

# LẬP TRÌNH PYTHON NÂNG CAO

Chương I: Thực Hành Luyện Tập Các Dạng Cơ Bản

Chương II: Chương Trình Con Và Thư Viện

Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Chương IV: Kiểu Dữ Liệu Xâu Ký Tự - Xử Lý Xâu Ký Tự

Chương V: Thực Hành Dữ Liệu Kiểu Xâu

Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

Chương IX: Lập Trình Và Giải Quyết Bài Toán

Chương X: Thực Hành Tổng Hợp

# Chương I: Thực Hành Luyện Tập Các Dạng Cơ Bản

## 1. Kiểm Tra Số.

Viết chương trình nhập vào số nguyên  $N$ , Hãy cho biết.

- $N$  là số chẵn hay lẻ?
- $N$  có chia hết cho 3 hay không?
- Xuất các số chẵn từ 1 đến  $N$  ra màn hình theo thứ tự tăng dần.
- Xuất các số chia hết cho 7 từ 1 đến  $N$  theo thứ tự giảm dần.
- $N$  có phải số nguyên tố không.
- Xuất ra màn hình các số hoàn hảo từ 1 đến  $N$ .
- Xuất ra màn hình các số chính phương từ 1 đến  $N$  theo thứ tự giảm dần.

# Chương I: Thực Hành Luyện Tập Các Dạng Cơ Bản

## 2. Các Bài Tập Tính Toán.

- Viết chương trình nhập vào 3 cạnh a, b, c của tam giác:  
Tính chu vi và diện tích tam giác đó.
- Viết chương trình nhập vào 2 số a và b và:  
Tính tổng, hiệu, tích và thương rồi xuất kết quả ra màn hình.
- Viết chương trình nhập vào 3 số a, b, c. Hãy tính:  
+ Trung bình cộng 3 số.  
+  $S = (a+b) - (a+c) / (b-c)$ .

# Chương I: Thực Hành Luyện Tập Các Dạng Cơ Bản

## 3. Xếp Loại Cho Học Sinh

Viết chương trình nhập vào điểm trung bình từng môn của 1 học sinh gồm:

Toán, Văn, Anh (Hệ số x2)

Lý, Hóa, Sinh, Sử, Địa lý, GDCD (Hệ số x1)

Hãy tính xếp loại cho học sinh đó, Biết:

Nếu điểm trung bình $\geq 8$	và không có môn nào dưới 6.5	Xếp loại giỏi
Từ 6.5 đến dưới 8	và không có môn nào dưới 5.0	Xếp loại khá
Từ 5 đến dưới 6.5	và không có môn nào dưới 3.5	Xếp loại TB
Còn lại	Xếp loại Yếu	

# Chương I: Thực Hành Luyện Tập Các Dạng Cơ Bản

## 4. Thanh Toán Bán Hàng.

Nhà bạn Hoàng bán trái cây, các loại trái cây Hoàng bán gồm:

Cam 30.000đ/kg

Chuối 20.000đ/kg

Bơ 35.000đ/kg

Táo 50.000đ/kg

Xoài 25.000đ/kg

Ổi 27.000đ/kg

Thanh Long 10.000đ/kg

Dưa hấu 15.000đ/kg

Hãy giúp Hoàng viết một chương trình để tính tổng thành tiền khi khách mua trái cây.

Nhập vào số lượng (kg) từng mặt hàng, cho ra kết quả là tổng tiền phải thanh toán.

*\*\* (Bài vận dụng lên nâng cao. không có gợi ý, hướng dẫn)*

# Chương II: Chương Trình Con Và Thư Viện

## 1. Khái Niệm

Trong khoa học máy tính, chương trình con (subprogram) hay subroutine là một đoạn chương trình được đóng gói thành một đơn vị trình, nó thực hiện một số tác vụ cụ thể mà chương trình cần thực hiện nhiều lần từ nhiều nơi trong thời gian chạy của nó. Khi chương trình cần đến tác vụ cụ thể đó thì bố trí chỉ thị gọi (call) đến chương trình con này và nhận kết quả nếu có sau khi nó thực thi xong .

Ngay từ lúc máy tính ra đời thì kỹ thuật lập trình kiểu cấu trúc module hóa với các chương trình con đã được thiết lập, và được củng cố trong các phần mềm lập trình hợp ngữ. Ngày nay trong ngôn ngữ bậc cao chương trình con được diễn đạt tùy theo ngôn ngữ là các hàm (function), thủ tục (procedure) và phương thức (method),... Một số ngôn ngữ lập trình, chẳng hạn Pascal và FORTRAN, phân biệt giữa hàm (một chương trình con có trả về giá trị) và thủ tục (không trả về giá trị). Các ngôn ngữ khác, ví dụ C và LISP, coi hai thuật ngữ này như nhau. Cái tên phương thức thường được dùng trong lập trình hướng đối tượng để gọi các chương trình con là một phần của các đối tượng.

# Chương II: Chương Trình Con Và Thư Viện

## 2. Khai báo và gọi thực hiện hàm trong Python.

Cú pháp khai báo:

```
def tên_hàm(tham số):  
    các câu lệnh  
    [return <giá trị trả về>]
```

Lưu ý:

- Tên hàm phải viết theo đúng quy tắc đặt tên trong Python.
- Tham số có thể có hoặc không.
- Phần thân hàm (các khối lệnh) phải viết lùi vào theo quy định của Python.

Cú pháp gọi hàm:

```
tên_hàm(danh sách biến)
```

# Chương II: Chương Trình Con Và Thư Viện

## 2. Khai báo và gọi thực hiện hàm trong Python.

Ví dụ 1:

```
Bài tập.py - D:\Python\Bài tập.py (3.10.11)
File Edit Format Run Options Window Help
def dien_tich_tam_giac(a,h):
    print("Diện tích là: ",a*h)

canh_day = int(input("Nhập vào cạnh đáy: "))
chieu_cao = int(input("Nhập vào chiều cao: "))
dien_tich_tam_giac(canh_day,chieu_cao)
```

Tên hàm Tham số

Gọi hàm

```
>>>
=====
Nhập vào cạnh đáy: 6
Nhập vào chiều cao: 8
Diện tích là: 48
```



# Chương II: Chương Trình Con Và Thư Viện

## 2. Khai báo và gọi thực hiện hàm trong Python.

Ví dụ 2:

```
File Edit Format Run Options Window Help
def dien_tich_hcn(a,b):
    s = a*b
    return s
def chu_vi_hcn(a,b):
    c = (a + b)*2
    return c

c_dai = int(input("Nhập vào cạnh đáy: "))
c_rong = int(input("Nhập vào chiều cao: "))
print("Diện tích là: ",dien_tich_hcn(c_dai,c_rong))
print("Chu vi là: ",chu_vi_hcn(c_dai,c_rong))
```

Tên hàm Tham số

Giá trị trả về

Gọi hàm

```
Nhập vào cạnh đáy: 4
Nhập vào chiều cao: 3
Diện tích là: 12
Chu vi là: 14
```

# Chương II: Chương Trình Con Và Thư Viện

## 3. Hàm có sẵn

Để đáp ứng tốt hơn nhu cầu đa dạng trong xử lý thông tin và giúp cho việc lập trình thuận lợi, một loạt các hàm được xây dựng sẵn, gắn với mỗi hệ thống ngôn ngữ lập trình bậc cao. Mỗi tập hợp gồm một số các hàm được xây dựng sẵn thường gọi là một thư viện. Trong chương trình của mình, người lập trình chỉ cần gọi hàm có sẵn (trong một thư viện) thực hiện mà không cần phải tự mình xây dựng lại hàm đó. Số lượng thư viện, số hàm trong mỗi thư viện, lời gọi tới chúng,... có thể thay đổi theo thời gian và phụ thuộc vào hệ thống ngôn ngữ lập trình.

Ngay từ những bước đầu tiên làm quen với lập trình Python, em đã sử dụng một số hàm trong thư viện chuẩn của Python như `print ()`, `input ()`,... và một số hàm toán học trong thư viện `math`. Thư viện `math` cung cấp các hằng và hàm toán học, ví dụ hàm `gcd (x,y)` trả về ước chung lớn nhất của `x` và `y`.

Để có thể sử dụng các hàm trong thư viện cần kết nối thư viện hoặc hàm đó với chương trình. Hai cách thông dụng để kết nối hàm và thư viện

# Chương II: Chương Trình Con Và Thư Viện

## 3. Hàm có sẵn

*Cú*

Kết nối chương trình tới tất cả hàm trong thư viện math.

```
import math
```

cách gọi hàm:

```
<Tên thư viện>.<tên hàm(tham số)>
```

Ví dụ:

```
File Edit Format Run Options Window Help
import math
a = int(input('a = '))
b = int(input('b = '))
print('Ước chung lớn nhất là: ',math.gcd(a,b))
```

*pháp:*

Kết nối chương trình tới hàm gcd trong thư viện math.

```
from math import gcd
```

```
<tên hàm(tham số)>
```

```
File Edit Format Run Options Window Help
from math import gcd
a = int(input('a = '))
b = int(input('b = '))
print('Ước chung lớn nhất là: ',gcd(a,b))
```

# Chương II: Chương Trình Con Và Thư Viện

## 4. Bài tập

Bài 1: Viết chương trình nhập vào 2 số  $a, b$  tìm ước chung lớn nhất của 2 số.

Bài 2: Viết chương trình nhập vào 3 cạnh  $a, b, c$  của 1 tam giác.

- a. Tính chu vi tam giác.
- b. Tính diện tích tam giác.

Bài 3: Viết chương trình giải phương trình bậc 1.

Bài 4: Viết chương trình nhập vào số nguyên  $n$ , xuất ra màn hình  $n$  số fibonacci theo thứ tự tăng dần từ 1.

*\*\*Tất cả sử dụng chương trình con hoặc hàm có sẵn (nếu có)*

# Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Câu 1: Chọn phát biểu đúng.

- a. Khi khai báo hàm, không nhất thiết phải có tham số.
- b. Khi khai báo hàm, bắt buộc phải có tham số.
- c. Tên hàm có thể đặt tự do.

Câu 2: Câu lệnh khai báo hàm nào sau đây đúng?

- a. `function tong(a,b)`
- b. `def trung_binh_cong(a,b,c)`
- c. `def ucln(a,b):`

# Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Câu 3: Chọn phát biểu sai.

- a. Các câu lệnh trong hàm phải lùi vào theo đúng quy tắc trong python.
- b. Trong Python khai báo hàm bắt đầu bằng function.
- c. Tham số của hàm có thể có hoặc không.

Câu 4: Tìm lỗi sai trong đoạn code sau và sửa lại cho đúng:

```
def dien_tich_hcn(a,b)
    s = a*b
    return s
```

# Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Câu 5: Các tên hàm nào đúng?

- a. else
- b. dien\_tich
- c. trung bình cộng

Câu 6: Kết quả của chương trình bên là?

- a. 12
- b. 14
- c. Diện tích là: 12
- d. Diện tích là:14

```
def dien_tich_hcn(a,b):  
    s = a*b  
    return s  
c_dai = int(input("Nhập vào cạnh đáy: "))  
c_rong = int(input("Nhập vào chiều cao: "))  
print("Diện tích là: ",dien_tich_hcn(c_dai,c_rong))
```

Input:

```
Nhập vào cạnh đáy: 4  
Nhập vào chiều cao: 3
```

# Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Câu 7: Có bao nhiêu kiểu kết nối với hàm có sẵn.

- a. 1.
- b. 2.
- c. 3.

Câu 8: Cách kết nối với tất cả các hàm trong thư viện math là?

- a. `import math`.
- b. `include math`.
- c. `using math`.



# Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Câu 9: Cách kết nối đến hàm gcd trong thư viện math là:

- a. `from math import gcd`.
- b. `using gcd`.
- c. `include gcd`.

Câu 10: Tìm lỗi sai trong chương trình sau:

```
import math
a = int(input('a = '))
b = int(input('b = '))
print('Ước chung lớn nhất là: ',gcd(a,b))
```

# Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Câu 11: Viết chương trình nhập vào 2 số  $a$  và  $b$ .

Tính tổng, hiệu, tích thương  $a$  và  $b$  và xuất kết quả ra màn hình.

Tìm ước chung lớn nhất và bội chung nhỏ nhất của 2 số.

Câu 12: Viết chương trình nhập vào số nguyên  $n$ , xuất ra màn hình:

Các số nguyên tố từ 1 đến  $n$ .

Các số hoàn hảo từ 1 đến  $n$ .

Các số chính phương từ 1 đến  $n$ .

Dãy  $n$  số fibonacci.

# Chương III: Thực Hành Lập Trình Với Hàm Và Thư Viện

Câu 13: Viết chương trình giải phương trình bậc 2.

Câu 14: Viết chương trình nhập vào 3 cạnh a, b, của 1 tam giác.

Kiểm tra xem đó là tam giác gì?.

Tính chu vi và diện tích tam giác đó.

Câu 15: Viết chương trình nhập vào điểm 3 môn: Word, Excel, PowerPoint. xuất kết quả học lực:

Nếu điểm TB 3 môn trên 8, xếp loại giỏi. Từ 6.5 - 8 xếp loại khá. 5 - 6.5, TB. Dưới 5: Thi trượt.

Nếu 1 môn bất kì dưới 5 => Thi trượt.

# Chương IV: Kiểu Dữ Liệu Xâu Ký Tự - Xử Lý Xâu Ký Tự

## 1. Kiểu xâu ký tự.

- Xâu ký tự trong Python là dãy các ký tự Unicode.
- Xâu có thể được coi là danh sách các ký tự nhưng không thay đổi từng ký tự của xâu.
- Truy cập từng ký tự của xâu qua chỉ số, chỉ số từ 0 đến độ dài `len()` -1. Vd: `s[1]`.
- Python không cho phép thay đổi từng ký tự của một xâu (khác với danh sách).
- Python không có kiểu dữ liệu ký tự.
- Xâu rỗng được định nghĩa: `empty = ""`.

# Chương IV: Kiểu Dữ Liệu Xâu Ký Tự - Xử Lý Xâu Ký Tự

## 2. Các thao tác với xâu.

a) Khởi tạo xâu:

<Tên biến> = <Xâu>

VD: 

```
s1 = ""      #tạo xâu rỗng
s2 = "Tin Hoc Sao Viet"
```

b) Duyệt qua từng phần tử của xâu.

VD: 

```
s2 = "Tin Hoc Sao Viet"
for i in s2:
    print(i,end=" ")
```

Output: `T i n H o c S a o V i e t`

# Chương IV: Kiểu Dữ Liệu Xâu Ký Tự - Xử Lý Xâu Ký Tự

## 2. Các thao tác với xâu.

c) Một số hàm với xâu.

Hàm	Ý nghĩa	Ví dụ
<code>str(x)</code>	Trả về xâu tương ứng giá trị của x	<code>n = 1025; s = str(n)</code>
<code>len(s)</code>	Trả về độ dài của xâu s	<code>s = "Viet Nam"; print(len(s))</code>
<code>ord(ch)</code>	Trả về mã của ký tự ch trong mã ASCII hoặc Unicode	<code>ch = "A"; print(ord(ch))</code>
<code>chr(x)</code>	Trả về ký tự tương ứng với mã x trong mã ASCII hoặc Unicode	<code>x = 'a'; print(chr(x))</code>
<code>upper()</code>	Trả về xâu in hoa tương ứng	<code>s = "Ha Noi"; print(s.upper())</code>
<code>lower()</code>	Trả về xâu in thường tương ứng	<code>s = "Viet Nam"; print(s.lower())</code>
<code>capitalize()</code>	Trả về xâu chỉ in hoa ký tự đầu còn lại in thường	<code>s = "HELLO"; print(s.capitalize())</code>
<code>title()</code>	Trả về xâu in hoa các ký tự đầu của các từ	<code>s = "Xin chao"; print(s.title())</code>

# Chương IV: Kiểu Dữ Liệu Xâu Ký Tự - Xử Lý Xâu Ký Tự

## 2. Các thao tác với xâu.

c) Một số hàm với xâu.

<code>swapcase()</code>	Trả về xâu có ký tự in hoa thành thường và ngược lại	<code>s = "VniTeach"; print(s.swapcase())</code>
<code>strip()</code>	Trả về xâu sau khi đã xóa các ký tự trống	<code>s = "Ha Noi"; print(s.strip())</code>
<code>lstrip()</code>	Trả về xâu sau khi đã xóa các ký tự trống phía trái	<code>s = "Viet Nam"; print(s.lstrip())</code>
<code>rstrip()</code>	Trả về xâu sau khi đã xóa các ký tự trống phía phải	<code>s = "HELLO"; print(s.rstrip())</code>
<code>find()</code>	Trả về chỉ số đầu tiên tìm thấy của xâu s	<code>st = "VniTeach"; print(st.find("ni"))</code>
<code>replace()</code>	<code>replace(&lt;old&gt;,&lt;new&gt;,[count = -1])</code>	Thay thế old thành new, count lần Mặc định count = -1 không hạn chế lần thay
<code>split()</code>	<code>split([&lt;sep = none&gt;, maxsplit() = -1])</code>	Tách xâu thành danh sách các xâu con
<code>join()</code>	<code>join(&lt;danh sách&gt;)</code>	Gộp danh sách thành xâu (ngược lại split())

# Chương IV: Kiểu Dữ Liệu Xâu Ký Tự - Xử Lý Xâu Ký Tự

## 3. Bài tập.

Bài 1: Viết chương trình xuất ra màn hình họ tên của em.

Bài 2: Viết chương trình nhập vào tên của em, đếm xem tên của em có bao nhiêu ký tự.

Bài 3: Viết chương trình nhập vào chuỗi ký tự bất kì, đếm xem có bao nhiêu chữ “t” trong tên.

Bài 4: Viết chương trình nhập vào 1 chuỗi ký tự, đếm xem trong chuỗi có bao nhiêu khoảng trắng “ ”.

Bài 5: Viết chương trình nhập vào họ tên em viết thường, hãy in hoa tất cả chữ cái đầu trong tên.



# Chương V: Thực Hành Dữ Liệu Kiểu Xâu

Câu 1: Kiểu xâu trong Python là gì?

- a) Là dãy các kí tự Unicode.
- b) Là một dạng số.
- c) Là kiểu dữ liệu kí tự.

Câu 2: Cách khởi tạo xâu nào là đúng trong Python?

- a) `s = xin chao`
- b) `s = "xin chao"`
- c) `<s> = <xin chao>`

# Chương V: Thực Hành Dữ Liệu Kiểu Xâu

Câu 3: Để đổi số 01234 thành kiểu xâu ta dùng?

a) `print(01234)`.

b) `int(01234)`.

c) `str(01234)`.

Câu 4: Trong Python, Hàm nào dùng để in hoa tất cả chuỗi?

a) `upper()`.

b) `=UPPER()`.

c) `in_hoa()`.

# Chương V: Thực Hành Dữ Liệu Kiểu Xâu

Câu 5: Cho xâu `s = "Tin Hoc Sao Viet"` . `s[0]` có giá trị là?

a) "".

b) 0.

c) T.

Câu 6: Trong Python, Hàm nào dùng để in hoa chữ cái đầu mỗi từ?

a) `upper()`.

b) `capitalize()`.

c) `title()`.

# Chương V: Thực Hành Dữ Liệu Kiểu Xâu

Câu 7: Kết quả chương trình sau là? 

```
s2 = "Tin Hoc Sao Viet"
print(s2[4])
```

a) "".

b) 4.

c) H.

Câu 8: Chương trình sau đúng hay sai, tại sao?

```
s = "tin hoc Sao Viet"
s[1] = "T"
s[4] = "H"
print(s)
```

# Chương V: Thực Hành Dữ Liệu Kiểu Xâu

Câu 9: Câu lệnh khai báo kiểu kí tự đúng trong Python là?

a) `s = hello`.

b) `s = "xin_chao"`.

c) Không có kiểu kí tự trong Python.

Câu 10: Tìm lỗi sai trong chương trình sau?

```
s = tin hoc Sao Viet  
print(s)
```

# Chương V: Thực Hành Dữ Liệu Kiểu Xâu

Câu 11: Viết chương trình nhập vào xâu bất kì.

Tính độ dài xâu đó.

In hoa chữ cái đầu mỗi từ.

Xuất ra màn hình kí tự thứ 4,5,6.

Câu 12: Viết chương trình nhập vào dãy số bất kì, đưa ra màn hình xâu ngược lại của số đó.

Câu 13: Viết chương trình nhập vào chuỗi bất kì.

Có bao nhiêu kí tự khoảng trắng “ ” trong chuỗi.

Thay thế khoảng trắng “ ” trong chuỗi thành “\_”.

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 1. Kiểu dữ liệu danh sách (list).

list là một kiểu dữ liệu rất đa năng trong Python có khả năng chứa một danh sách các phần tử (có thể khác kiểu).

Python quy định giá trị kiểu list phải viết trong cặp dấu ngoặc vuông, các phần tử tách nhau bởi dấu phẩy. Các phần tử trong danh sách được đánh chỉ số (index). Chỉ số bắt đầu từ 0 và theo vị trí xuất hiện của phần tử.

Vd:

```
danh_sach_so = [1, 2, 3, 4, 5]
danh_sach_ten = ['John', 'Jane', 'Alice', 'Bob']
danh_sach_honhop = [1, 'Alice', True, 3.14]
```

Danh sách có thể chứa cả danh sách khác. Khi này nó được gọi là danh sách lồng nhau (nested list).

Vd:

```
danh_sach_lon = [[1, 2, 3], [4, 5, 6], ['a', 'b', 'c']]
```

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 2. Truy xuất phần tử trong danh sách.

Để truy xuất các phần tử trong một danh sách (list) trong Python, bạn có thể sử dụng chỉ số (index) của phần tử đó. Chỉ số bắt đầu từ 0 cho phần tử đầu tiên, 1 cho phần tử thứ hai và tiếp tục tăng dần cho các phần tử tiếp theo.

Vd: Ta có danh sách: `danh_sach_so = [1, 2, 3, 4, 5]`

Truy xuất phần tử bằng chỉ số:

```
phan_tu_dau_tien = danh_sach_so[0]    # Kết quả: 1
phan_tu_thu_hai = danh_sach_so[1]    # Kết quả: 2
phan_tu_cuoi_cung = danh_sach_so[4]   # Kết quả: 5
```

Truy xuất phần tử bằng chỉ số âm (để lấy phần tử từ cuối danh sách):

```
phan_tu_cuoi_cung = danh_sach_so[-1]  # Kết quả: 5
phan_tu_thu_nhi_cuoi_cung = danh_sach_so[-2]  # Kết quả: 4
```



# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 2. Truy xuất phần tử trong danh sách.

Truy xuất nhiều phần tử bằng slicing (kết hợp cú pháp start:stop:step):

```
phan_tu_tu_2_toi_4 = danh_sach_so[1:4]      # Kết quả: [2, 3, 4]
phan_tu_chia_2 = danh_sach_so[::2]         # Kết quả: [1, 3, 5] (lấy các phần tử ở vị trí chẵn)
```

Truy xuất phần tử trong danh sách lồng nhau (nested list):

```
danh_sach_lon = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
phan_tu_3_dau_tien_cua_danh_sach_lon = danh_sach_lon[0][2]

# Kết quả: 3 (phần tử thứ 3 trong danh sách con đầu tiên)
```

\*Chú ý rằng nếu bạn sử dụng chỉ số vượt quá giới hạn của danh sách, Python sẽ gây ra lỗi "IndexError". Vì vậy, hãy đảm bảo rằng chỉ số của bạn nằm trong phạm vi của danh sách trước khi truy xuất để tránh lỗi này.

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 3. Thêm mới, cập nhật và xóa danh sách.

Trong Python, bạn có thể thêm mới, cập nhật và xóa các phần tử trong danh sách (list) bằng một số phương pháp sau:

Sử dụng `append()`: Thêm một phần tử vào cuối danh sách.

```
danh_sach_so = [1, 2, 3]
danh_sach_so.append(4)
print(danh_sach_so) # Kết quả: [1, 2, 3, 4]
```

Sử dụng `insert()`: Thêm một phần tử vào vị trí cụ thể trong danh sách.

```
danh_sach_so = [1, 2, 3]
danh_sach_so.insert(1, 10) # Chèn số 10 vào vị trí thứ 1
print(danh_sach_so) # Kết quả: [1, 10, 2, 3]
```

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 3. Thêm mới, cập nhật và xóa danh sách.

Sử dụng toán tử “+” để kết hợp danh sách (list concatenation):

```
danh_sach_so = [1, 2, 3]
danh_sach_moi = [4, 5, 6]
danh_sach_tong = danh_sach_so + danh_sach_moi
print(danh_sach_tong)    # Kết quả: [1, 2, 3, 4, 5, 6]
```

Cập nhật phần tử trong danh sách:

```
danh_sach_so = [1, 2, 3]
danh_sach_so[1] = 20    # Cập nhật giá trị của phần tử ở vị trí thứ 1 thành 20
print(danh_sach_so)    # Kết quả: [1, 20, 3]
```

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 3. Thêm mới, cập nhật và xóa danh sách.

Sử dụng `remove()` để xóa một phần tử cụ thể khỏi danh sách.

```
danh_sach_so = [1, 2, 3]
danh_sach_so.remove(2) # Xóa số 2 khỏi danh sách
print(danh_sach_so)   # Kết quả: [1, 3]
```

Sử dụng `pop()` để xóa một phần tử tại một vị trí cụ thể và trả về giá trị của phần tử bị xóa.

```
danh_sach_so = [1, 2, 3]
phan_tu_bi_xoa = danh_sach_so.pop(1) # Xóa phần tử ở vị trí thứ 1 (số 2) và lưu giá trị vào biến phan_tu_bi_xoa
print(phan_tu_bi_xoa)   # Kết quả: 2
print(danh_sach_so)     # Kết quả: [1, 3]
```

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 4. Một số thao tác với danh sách.

Lấy độ dài của danh sách:

Bạn có thể sử dụng hàm len() để lấy số lượng phần tử trong danh sách.

```
danh_sach_so = [1, 2, 3, 4, 5]
do_dai = len(danh_sach_so)
print(do_dai)    # Kết quả: 5
```

Kiểm tra phần tử có tồn tại trong danh sách hay không:

Bạn có thể sử dụng từ khoá in để kiểm tra xem một phần tử có tồn tại trong danh sách hay không.

```
danh_sach_so = [1, 2, 3, 4, 5]
kiem_tra = 3 in danh_sach_so
print(kiem_tra)    # Kết quả: True (vì số 3 tồn tại trong danh sách)
```

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 4. Một số thao tác với danh sách.

Duyệt qua từng phần tử trong danh sách:

Bạn có thể sử dụng vòng lặp for để duyệt qua từng phần tử trong danh sách.

```
danh_sach_so = [1, 2, 3, 4, 5] # Kết quả:  
for so in danh_sach_so:      # 1  
    print(so)                # 2  
                              # 3  
                              # 4  
                              # 5
```

Sắp xếp danh sách:

Bạn có thể sử dụng phương thức sort() để sắp xếp danh sách theo thứ tự tăng dần (mặc định) hoặc sort(reverse=True) để sắp xếp theo thứ tự giảm dần.

```
danh_sach_so = [5, 2, 1, 4, 3]  
danh_sach_so.sort() # Sắp xếp tăng dần  
print(danh_sach_so) # Kết quả: [1, 2, 3, 4, 5]  
  
danh_sach_so.sort(reverse=True) # Sắp xếp giảm dần  
print(danh_sach_so) # Kết quả: [5, 4, 3, 2, 1]
```

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 4. Một số thao tác với danh sách.

Sao chép danh sách: copy()

```
danh_sach_so = [1, 2, 3]
danh_sach_so_moi = danh_sach_so.copy()    # Hoặc danh_sach_so_moi = list(danh_sach_so)
danh_sach_so_moi.append(4)
print(danh_sach_so)      # Kết quả: [1, 2, 3]
print(danh_sach_so_moi)  # Kết quả: [1, 2, 3, 4]
```

Tạo danh sách từ chuỗi:

Bạn có thể chuyển đổi một chuỗi thành danh sách bằng cách sử dụng phương thức split() và cung cấp ký tự phân tách.

```
chuoi = "apple,banana,orange"
danh_sach_trai_cay = chuoi.split(",")    # Chuyển chuỗi thành danh sách, phân tách bằng dấu phẩy
print(danh_sach_trai_cay)                # Kết quả: ['apple', 'banana', 'orange']
```

# Chương VI: Kiểu Dữ Liệu Danh Sách - Xử Lý Danh Sách

## 5. Bài tập.

Viết chương trình nhập vào danh sách các số chẵn ngẫu nhiên.

Xuất ra màn hình các phần tử trong danh sách.

Xuất ra màn hình các số chẵn.

Xuất ra màn hình các số chia hết cho 5.

Xuất ra màn hình các số nguyên tố.

Xuất ra màn hình số lớn nhất.

Xuất ra màn hình giá trị trung bình.



# Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 1: Kiểu danh sách trong Python là gì?

- a) Là một kiểu dữ liệu rất đa năng trong Python có khả năng chứa một danh sách các phần tử.
- b) Là xâu kí tự.
- c) Là kiểu dữ liệu kí tự.

Câu 2: Cách khởi danh sách nào là đúng trong Python?

- a) `s = []`
- b) `s = "xin chao"`
- c) `s = <xin chao>`

# Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 3: Câu lệnh nào sau đây sai?

a) `s = [4,5,6]`

b) `s = [[2,4,6],[8,9]]`

c) `s = [a,b,c]`

Câu 4: Tìm lỗi sai trong chương trình sau?

```
danh_sach = (1,2,3,'a','b','c')  
print(danh_sach)
```

## Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 5: Kết quả của chương trình bên là?

```
danh_sach = [1,2,3,'a','b','c']  
do_dai = len(danh_sach)  
print(do_dai)
```

a) 1,2,3.

b) 6.

c) 1,2,3,a,b,c.

Câu 6: Để khai báo danh sách a và khởi tạo danh sách a có 3 phần tử 10, 20, 30, phương án nào sau đây đúng?

a) a = (10,20,30)

b) a = {10,20,30}

c) a = [10,20,30]

## Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 7: Để khai báo danh sách a rỗng, phương án nào sau đây đúng?

a) `a == []`

b) `= 0`

c) `= []`

Câu 8: Để xóa phần tử thứ 2 trong danh sách a ta dùng lệnh:

a) `del(a[1])`

b) `remove(a[2])`

c) `remove(a[1])`

## Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 9: Cho đoạn lệnh sau:

```
a=[7,3,8,1,9]
del(a[3])
```

Danh sách a thu được là?

- a) [7,8,1,9]
- b) [7,3,1,9]
- c) [7,3,8,9]

Câu 10: Để xóa phần tử thứ 2 trong danh sách a ta dùng lệnh:

- a) del(a[1])
- b) remove(a[2])
- c) remove(a[1])

## Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 11: Cho đoạn lệnh sau:

```
a=[5,7,3,4,1]  
a.insert(2,9)
```

Danh sách a thu được là?

- a) [5,7,9,3,4,1]
- b) [5,7,3,4,1,2,9]
- c) [2,9,5,7,3,4,1]

Câu 12: Để sắp xếp danh sách a, phương án nào sau đây đúng?

- a) a.reverse()
- b) a.sort()
- c) sort(a)

## Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 13: Cho danh sách  $a = [1, 2, 3, 4, 5]$ . Lệnh nào dưới đây sẽ thêm số 6 vào danh sách a?

a) `a.append(6)`

b) `a.insert(6)`

c) `a.add(6)`

Câu 14: Cho danh sách  $a = [1, 2, 3, 4, 5]$ . Lệnh nào dưới đây sẽ trả về số lượng phần tử trong danh sách?

a) `len(a)`

b) `size(a)`

c) `count(a)`

# Chương VII: Thực Hành Với Kiểu Dữ Liệu Danh Sách

Câu 15: Viết chương trình nhập vào danh sách có  $n$  số nguyên.

- Đếm xem có bao nhiêu số chẵn trong dãy số.
- Độ dài của danh sách là? xuất ra màn hình.
- Hãy sắp xếp danh sách GIẢM DẦN và đưa kết quả ra màn hình.
- Lấy ra phần tử nằm giữa danh sách. (lấy ra 2 nếu danh sách số phần tử chẵn)
- Tính trung bình cộng các phần tử và thêm nó vào cuối danh sách. xuất danh sách ra màn hình.
- Đưa ra màn hình tổng các số ở vị trí chẵn.
- Đưa ra màn hình các số nguyên tố có trong danh sách.
- Đưa ra màn hình các số chính phương có trong danh sách.



# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 1. Lỗi trong chương trình và kiểm thử

Chương trình chúng ta viết ra rất có thể có lỗi. Ngay cả những người lập trình giàu kinh nghiệm cũng có thể viết ra những chương trình còn lỗi. Quá trình xác định lỗi và sửa lỗi được gọi là gỡ lỗi. Người lập trình thường gặp các loại lỗi như sau

- Lỗi cú pháp là lỗi câu lệnh viết không theo đúng quy định của ngôn ngữ, ví dụ như thiếu hoặc thừa ngoặc trong biểu thức, tên biến sai quy cách,... Loại lỗi này được môi trường lập trình phát hiện và thông báo khá cụ thể, rõ ràng, cả về đặc trưng lỗi và nơi xảy ra lỗi.
- Lỗi ngoại lệ (Exceptions Error) còn gọi là lỗi Runtime, là lỗi xảy ra khi chương trình đang chạy, một lệnh nào đó không thể thực hiện được. Lỗi này sẽ được thông báo ngay trên màn hình
- Lỗi ngữ nghĩa (còn gọi là lỗi logic) là lỗi mặc dù các câu lệnh viết đúng quy định của ngôn ngữ nhưng sai trong thao tác xử lý nào đó, ví dụ như viết nhầm dấu phép tính, nhầm tên biến, gọi hàm có tham số không đúng kiểu, chỉ sai phạm vi duyệt, thiếu câu lệnh cần thiết,... Đây là loại lỗi rất khó phát hiện vì trong rất nhiều trường hợp chương trình vẫn được thực hiện trọn vẹn, nhưng kết quả đưa ra không phù hợp hoặc kết quả chỉ sai với một số bộ dữ liệu vào nào đó.

# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 1. Lỗi trong chương trình và kiểm thử

Việc đọc kỹ lại chương trình để tìm lỗi chỉ thích hợp với các chương trình ngắn, đơn giản và ngay cả trong trường hợp này cũng mất khá nhiều thời gian, công sức.

Môi trường lập trình của những ngôn ngữ lập trình bậc cao nói chung và Python nói riêng có công cụ hỗ trợ cho người dùng tìm lỗi.

Các lỗi ngữ nghĩa thì khó phát hiện hơn, chỉ có thể đoán nhận và tìm thấy thông qua quan sát kết quả thực hiện chương trình với các bộ dữ liệu vào (các bộ test) khác nhau. Những kết quả kiểm thử như vậy có thể dẫn đến việc chỉnh lí, bổ sung hoặc thay đổi thuật toán.

Khi không phát hiện thêm lỗi, ta có thể đưa chương trình vào khai thác, phục vụ mục đích thực tiễn của bài toán.

# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 1. Lỗi trong chương trình và kiểm thử

Để kiểm tra tính đúng đắn của chương trình so với yêu cầu của đề bài, trước hết cần chuẩn bị các bộ dữ liệu vào.

Dữ liệu kiểm thử phải phù hợp với các ràng buộc đã cho và chia thành ba nhóm

- Kiểm thử những trường hợp thường gặp trong thực tế.
- Kiểm thử những trường hợp đặc biệt (ví dụ, khi danh sách chỉ bao gồm một phần tử).
- Kiểm thử những trường hợp các tham số nhận giá trị lớn nhất có thể.

Dữ liệu kiểm thử ở hai nhóm đầu cần có kích thước đủ nhỏ để ta có thể kiểm chứng các kết quả do chương trình đưa ra. Dữ liệu ở nhóm thứ hai là để kiểm tra tính trọn vẹn của thuật toán trong thực hiện chương trình. Dữ liệu ở nhóm thứ ba nhằm kiểm tra tính hiệu quả của chương trình và tính hợp lý trong tổ chức dữ liệu. Có thể chương trình viết ra đã lưu quá nhiều dữ liệu trung gian nên khi dữ liệu vào có kích thước lớn thì không đủ bộ nhớ để thực hiện. Kiểm thử với dữ liệu thuộc nhóm thứ ba, ta chỉ có thể đánh giá được tính hợp lý của kết quả.

# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 1. Lỗi trong chương trình và kiểm thử

Ví dụ:

*Lỗi cú pháp:*

```
File Edit Format Run Options Window Help
danh_sach = [1,2,3,'a','b','c']
do_dai = len(danh_sach)
cout>>do_dai
```

```
File Edit Format Run Options Window Help
n := int(input("Nhập vào số nguyên n: "))
a := 0
while a<=n:
    print(a)
    a = a + 2
```

*Lỗi ngoại lệ:*

```
nhap vao so phan tu: 4
Nhap vao phan tu thu 1
2
Nhap vao phan tu thu 2
1
Nhap vao phan tu thu 3
4
Nhap vao phan tu thu 4
6
Traceback (most recent call last):
  File "D:\Python\Bài tập.py", line 14, in <module>
    so_chan(ds)
TypeError: so_chan() takes 0 positional arguments but 1 was given
```

# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 1. Lỗi trong chương trình và kiểm thử

Ví dụ:

*Lỗi Ngữ Nghĩa:*

```
File Edit Format Run Options Window Help
def nhapds(a):
    n = int(input('nhap vao so phan tu: '))
    for i in range(0,n):
        print('Nhap vao phan tu thu', i+1)
        k = int(input())
        a.insert(i,k)
    return a
def so_chan(b):
    for i in range(0,len(b)):
        if b[i] % 2 == 0:
            print(i)
ds = []
nhapds(ds)
print('Danh sach so chan la: ')
so_chan(ds)
```

```
nhap vao so phan tu: 5
Nhap vao phan tu thu 1
2
Nhap vao phan tu thu 2
6
Nhap vao phan tu thu 3
1
Nhap vao phan tu thu 4
7
Nhap vao phan tu thu 5
9
Danh sach so chan la:
0
1
```

# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 2. Bổ sung câu lệnh trung gian để kiểm tra lỗi.

- Bổ sung vào giữa các dòng lệnh câu lệnh `print()` để in ra biến trung gian, kiểm tra quy trình hay thuật toán.

- Ví dụ: Nhập từ bàn phím hai số tự nhiên  $m, n$ . Tìm UCLN của hai số này.

```
#Tính UCLN của m, n
m = int(input("Nhập số tự nhiên m: "))
n = int(input("Nhập số tự nhiên n: "))
while m != n:
    if m < n:
        n = n - m
    else:
        m = m - n
print("Đáp số:", m)
```

- Bổ sung biến  $k$  và hai lệnh `print()` như sau:

```
m = int(input("Nhập số tự nhiên m: "))
n = int(input("Nhập số tự nhiên n: "))
k = 0
while m != n:
    k = k + 1
    print("Vòng lặp", k, ":", m, n)
    if m < n:
        n = n - m
    else:
        m = m - n
print("Kết thúc vòng lặp:", m, n)
print("Đáp số:", m)
```

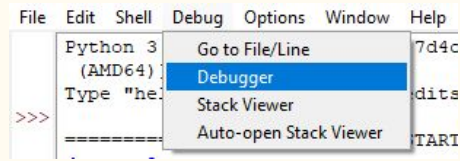
```
Nhập số tự nhiên m: 20
Nhập số tự nhiên n: 16
Vòng lặp 1 : 20 16
Vòng lặp 2 : 4 16
Vòng lặp 3 : 4 12
Vòng lặp 4 : 4 8
Kết thúc vòng lặp: 4 4
Đáp số: 4
```

# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 3. Sử dụng công cụ Debugger để kiểm tra.

Trong cửa sổ Shell, mở chương trình cần gỡ lỗi.

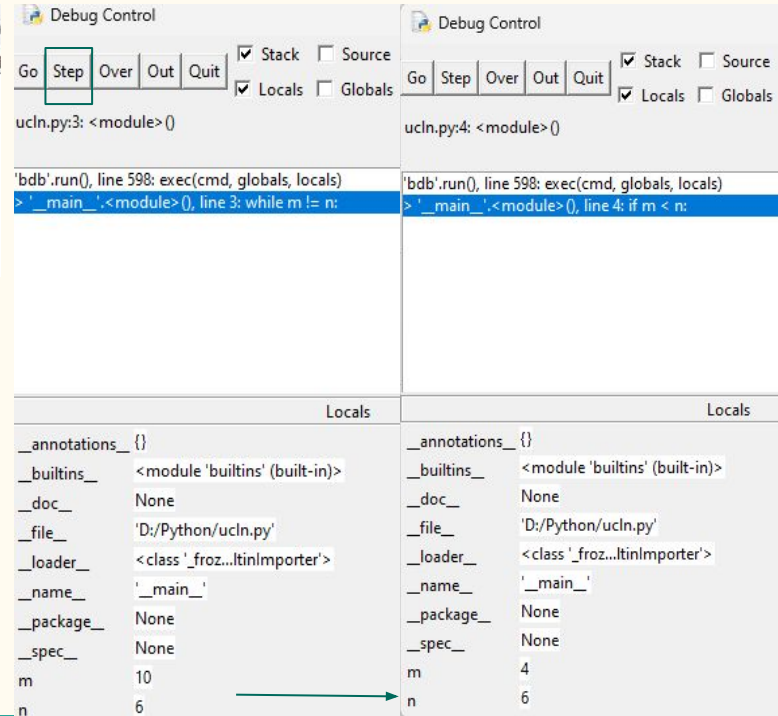
Chọn Debug trên cửa sổ Shell > Debugger.



Mở chế độ Run Mode (F5) để chạy chương trình.

Chọn Step để thực hiện câu lệnh hiển thị phía dưới. Riêng câu lệnh vào ra, có thể nhấn chuột nhiều lần hoặc click Over để bỏ qua nhanh nhiều bước.

```
ucln.py - D:/Python/ucln.py
File Edit Format Run O
m = 10
n = 6
while m != n:
    if m < n:
        n = n - m
    else:
        m = m - n
print("dap so", m)
```



The image shows the Python IDE's Debug Control and Locals panels. The Debug Control panel has 'Step' selected, and the Locals panel shows the current state of variables.

Locals	Locals
<code>__annotations__</code> : {}	<code>__annotations__</code> : {}
<code>__builtins__</code> : <module 'builtins' (built-in)>	<code>__builtins__</code> : <module 'builtins' (built-in)>
<code>__doc__</code> : None	<code>__doc__</code> : None
<code>__file__</code> : 'D:/Python/ucln.py'	<code>__file__</code> : 'D:/Python/ucln.py'
<code>__loader__</code> : <class 'froz...ItinImporter'>	<code>__loader__</code> : <class 'froz...ItinImporter'>
<code>__name__</code> : '__main__'	<code>__name__</code> : '__main__'
<code>__package__</code> : None	<code>__package__</code> : None
<code>__spec__</code> : None	<code>__spec__</code> : None
<code>m</code> : 10	<code>m</code> : 4
<code>n</code> : 6	<code>n</code> : 6

# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 4. Bài tập thực hành.

Bài 1: Viết chương trình nhập vào 2 số nguyên a, b

- Tìm ước chung lớn nhất của 2 số.
- Tìm bội chung nhỏ nhất giữa 2 số.
- Xuất ra màn hình tất cả ước chung của 2 số.

*\*\* (Không sử dụng hàm có sẵn)*

Bài 2: Tìm lỗi sai trong chương trình bên và sửa lại để chương trình có thể chạy đúng.

- Chương trình nhập vào danh sách các số nguyên, Xuất ra màn danh sách các phần tử có giá trị là số chẵn.

```
File Edit Format Run Options Window Help
def nhapds(a):
    n = int(input('nhap vao so phan tu: '))
    for i in range(0,n):
        print('Nhap vao phan tu thu', i+1)
        k = int(input())
        insert(i,k)
    return a
def so_chan(b)
    for i in range(0,len(b)):
        if b[i] % 2 == 0:
            print(i)

ds = []
nhapds(ds)
print('Danh sach so chan la: ')
so_chan(ds)
```



# Chương VIII: Kiểm Thử Và Gỡ Lỗi Chương Trình

## 4. Bài tập thực hành.

Bài 1: Viết chương trình nhập vào 2 số nguyên a, b

- Tìm ước chung lớn nhất của 2 số.
- Tìm bội chung nhỏ nhất giữa 2 số.
- Xuất ra màn hình tất cả ước chung của 2 số.

*\*\* (Không sử dụng hàm có sẵn)*

Bài 2: Tìm lỗi sai trong chương trình bên và sửa lại để chương trình có thể chạy đúng.

Chương trình nhập vào danh sách các số nguyên, Xuất ra màn danh sách các phần tử có giá trị là số chẵn.

```
File Edit Format Run Options Window Help
def nhapds(a):
    n = int(input('nhap vao so phan tu: '))
    for i in range(0,n):
        print('Nhap vao phan tu thu', i+1)
        k = int(input())
        insert(i,k)
    return a
def so_chan(b)
    for i in range(0,len(b)):
        if b[i] % 2 == 0:
            print(i)

ds = []
nhapds(ds)
print('Danh sach so chan la: ')
so_chan(ds)
```

# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 1. Các bước giải bài toán trên máy tính.

Hiện nay việc giải bài toán trên máy tính được tiến hành bởi 5 bước như sau: Xác định bài toán → Lựa chọn, thiết kế thuật toán → Viết chương trình → Kiểm thử, Hiệu chỉnh → Viết tài liệu.

Xác định bài toán: mỗi bài toán được đặc tả bởi hai thành phần là Input và Output. Việc xác định bài toán chính là xác định rõ hai thành phần này và mối quan hệ giữa chúng. Các thông tin đó cần được nghiên cứu cẩn thận để có thể lựa chọn thuật toán, cách thể hiện các đại lượng đã cho, các đại lượng phát sinh trong quá trình giải bài toán và ngôn ngữ lập trình thích hợp.

Lựa chọn, thiết kế thuật toán: là bước quan trọng nhất để giải một bài toán. Mỗi thuật toán chỉ giải một bài toán nào đó, nhưng có thể có nhiều thuật toán khác nhau cùng giải một bài toán. Do đó người dùng cần thiết kế hoặc chọn một thuật toán phù hợp đã có để giải bài toán cho trước.

# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 1. Các bước giải bài toán trên máy tính.

Viết chương trình: Việc viết chương trình là tổng hợp giữa việc lựa chọn cách tổ chức dữ liệu và sử dụng ngôn ngữ lập trình để diễn đạt đúng thuật toán. Khi viết chương trình ta nên chọn một ngôn ngữ lập trình hoặc một phần mềm chuyên dụng thích hợp với thuật toán. Viết chương trình trong ngôn ngữ nào thì cần phải tuân theo đúng quy định ngữ pháp của ngôn ngữ đó.

Hiệu chỉnh là việc thử chương trình với 1 số bộ Input tương ứng với Output đã biết trước, từ đó ta xác định được các lỗi của chương trình. Mỗi bộ Input – Output được gọi là các Test.

Viết tài liệu: Tài liệu rất có ích cho người sử dụng chương trình và cho việc đề xuất những khả năng hoàn thiện thêm. Các bước có thể lặp đi lặp lại nhiều lần cho đến khi ta cho rằng chương trình đã làm việc đúng đắn và hiệu quả.

# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 2. Bài toán ví dụ.

Bài toán quản lý tiền điện

Hãy lập chương trình quản lý tiền điện nhà em. Tính tổng tiền điện trong năm, tiền điện trung bình mỗi tháng và thống kê những tháng nào sử dụng điện trên mức trung bình.

Dữ liệu vào: Dãy 12 số nguyên tương ứng với tiền điện mỗi tháng, cách nhau bởi dấu cách.

Dữ liệu ra: Tổng tiền điện trong năm  
Tiền điện trung bình mỗi tháng

Danh sách các tháng dùng điện vượt mức trung bình.

# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 2. Bài toán ví dụ.

Bài toán quản lý tiền điện

a) Xác định bài toán.

### Bài toán quản lý tiền điện

Cho dãy 12 số nguyên  $t_1, t_2, \dots, t_{12}$

*Yêu cầu:*

- Tính tổng  $s = t_1 + t_2 + \dots + t_{12}$ .
- Tính trung bình  $av = s/12$ .
- Liệt kê các  $t$  lớn hơn  $av$ .

# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 2. Bài toán ví dụ.

Bài toán quản lý tiền điện

*b) Lựa chọn và thiết kế thuật toán.*

### **Thuật toán giải bài toán quản lý tiền điện**

Bước 1: Nhập dãy số tiền điện  $t_1, t_2, \dots, t_{12}$ .

Bước 2: Khởi tạo giá trị tổng  $s$  ban đầu là 0:  $s=0$ .

Bước 3: Cộng dồn các giá trị trong dãy số vào tổng  $s$ .

Bước 4: Đưa ra giá trị  $s$ .

Bước 5: Tính giá trị trung bình  $av = s/12$ .

Bước 6: Duyệt từ  $t_1$  đến  $t_{12}$  và đưa  $i$  ra ngoài màn hình nếu  $t[i] > av$

# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 2. Bài toán ví dụ.

Bài toán quản lý tiền điện

c) *Viết chương trình.*

```
File Edit Format Run Options Window Help
t = []
for i in range(12):
    t.append(int(input()))
s=0
for i in t: s=s+i
print("số tiền điện phải trả cả năm là: ",s," ngàn đồng.")
av = s/12
print("tiền điện trung bình hàng tháng là: ",av," ngàn đồng")
print("các tháng dùng điện nhiều hơn mức trung bình:")
for i in range(12):
    if t[i]>av: print(i+1)
```

# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 2. Bài toán ví dụ.

Bài toán quản lý tiền điện

*d) Kiểm thử chương trình.*

Sau khi viết chương trình xong. Cần phải chạy thử và kiểm tra xem có lỗi hay không và sửa tất cả lỗi tìm được để hoàn thiện chương trình.

*e) Viết tài liệu.*

Viết tài liệu để hướng dẫn người dùng sử dụng, ghi chép lại mô tả thuật toán, kết quả thử nghiệm. v.v



# Chương IX: Lập Trình Và Giải Quyết Bài Toán

## 2. Bài tập.

Bài 1: Có nhất thiết phải tìm được thuật toán trước khi viết chương trình để giải bài toán đó không?

Bài 2: Trong các bước giải bài toán trên máy tính, bước nào là quan trọng nhất? Vì sao?

Bài 3: Em hãy đưa ra một vài ví dụ về bài toán thực tế mà em biết và trình bày các bước cần thực hiện bài toán đó bằng máy tính.

Bài 4: Viết chương trình quản lý chăn nuôi gà, Cho trước chi phí mua giống, chi phí chăn nuôi hàng tháng, số lượng gà nuôi, thời gian xuất chuồng. Cho biết mỗi lứa gà tốn chi phí bao nhiêu.

Gà giống (con)	Chi phí hàng tháng (mỗi 100 con)	Số lượng (con)	Thời gian
10.000 đ	3.000.000 đ	500	3.5 tháng

# Chương X: Thực Hành Tổng Hợp

Câu 1: Chọn phát biểu đúng.

- a. Khi khai báo hàm, không nhất thiết phải có tham số.
- b. Khi khai báo hàm, bắt buộc phải có tham số.
- c. Tên hàm có thể đặt tự do.

Câu 2: Câu lệnh khai báo hàm nào sau đây đúng?

- a. `function tong(a,b)`
- b. `def trung_binh_cong(a,b,c)`
- c. `def ucln(a,b):`

# Chương X: Thực Hành Tổng Hợp

Câu 3: Chọn phát biểu sai.

- a. Các câu lệnh trong hàm phải lùi vào theo đúng quy tắc trong python.
- b. Trong Python khai báo hàm bắt đầu bằng function.
- c. Tham số của hàm có thể có hoặc không.

Câu 4: Tìm lỗi sai trong đoạn code sau và sửa lại cho đúng:

```
def dien_tich_hcn(a,b)
    s = a*b
    return s
```

# Chương X: Thực Hành Tổng Hợp

Câu 5: Các tên hàm nào đúng?

- a. else
- b. dien\_tich
- c. trung bình cộng

Câu 6: Kết quả của chương trình bên là?

- a. 12
- b. 14
- c. Diện tích là: 12
- d. Diện tích là:14

```
def dien_tich_hcn(a,b):  
    s = a*b  
    return s  
c_dai = int(input("Nhập vào cạnh đáy: "))  
c_rong = int(input("Nhập vào chiều cao: "))  
print("Diện tích là: ",dien_tich_hcn(c_dai,c_rong))
```

Input:

```
Nhập vào cạnh đáy: 4  
Nhập vào chiều cao: 3
```

# Chương X: Thực Hành Tổng Hợp

Câu 7: Có bao nhiêu kiểu kết nối với hàm có sẵn.

- a. 1.
- b. 2.
- c. 3.

Câu 8: Cách kết nối với tất cả các hàm trong thư viện math là?

- a. `import math`.
- b. `include math`.
- c. `using math`.

# Chương X: Thực Hành Tổng Hợp

Câu 9: Cách kết nối đến hàm gcd trong thư viện math là:

- a. from math import gcd.
- b. using gcd.
- c. include gcd.

Câu 10: Tìm lỗi sai trong chương trình sau:

```
import math
a = int(input('a = '))
b = int(input('b = '))
print('Ước chung lớn nhất là: ',gcd(a,b))
```

# Chương X: Thực Hành Tổng Hợp

Câu 11: Kiểu xâu trong Python là gì?

- a) Là dãy các kí tự Unicode.
- b) Là một dạng số.
- c) Là kiểu dữ liệu kí tự.

Câu 12: Cách khởi tạo xâu nào là đúng trong Python?

- a) `s = xin chao`
- b) `s = "xin chao"`
- c) `<s> = <xin chao>`

# Chương X: Thực Hành Tổng Hợp

Câu 13: Để đổi số 01234 thành kiểu xâu ta dùng?

a) `print(01234)`.

b) `int(01234)`.

c) `str(01234)`.

Câu 14: Trong Python, Hàm nào dùng để in hoa tất cả chuỗi?

a) `upper()`.

b) `=UPPER()`.

c) `in_hoa()`.



# Chương X: Thực Hành Tổng Hợp

Câu 15: Cho xâu  $s = \text{"Tin Hoc Sao Viet"}$ .  $s[0]$  có giá trị là?

- a) "".
- b) 0.
- c) T.

Câu 16: Trong Python, Hàm nào dùng để in hoa chữ cái đầu mỗi từ?

- a) upper().
- b) capitalize().
- c) title().

# Chương X: Thực Hành Tổng Hợp

Câu 17: Kết quả chương trình sau là? `s2 = "Tin Hoc Sao Viet"`  
`print(s2[4])`

- a) "".
- b) 4.
- c) H.

Câu 18: Chương trình sau đúng hay sai, tại sao?

```
s = "tin hoc Sao Viet"
s[1] = "T"
s[4] = "H"
print(s)
```

# Chương X: Thực Hành Tổng Hợp

Câu 19: Câu lệnh khai báo kiểu kí tự đúng trong Python là?

a) `s = hello.`

b) `s = "xin_chao".`

c) Không có kiểu kí tự trong Python.

Câu 20: Tìm lỗi sai trong chương trình sau?

```
s = tin hoc Sao Viet
print(s)
```

# Chương X: Thực Hành Tổng Hợp

Câu 21: Kiểu danh sách trong Python là gì?

- a) Là một kiểu dữ liệu rất đa năng trong Python có khả năng chứa một danh sách các phần tử.
- b) Là xâu kí tự.
- c) Là kiểu dữ liệu kí tự.

Câu 22: Cách khởi danh sách nào là đúng trong Python?

- a) `s = []`
- b) `s = "xin chao"`
- c) `s = <xin chao>`

# Chương X: Thực Hành Tổng Hợp

Câu 23: Câu lệnh nào sau đây sai?

a) `s = [4,5,6]`

b) `s = [[2,4,6],[8,9]]`

c) `s = [a,b,c]`

Câu 24: Tìm lỗi sai trong chương trình sau?

```
danh_sach = (1,2,3,'a','b','c')  
print(danh_sach)
```

# Chương X: Thực Hành Tổng Hợp

Câu 25: Kết quả của chương trình bên là?

a) 1,2,3.

b) 6.

c) 1,2,3,a,b,c.

Câu 26: Để khai báo danh sách a và khởi tạo danh sách a có 3 phần tử 10, 20, 30, phương án nào sau đây đúng?

a)  $a = (10,20,30)$

b)  $a = \{10,20,30\}$

c)  $a = [10,20,30]$

# Chương X: Thực Hành Tổng Hợp

Câu 27: Để khai báo danh sách a rỗng, phương án nào sau đây đúng?

a) `a == []`

b) `= 0`

c) `= []`

Câu 28: Để xóa phần tử thứ 2 trong danh sách a ta dùng lệnh:

a) `del(a[1])`

b) `remove(a[2])`

c) `remove(a[1])`

# Chương X: Thực Hành Tổng Hợp

Câu 29: Cho đoạn lệnh sau: 

```
a=[7,3,8,1,9]
del(a[3])
```

Danh sách a thu được là?

- a) [7,8,1,9]
- b) [7,3,1,9]
- c) [7,3,8,9]

Câu 30: Để xóa phần tử thứ 2 trong danh sách a ta dùng lệnh:

- a) del(a[1])
- b) remove(a[2])
- c) remove(a[1])



# Chương X: Thực Hành Tổng Hợp

Câu 31: Cho đoạn lệnh sau:

```
a=[5,7,3,4,1]
a.insert(2,9)
```

Danh sách a thu được là?

- a) [5,7,9,3,4,1]
- b) [5,7,3,4,1,2,9]
- c) [2,9,5,7,3,4,1]

Câu 32: Để sắp xếp danh sách a, phương án nào sau đây đúng?

- a) a.reverse()
- b) a.sort()
- c) sort(a)

# Chương X: Thực Hành Tổng Hợp

Câu 33: Cho danh sách  $a = [1, 2, 3, 4, 5]$ . Lệnh nào dưới đây sẽ thêm số 6 vào danh sách a?

a) `a.append(6)`

b) `a.insert(6)`

c) `a.add(6)`

Câu 34: Cho danh sách  $a = [1, 2, 3, 4, 5]$ . Lệnh nào dưới đây sẽ trả về số lượng phần tử trong danh sách?

a) `len(a)`

b) `size(a)`

c) `count(a)`

# Chương X: Thực Hành Tổng Hợp

Câu 35: Các bước giải bài toán trên máy tính là:

- a) Xác định bài toán ⇨ Lựa chọn, thiết kế thuật toán ⇨ Viết chương trình ⇨ Kiểm thử, Hiệu chỉnh ⇨ Viết tài liệu.
- b) Xác định bài toán ⇨ Viết tài liệu ⇨ Lựa chọn, thiết kế thuật toán ⇨ Viết chương trình ⇨ Kiểm thử, Hiệu chỉnh.
- c) Viết tài liệu ⇨ Xác định bài toán ⇨ Lựa chọn, thiết kế thuật toán ⇨ Viết chương trình ⇨ Kiểm thử, Hiệu chỉnh.

Câu 36: Bước quan trọng nhất để giải bài toán trên máy tính là:

- a) Xác định bài toán
- b) Lựa chọn, thiết kế thuật toán
- c) Viết chương trình

# Chương X: Thực Hành Tổng Hợp

Câu 37: Điền vào “...” hoàn thành phát biểu sau: “Chương trình chạy không lỗi ngoại lệ nhưng kết quả đưa ra .... Đây là lỗi ... bên trong chương trình.”

- a) sai, logic.
- b) sai, Syntax Error.
- c) đúng, logic

Câu 38: Chương trình sau thông báo lỗi gì?

```
for i in range(10) print(i)
```

- a) Type Error.
- b) NameError.
- c) SyntaxError.

# Chương X: Thực Hành Tổng Hợp

Câu 39: Khi dòng lệnh thụt vào không thẳng hàng chương trình đưa ra mã lỗi ngoại lệ nào?

- a) ZeroDivisionError.
- b) TypeError.
- c) IndentationError.

Câu 40: Chương trình sau thông báo lỗi gì?

```
n = 5
for i in range(n):
    prin(t)
```

- a) Type Error.
- b) NameError.
- c) SyntaxError.

# Chương X: Thực Hành Tổng Hợp

Câu 41: Dòng lệnh sau bị lỗi gì? `A = [1, 2]`  
`A[0.5]`

- a) Lỗi logic.
- b) Lỗi ngoại lệ.
- c) Lỗi cú pháp.

Câu 42: Chọn phát biểu sai trong các phát biểu sau:

- a) Khi có lỗi sai cú pháp, chương trình lập tức dừng và thông báo lỗi.
- b) Lỗi ngoại lệ là lỗi không thể thực hiện một lệnh trong chương trình.
- c) Cách xử lý các loại lỗi giống nhau.

# Chương X: Thực Hành Tổng Hợp

Câu 43: Chương trình sau thông báo lỗi gì?

```
lst = [1, 2, 3, 4, 5]
for i in range(10):
    print(lst[i])
```

- a) IndexError.
- b) NameError.
- c) SyntaxError.

Câu 44: Chọn phát biểu sai trong các phát biểu sau:

- a) Chương trình chạy khi lỗi logic xảy ra
- b) Khi có lỗi sai cấu trúc ngôn ngữ, chương trình vẫn chạy.
- c) Khi có lỗi ngoại lệ, chương trình dừng và thông báo lỗi.

# Chương X: Thực Hành Tổng Hợp

Câu 45: Viết chương trình nhập vào số nguyên  $n$ ,

- Xuất ra màn hình các số chẵn từ 1 đến  $n$ .
- Xuất ra màn hình các số lẻ từ  $-n$  đến 1.
- Xuất ra màn hình các số nguyên tố từ 1 đến  $n$ .
- Xuất ra màn hình dãy  $n$  số fibonacci.

Câu 46: Viết chương trình nhập vào điểm trung bình các môn: Toán, Anh, Văn, Lịch Sử, Địa Lý, GDCD, Vật Lý, Hóa Học.

Hãy xếp loại học sinh. Các môn Anh, Toán, Văn hệ số x2.

<b>Điểm TB</b>	<b><math>\geq 8</math></b>	<b><math>\geq 6.5</math></b>	<b><math>\geq 5</math></b>	<b><math>&lt; 5</math></b>
<b>Xếp Loại</b>	Giỏi	Khá	TB	Yếu



# Chương X: Thực Hành Tổng Hợp

Câu 47: Viết chương trình nhập vào danh sách n số nguyên.

- Đếm xem có bao nhiêu số chia hết cho 3 trong dãy số.
- Tính độ dài của danh sách, Gán vào biến `do_dai` và đưa ra màn hình.
- Lấy ra phần tử nằm giữa danh sách. (lấy ra 2 nếu danh sách số phần tử chẵn)
- Tính trung bình cộng các phần tử và thêm nó vào đầu danh sách. xuất danh sách ra màn hình.
- Đưa ra màn hình tổng các số ở vị trí chẵn.
- Đưa ra màn hình các số nguyên tố có trong danh sách.
- Tính `m`, Biết `m = Trung bình cộng các phần tử chia cho "do_dai"`. Đưa `m` vào cuối danh sách và xuất danh sách ra màn hình.

# Chương X: Thực Hành Tổng Hợp

Câu 47: Viết chương trình nhập vào danh sách n số nguyên.

- Đếm xem có bao nhiêu số chia hết cho 3 trong dãy số.
- Tính độ dài của danh sách, Gán vào biến `do_dai` và đưa ra màn hình.
- Lấy ra phần tử nằm giữa danh sách. (lấy ra 2 nếu danh sách số phần tử chẵn)
- Tính trung bình cộng các phần tử và thêm nó vào đầu danh sách. xuất danh sách ra màn hình.
- Đưa ra màn hình tổng các số ở vị trí chẵn.
- Đưa ra màn hình các số nguyên tố có trong danh sách.
- Tính m, Biết  $m = \text{Trung bình cộng các phần tử chia cho "do\_dai"}$ . Đưa m vào cuối danh sách và xuất danh sách ra màn hình.

# Chương X: Thực Hành Tổng Hợp

Câu 48: Gỡ lại chương trình sau và tìm lỗi trong chương trình, sửa lỗi để có thể hoạt động chính xác.

```
File Edit Format Run Options Window Help
t = 0
for i in range(12):
    t.append(input())
s=0
for i in t do s=s+i
print("số tiền điện phải trả cả năm là: ",s," ngàn đồng.")
av = s/12
print("tiền điện trung bình hàng tháng là: ",av," ngàn đồng")
print("các tháng dùng điện nhiều hơn mức trung bình:")
for i in range(12):
    if t[i]>av then print(i+1)
```

# Chương X: Thực Hành Tổng Hợp

Câu 49: Hãy giải quyết bài toán quản lý tiền điện nhà em trên máy tính.

- Dữ liệu vào là số trên đồng hồ điện từ tháng 1 đến tháng 12.
- Tính tiền điện từng tháng dựa vào bảng Giá bán lẻ điện sinh hoạt.
- Tính tổng tiền điện trong năm.
- Tính số kWh trung bình mỗi tháng.
- Thống kê các tháng dùng điện vượt quá mức trung bình

Tháng	1	2	3	4	5	6	7	8	9	10	11	12
Số đồng hồ	7027	7132	7288	7397	7992	8122	8347	7463	8577	8689	8799	8912

Giá bán lẻ điện sinh hoạt	
Bậc 1: Cho kWh từ 0 - 50	1.728
Bậc 2: Cho kWh từ 51 - 100	1.786
Bậc 3: Cho kWh từ 101 - 200	2.074
Bậc 4: Cho kWh từ 201 - 300	2.612
Bậc 5: Cho kWh từ 301 - 400	2.919
Bậc 6: Cho kWh từ 401 trở lên	3.015

# Chương X: Thực Hành Tổng Hợp

Câu 50: Em hãy giải quyết bài toán sau trên máy tính: Một quả táo rơi từ trên cây cách mặt đất  $s$  (mét), có trọng lượng  $m$  (gram). biết gia tốc trọng trường là  $9.8\text{m/s}^2$ . Tính động năng khi quả táo chạm đất.

- Động năng (kinetic energy) của quả táo khi chạm đất được tính bằng công thức:

$$K = 0.5 * m * v^2$$

Trong đó:

- $K$  là động năng của quả táo.
- $m$  là khối lượng của quả táo, được chuyển đổi thành đơn vị kilogram để tính toán.
- $v$  là vận tốc của quả táo khi chạm đất, đã tính được ở câu trước,  $v \approx 8.85 \text{ m/s}$ .

- Để tính vận tốc của quả táo khi chạm đất, ta có thể sử dụng công thức vật lý:

$$v^2 = u^2 + 2as$$

Trong đó:

- $v$  là vận tốc của quả táo khi chạm đất (để tìm).
- $u$  là vận tốc ban đầu của quả táo khi nó rơi từ trên cây xuống đất, trong trường hợp này  $u = 0 \text{ m/s}$  vì quả táo chỉ có khối lượng và không có vận tốc ban đầu theo phương dọc (trong chiều rơi xuống).
- $a$  là gia tốc trọng trường,  $a = 9.8 \text{ m/s}^2$ .
- $s$  là khoảng cách từ quả táo xuống mặt đất.