**BỘ ĐỀ ÔN TẬP LẬP TRÌNH CƠ BẢN VỚI PYTHON**

**Bộ đề 01:**

**Câu 1:** Viết chương trình nhập số tự nhiên n từ bàn phím, sau đó kiểm tra xem số n có phải là số nguyên tố hay không, thông báo trên màn hình.

*(Gợi ý: Số nguyên tố là số chỉ chia hết cho 1 và chính nó)*

**Câu 2:** Viết chương trình nhập từ bàn phím số tự nhiên ss tính theo giây, sau đó đổi số giây này ra thành giờ, phút, giây và thông báo trên màn hình. Ví dụ nhập ss=3740 thì kết quả hiển thị như sau:

3740 giây=1 giờ 2 phút 20 giây

*(Gợi ý: Sử dụng hàm phép tính nâng cao : hàm divmod()*

* *Ý nghĩa hàm divmod(): Như ta đã biết có phép // và % trong python. Thì // tương ứng là div, còn % tương ứng là mod. Vậy hàm divmod() thay thế cho cả 2 phép tính trên.*
* *Công thức: divmod(x,y)=x//y, x%y*
* *Ví dụ: divmod(12.5, 2) sẽ cho kết quả là: (6.0, 0.5)*

**Câu 3:** Viết chương trình nhập số tự nhiên n và in ra trên màn hình tất cả các ước số thực sự của n

*(Gợi ý: Sử dụng for và range(start,stop). Ước số của n là những số n chia hết cho số đó)*

**Câu 4:** Viết chương trình nhập số tự nhiên n và đếm số các ước số thực sự của n. In kết quả ra màn hình

**Câu 5:** Viết chương trình nhập số tự nhiên n và tính các giá trị sau:

1. Tính tổng 1+2+…+2n
2. Tính tổng các số tự nhiên <n và là số lẻ
3. Tính tổng các số tự nhiên <n và là số chẵn

*(Gợi ý: Câu a,b,c chỉ trong 1 chương trình)*

**Câu 6:** Viết chương trình nhập số tự nhiên n và tính các giá trị sau:

1. Tính tổng các ước số thực sự của n (không tính n)
2. Tính tổng 12+22+22+…+n2

**Câu 7:** Viết chương trình nhập số tự nhiên n từ bàn phím. Tính và kiểm tra tính đúng đắn của biểu thức sau:

12+22+22+…+n2=

*(Gợi ý: tính từ bên biểu thức và đưa ra 2 kết quả có giống nhau không?)*

**Câu 8:** Viết chương trình nhập số tự nhiên n từ bàn phím. Tính và kiểm tra tính đúng đắn của biểu thức sau:

13+23+23+…+n3=(1+2+3+…+n)2

**Câu 9:** Số tự nhiên được gọi là hoàn hảo nếu số đó bằng tổng các ước số thực sự của mình. Ví dụ 6=1+2+3 vậy 6 là số hoàn hảo. Viết chương trình nhập số tự nhiên n từ bàn phím và kiểm tra xem n có phải là số hoàn hảo hay không. Thông báo trên màn hình

**Câu 10:** Viết chương trình nhập số tự nhiên n từ bàn phím và in ra tất cả các số hoàn hảo <n.

**Bộ đề 02:**

**Câu 1:** Sau khi thực hiện chương trình sau thì **kq** là bao nhiêu?

kq=1

def calc(x):

kq=4\*x-1

return kq

calc(7)

**Câu 2:** Đoạn chương trình sau sẽ in ra kết quả bao nhiêu:

def f(x):

return x\*\*2, x\*\*x\*\*x

x,y=f(2)

print(x+y)

**Câu 3:** Viết các hàm **lambda** mô tả các biểu thức sau:

1. x+y+z
2. x\*\*(y+z)
3. (x+y)\*\*2

**Câu 4:** Viết hàm số f(n) với tham số n là số tự nhiên theo yêu cầu sau:

1. Hàm trả lại số các ước số thực sự của n
2. Hàm trả lại 1 nếu n là nguyên tố, ngược lại hàm trả về 0
3. Hàm đếm số các ước số là lẻ của n
4. Hàm đếm số các ước số nguyên tố của n
5. Hàm tính tổng tất cả các ước số thực sự của n

**Câu 5:** Viết hàm số f(m,n) với tham số m,n là số tự nhiên theo yêu cầu sau:

1. Hàm đếm số các ước chung của các số m,n
2. Hàm đếm số các ước số của m nhưng không là ước số của n
3. Hàm tính tổng các ước số chung của m và n

**Bộ đề 03:**

**Câu 1:** Thiết lập lớp **QuadriSquare** bao gồm đối tượng hình chữ nhật hoặc hình vuông.

* Mỗi đối tượng sẽ có 2 thuộc tính chiều dài w và chiều cao h
* Các method chính của lớp này là các hàm tính chu vi (Chuvi()) và diện tích (Dientich()) của mỗi đối tượng

**Bài mẫu cho các em tham khảo:**

**class QuadriSquare**:

**def \_\_init\_\_**(self, w, h):

self.w = w

self.h = h

**def dientich**(self):

**return** self.w \* self.h

**def chuvi**(self):

r**eturn** (self.w + self.h) \*2

**Câu 2:** Thiết lập lớp **Hocsinh** dùng để quản lý các đối tượng học sinh trong lớp em

Các thuộc tính của mỗi học sinh bao gồm:

* Họ và tên
* Địa chỉ
* Chiều cao
* Cân nặng
* Học lực

Lớp **Hocsinh** được thiết lập các phương thức sau:

* Hàm \_\_init\_\_() thiết lập tên học sinh, địa chỉ, chiều cao và cân nặng
* Hàm cập nhật địa chỉ mới
* Hàm cập nhật chiều cao, cân nặng mới
* Hàm cập nhật học lực học sinh

**Câu 3:** Thiết lập lớp **Triple** bao gồm các đối tượng là một bộ 3 số thực (a,b,c). Thuộc tính của mỗi đối tượng bao gồm:

* Tên của đối tượng
* Bộ 3 số a,b,c
* Tổng 3 số này, ký hiệu S

Thiết lập các phương thức sau:

* Hàm \_\_init\_\_() dùng để khởi tạo 1 bộ triple
* Hàm Update() dùng để cập nhật giá trị S
* Hàm Change() có chức năng thay đổi các giá trị a,b,c và đồng thời cập nhật S
* Hàm isTriange() kiểm tra xem bộ 3 số này có tạo thành một tam giác hay không?

**Câu 4:** Thiết lập lớp **Tam\_giac** là lớp con của lớp **Triple** trong bài tập trên. Bổ sung các phương thức sau cho lớp con này:

* Hàm tính diện tích tam giác
* Hàm tính bán kính vòng tròn ngoại tiếp tam giác
* Hàm tính bán kính vòng tròn nội tiếp tam giác

*Ta có các công thức sau tính diện tích S, bán kính vòng tròn ngoại tiếp R và bán kính vòng tròn nội tiếp r của tam giác có các cạnh a,b,c.*

*với p = nửa chu vi tam giác*

***Gợi ý đoạn đầu:***

**class *Tam\_giac*(Triple):**

**def *getpchuvi***(self):

**return** self.S/2

**def** ………

**Câu 5:** Thiết lập lớp Circle (vòng tròn) bao gồm thuộc tính R (bán kính) và tọa độ âm X,Y. Thiết lập các phương thức tính chu vi và diện tích của đối tượng hình tròn.

**Bộ đề 04:**

**Câu 1:** Viết chương trình tạo list gồm 10 số Fibonacci đầu tiên.

**Câu 2:** Viết chương trình tạo list bao gồm N số nguyên tố đầu tiên với N số tự nhiên cho trước (hoặc được nhập từ bàn phím)

**Câu 3:** Viết chương trình nhập số N, sau đó sẽ lần lượt nhập N tên học sinh, đưa các học sinh này vào một list và in list này ra màn hình

**Câu 4:** Viết chương trình nhập số N từ bàn phím, in và liệt kê danh sách các ước số thực của n như một list

Ví dụ: Nếu N=15 thì kết quả list là [1,3,5]

**Câu 5(nâng cao):** Viết chương trình nhập thực hiện các công việc sau:

* Nhập số tự nhiên N từ bàn phím
* Nhập lần lượt N số tự nhiên thành dãy a1,a2,…,aN. Giả sử dãy này gọi là A
* Đưa ra dãy con b1,b2,…bK của dãy A gồm các phần từ
* Gợi ý: *Thực hiện theo ý tưởng của sàng nguyên tố Eratosthenes*

**Câu 6:** Nhập dãy A bao gồm n phần tử từ bàn phím và tính tổng các phần tử của A

**Câu 7:** Viết chương trình nhập số tự nhiên n từ bàn phím và tính các giá trị sau: (dùng hàm)

a) Tính tổng 12+22+32+…+n2

b) Tính tổng 1+1/2+1/3+…+1/n

c) Tính tổng

**Câu 8:** Cho trước 2 dãy A,B. Dãy C được thiết lập bằng cách lần lượt lấy các phần tử từ A, B đưa vào C theo thứ tự tăng dần.

Ví dụ:

A=[0.2.4.6.8.10]

B=[1,3,5,7,9,11]

Thì C=[0,1,2,3,4,5,6,7,8,910,11]

**Câu 9:** Cho trước dãy A. Dãy B được thiết lập bằng cách tách các phần tử là số (số nguyên hoặc số thực) từ dãy A.

Ví dụ:

A=[1.2, “one”, 0.15, “A”, “B”, “C”, 1, 50, 4.56, “C”]

Thì B=[1.2, 0.15, 1, 50, 4.56]

***(Gợi ý: Có 2 giải thuật cơ bản: Giải thuật sử dụng type và giải thuật dùng hàm isinstance() )***

**Câu 10:** Viết chương trình nhập một xây ký tự từ bàn phím và in ra màn hình là các ký tự của xâu này.

Ví dụ: Việt Nam 🡺 [“V”, “i”,….]

**Câu 11:** Viết chương trình nhập số tự nhiên n và thiết lập:

* Dãy A bao gồm các số tự nhiên lẻ nhỏ hơn n
* Dãy B bao gồm các số tự nhiên chẵn nhỏ hơn n

**Câu 12:** Cho trước 2 dãy số A,B. Dãy C được tạo theo quy tắc sau:

Giả sử số phần tử của A là n và nhỏ hơn số phần tử của B. Trường hợp ngược lại làm tương tự.

* n phần tử đầu tiên của C là tổng các số hạng tương ứng của A,B
* Các phần tử còn lại lấy từ B

Viết thủ tục với 2 tham số đầu vào là A,B và in ra màn hình dãy C

**(Gợi ý: Dùng hàm randint() để sinh bộ dữ liệu ngẫu nhiên và dùng các phương thức copy)**

**Bộ đề 05:**

**Câu 1:** Viết các hàm số với đầu vào là dãy A (bao gồm số và xây ký tự)

a) Tính tổng các phần tử trong dãy A

b) Đếm số các phần tử là xây kỹ tự từ dãy A

c) Đưa ra kết quả là 2 dãy B,C. Dãy B bao gồm các phần tử là số của A, dãy C bao gồm các phần tử là xâu ký tự của A

\* Gợi ý: Thiết lập một số hàm

**Câu 2:** Viết một hàm số thay thế cho phương thức **sort()** của dãy list. Tham số đầu vào là dãy A bao gồm các phần tử là số (nguyên hay thực). Hàm trả lại là một dãy bao gồm các số đã được sắp xếp theo thứ tự tăng dần các phần tử từ A.

**Câu 3:** Viết một hàm số tổng quát hơn phương thức sort() của dãy (list) như sau:

- Đầu vào là dãy A bao gồm có thể là số hoặc ký tự

- Đầu ra là dãy B thu được từ dãy A bằng cách sắp xếp các phần tử là số theo thứ tự tăng dần, còn các vị trí là ký tự thì giữ nguyên

**Câu 4:** Cho trước 2 dãy(list) A,B và số tự nhiên k, 0<=k<len(A). Viết hàm số trả lại dãy (list) thu được từ A sau khi chèn dãy B vào vị trí của chỉ số k

**Câu 5:** Cho trước dãy(list) A bao gồm các số nguyên, cho trước số nguyên k. Biết rằng dãy(list) trong Python có 2 phương thức **count()** và **remove()** có ý nghĩa như sau:

a.count(<giá trị>) sẽ trả lại số lần tìm thấy của <giá trị> trong a, nếu không tìm thấy sẽ trả lại 0

a.remove(<giá trị>) sẽ xóa phần tử đầu tiên có <giá trị> trong a, nếu không tìm thấy phần tử có <giá trị> sẽ báo lỗi

Viết chương trình xóa tất cả các phần tử trong dãy a có giá trị k. Chú ý khi viết chương trình không được để gặp lỗi

**Câu 6:** Cho trước dãy(list) a bao gồm các số nguyên, cho trước 2 số nguyên k,m. Viết chương trình thay thế tất cả các phần tử của a có giá trị k bằng giá trị m.

Gợi ý: Sử dụng phương thức **index()** của list trong Python

**Câu 7:** Cho trước 2 dãy(list) a,b. Viết chương trình thiết lập dãy c bao gồm các phần tử lấy lần lượt từ phần cuối của a,b.

Gợi ý: sử dụng PT **pop()**

**Câu 8:** Cho trước dãy(list) A. Viết chương trình tạo dãy B bao gồm các phần tử lấy từ A nhưng theo thứ tự ngược lại. Chú ý giữ nguyên A không thay đổi

Gợi ý: Có nhiều cách giải

**Câu 9:** Cho trước 2 dãy A,B có số phần tử bằng nhau và đều là các số tự nhiên. Viết chương trình tạo dãy C cũng có số phần tử bằng A, B và các phần tử của C là ước số chung lớn nhất của các phần tử tương ứng của A,B.