mở và đọc nội dung tập tin như sau:**

```

Nhập tên tập tin:
HumptyDumpty.txt
Nội dung tập tin:
Humpty Dumpty sat on a wall,
Humpty Dumpty had a great fall.
All the king's horses and all the king's men
Couldn't put Humpty together again.
    
```

- ✓ **Hướng dẫn sử dụng:**

- Nhập tên tập tin cần đọc => đọc và in ra nội dung như trên.

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- Tên tập tin

- **Xuất:**

- Nội dung tập tin

- ✓ **Hướng dẫn**

- Trong project Python_co_ban, tạo package **Bai11**
- Trong package Bai11, tạo một module có tên là **xu_ly_tap_tin.py**
 - Trong xu_ly_tap_tin.py, xây dựng phương thức **read_file(filename)** trả về nội dung tập tin
- Trong package Bai11, tạo một module có tên là **mo_doc_tap_tin.py**
 - Cho người dùng nhập tên tập tin
 - Gọi phương thức read_file(filename) để xem kết quả

11.2. Mở - đọc – thống kê - đóng file

- ✓ **Yêu cầu: Viết chương trình mở - đọc – thống kê nội dung tập tin như sau:**



```

Nhập tên tập tin:
HumptyDumpty.txt
Content of file:
Humpty Dumpty sat on a wall,
Humpty Dumpty had a great fall.
All the king's horses and all the king's men
Couldn't put Humpty together again.
-----Report: Lines/ Words/ Chars-----
Lines: 4 , Words: 26 , Chars: 141
    
```

✓ **Hướng dẫn sử dụng:**

- Nhập tên tập tin => Hiển thị nội dung tập tin, số dòng, số từ, số ký tự trong tập tin.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Tên tập tin

▪ **Xuất:**

- Nội dung tập tin
- Số dòng, số từ, số ký tự

✓ **Hướng dẫn**

- Trong **xu_ly_tap_tin.py**, xây dựng phương thức **read_report_file(filename)**: in kết quả là nội dung tập tin, thống kê số dòng, số từ, số ký tự
- Trong package Bai11, tạo một module có tên là **mo_doc_thong_ke_tap_tin.py**
 - Cho người dùng nhập tên tập tin
 - Gọi phương thức **read_report_file(filename)** để xem kết quả

11.3. Mở - đọc – ghi file

✓ **Yêu cầu: Viết chương trình ghi nội dung vào tập tin như sau:**

- Nhập tên tập tin và nội dung
- Nếu chưa tồn tại tập tin: tạo tập tin và ghi nội dung vào
- Nếu đã tồn tại tập tin: xóa nội dung cũ, ghi nội dung mới

```

Nhập tên tập tin:
Rain.txt
Nhập nội dung:
Rain rain, go away; Come again another day...
Đã ghi nội dung vào tập tin Rain.txt
Rain rain, go away; Come again another day...
    
```

✓ **Hướng dẫn sử dụng:**



- Nhập tên tập tin và nội dung => ghi nội dung vào tập tin. Hiển thị nội dung sau khi ghi.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Tên tập tin
- Nội dung

▪ **Xuất:**

- Nội dung tập tin sau khi ghi

✓ **Hướng dẫn**

- Trong **xu_ly_tap_tin.py**, xây dựng phương thức **write_file(filename, content)**: thực hiện việc mở tập tin filename, ghi content, đóng tập tin
- Trong package Bai11, tạo một module có tên là **mo_ghi_tap_tin.py**
 - Cho người dùng nhập tên tập tin, nội dung
 - Gọi phương thức write_file(filename, content)
 - Gọi phương thức read_file(filename) để xem kết quả

11.4. Mở - đọc – ghi vào cuối file

✓ **Yêu cầu: Viết chương trình ghi nội dung vào tập tin như sau:**

- Nhập tên tập tin và nội dung
- Nếu chưa tồn tại tập tin: tạo tập tin và ghi nội dung vào
- Nếu đã tồn tại tập tin: ghi nội dung vào cuối tập tin, xuống dòng
- Hỏi người dùng có muốn tiếp tục ghi nữa hay không? Người dùng chọn 1: có, chọn 0: không
 - Nếu chọn 1: yêu cầu người dùng nhập nội dung => ghi nội dung vào cuối tập tin, xuống dòng
 - Nếu chọn 0: kết thúc chương trình

```

Nhập tên tập tin:
JohnnyJohnny.txt
Nhập nội dung:
Johnny Johnny
Bạn có muốn tiếp tục ghi nội dung vào file? 1: Có; 0: Không
1
Nhập nội dung:
Yes Papa
Bạn có muốn tiếp tục ghi nội dung vào file? 1: Có; 0: Không
1
Nhập nội dung:
Eating sugar
Bạn có muốn tiếp tục ghi nội dung vào file? 1: Có; 0: Không
1
Nhập nội dung:
No, Papa
Bạn có muốn tiếp tục ghi nội dung vào file? 1: Có; 0: Không
0
Đã ghi nội dung vào tập tin JohnnyJohnny.txt
Johnny Johnny
Yes Papa
Eating sugar
No, Papa
    
```

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Tên tập tin
- Lần lượt nhập các nội dung

▪ **Xuất:**

- Nội dung tập tin sau khi ghi

✓ **Hướng dẫn**

- Trong **xu_ly_tap_tin.py**, xây dựng phương thức **write_end_of_file(filename)**: thực hiện việc mở tập tin filename, liên tục ghi content vào cuối tập tin, đóng tập tin
- Trong package Bai11, tạo một module có tên là **mo_ghi_cuoi_tap_tin.py**
 - Cho người dùng nhập tên tập tin
 - Gọi phương thức write_end_of_file(filename)
 - Gọi phương thức read_file(filename) để xem kết quả

11.5. Mở - đọc - đóng file CSV

✓ **Yêu cầu: Viết chương trình mở và đọc nội dung tập tin CSV như sau:**

Nhập tên tập tin:

`menu.csv`

Nội dung tập tin:

```
[ 'Monday', 'ham', 'biscuits', 'corn', 'spinach', 'apple pie' ]
[ 'Tuesday', 'steak', 'rolls', 'yams', 'beets', 'crème brulee' ]
[ 'Wednesday', 'fried chicken', 'biscuits', 'mashed potatoes', 'cole slaw', 'vanilla ice cream' ]
[ 'Thursday', 'clam rolls', 'tartar sauce', 'french fries', 'ice tea', 'chocolate ice cream' ]
[ 'Friday', 'meat loaf', 'biscuits', 'lima beans', 'garden salad', 'chocolate cake' ]
None
```

✓ **Hướng dẫn sử dụng:**

- Nhập tên tập tin csv cần đọc => đọc và in ra nội dung như trên.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Tên tập tin csv

▪ **Xuất:**

- Nội dung tập tin

✓ **Hướng dẫn**

- Tạo package Bai11
- Trong package Bai11, tạo một module có tên là **xu_ly_tap_tin_csv.py**
 - Trong xu_ly_tap_tin_csv.py, xây dựng phương thức **read_file_csv(filename)** trả về nội dung tập tin.
- Trong package Bai11, tạo một module có tên là **mo_doc_tap_tin_csv.py**
 - Cho người dùng nhập tên tập tin
 - Gọi phương thức read_file_csv(filename) để xem kết quả

11.6. Mở - đọc – ghi danh sách vào cuối file csv

✓ **Yêu cầu: Viết chương trình ghi danh sách số điện thoại vào cuối file csv như sau:**

- Nhập tên tập tin csv (ví dụ: danhba.csv), danh sách số điện thoại (Tên – Số điện thoại)
- Ghi danh sách số điện thoại vào tập tin

Nhập tên tập tin:

`dienthoai.csv`

```
name      fone
Jonny      0989 753951
Lucy       0903 123456
Jack       0913 753951
Johnny Lee  0913 753852
Peter Son  0989 753951
Johnnathan 0903 123456
```

✓ **Tóm tắt yêu cầu**



- **Nhập:**

- Tên tập tin csv
- Danh sách số điện thoại

- **Xuất:**

- Nội dung tập tin sau khi ghi

✓ **Hướng dẫn**

- Trong **xu_ly_tap_tin_csv.py**, xây dựng phương thức **write_csv_file(filename, listContent)**: thực hiện việc mở tập tin filename, ghi listContent vào tập tin, đóng tập tin
- Trong package Bai11, tạo một module có tên là **mo_ghi_tap_tin_csv.py**
 - Cho người dùng nhập tên tập tin, danh bạ điện thoại
 - Gọi phương thức write_csv_file(filename, listContent)
 - Gọi phương thức read_csv_file(filename) để xem kết quả



Mục lục

1.1. Xuất hình "HELLO"	2
1.2. Tính toán đơn giản	2
BÀI 2: Biến, hằng và các kiểu dữ liệu.....	4
2.1. Tính tiền hàng	4
2.2. Đổi nhiệt độ từ độ C sang độ F	4
2.3. Tính diện tích và chu vi hình tròn	5
2.4. Xử lý chuỗi	6
2.5. Tính tiền lãi gửi tiết kiệm	7
BÀI 3: Toán tử	9
3.1. Tính giá trị biểu thức	9
3.2. Tính kết quả 2	9
3.3. Tính kết quả 3	10
3.4. Tính kết quả 4	10
3.5. Tính kết quả 5	10
3.6. Tính kết quả 6	11
BÀI 4: Cấu trúc điều kiện	12
4.1. Tìm số lớn nhất – số nhỏ nhất	12
4.2. Tìm $ x $	12
4.3. Tính cước taxi	13
4.4. Tính tiền điện	14
4.5. Tính tiền thuê phòng của resort	15
BÀI 5: Cấu trúc lặp	17
5.1. Count down.....	17
5.2. Tính S	17



5.3. Tính A	18
5.4. Kiểm tra số nguyên tố.....	19
5.5. Tính giá trị biểu thức	19
BÀI 6: Numbers – Strings - Datetimes.....	21
6.1. Sử dụng hàm max, min.....	21
6.2. Sử dụng hàm abs	21
6.3. Sử dụng hàm pow	21
6.4. Sử dụng hàm pow	21
6.5. Giải phương trình bậc 2	21
6.6. Sử dụng hàm Strings	23
6.7. Sử dụng hàm Datetimes	24
BÀI 7: List – Tuple – Set – Dictionary	26
7.1. List of animal	26
7.2. List numbers 1	26
7.3. List numbers 2	27
7.4. Tuple strings	28
7.5. Tuple numbers	29
7.6. Set numbers.....	30
7.7. Danh bạ điện thoại.....	31
7.8. Từ điển	33
BÀI 8: Phương thức/hàm	35
8.1. Tính năm âm lịch	35
8.2. Tính chỉ số BMI	35
8.3. Phương thức - hàm Bài 5.....	36
8.4. Phương thức - hàm bài 7	37
8.5. Lambda	37



8.6. Map, filter, reduce.....	38
BÀI 9: Module.....	39
9.1. Module 1	39
9.2. Module 2	39
BÀI 10: Xử lý ngoại lệ.....	40
10.1. Tính giá trị biểu thức	40
10.2. Xử lý ngoại lệ cho các bài tập.....	41
BÀI 11: Files I/O	42
11.1. Mở - đọc - đóng file.....	42
11.2. Mở - đọc – thống kê - đóng file	42
11.3. Mở - đọc – ghi file	43
11.4. Mở - đọc – ghi vào cuối file.....	44
11.5. Mở - đọc - đóng file CSV	45
11.6. Mở - đọc – ghi danh sách vào cuối file csv	46
Mục lục.....	48