HƯỚNG DẪN CHI TIẾT

1. Viết một chương trình tạo list chứa các giá trị bình phương của các số từ 1 đến 20 (bao gồm cả 1 và 20) và in 5 mục đầu tiên trong list.

Gợi ý:

Sử dụng toán tử ** để lấy giá trị bình phương.

Sử dụng range() cho vòng lặp.

Sử dụng list.append() để thêm giá trị vào list.

```
1  li=list()
2  for i in range(1,21):
3      li.append(i**2)
4  print('05 giá trị bình phương đầu tiên của list là: ',end='')
5  for i in range(1,6):
6      print(li[i-1],end=' ')
```

Kết quả chạy chương trình:

05 giá trị bình phương đầu tiên của list là: 1 4 9 16 25

2. Viết một chương trình trong có thể tạo list chứa giá trị bình phương của các số từ 1 đến 20 (bao gồm cả 1 và 20). Sau đó in tất cả các giá trị của list, trừ 5 mục đầu tiên.

```
1  li=list()
2  for i in range(1,21):
3      li.append(i**2)
4  print('Các giá trị của list trừ năm mục đầu tiên là: ',end='')
5  print(li[5:])
```

Kết quả chạy chương trình:

Các giá trị của list trừ năm mục đầu tiên là:

[36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]

1. Viết chương trình Python có thể lọc các số chẵn trong danh sách sử dụng hàm filter. Danh sách là [1,2,3,4,5,6,7,8,9,10].

Gợi ý:

Sử dụng filter() để lọc các yếu tố trong một list.

Sử dụng lambda để định nghĩa hàm chưa biết.

```
1  li_goc=[1,2,3,4,5,6,7,8,9,10]
2  li_so_chan=list(filter(lambda x:x%2==0,li_goc))
3  print(li_so_chan)
```

Kết quả chạy chương trình:

[2, 4, 6, 8, 10]

2. Tạo một list, có 100 số tự nhiên đầu tiên. Tính tổng của chúng và đưa kết quả ra màn hình

```
1  li = []
2  for i in range(1,101):
3      li.append(i)
4  sum = 0
5  for i in li:
6      sum+=i
7  print("Tổng của 100 số tự nhiên đâu tiên: %d"%sum)
```

Kết quả chạy chương trình:

Tổng của 100 số tự nhiên đâu tiên: 5050

3. Nhập từ bàn phím 1 số tự nhiên n. Tạo một list có n phần tử từ 1 đến n. Tìm và in ra màn hinh các số nguyên tố thuộc dãy đã cho

```
1  n = int(input("Nhập giá trị n ="))
2  li = []
3  for i in range(1,n+1):
4      li.append(i)
5  def check_so_nguyen_to(x):
6      flag = True
7      if x < 2 : flag = False
8      else:
9          for i in range(2,x):
10             if x %i == 0:
11                 flag = False
12                 break
13
14
15
```

```
    return flag
print("Danh sách các số nguyên tố thuộc li là: ")
for j in li:
    if check_so_nguyen_to(j):
        print(j, end =' ')pos = str.rfind(" ")
print("Tên sinh viên là:",str[pos+1:])
```

1. Viết chương trình nhập họ tên một sinh viên từ bàn phím và trả về (in ra màn hình) tên của sinh viên đó.

Hướng dẫn:

```
hoten=input('Nhập họ tên sinh viên: ')
lst=hoten.strip().split(" ")    #hàm split(" ") chuyển chuỗi hoten thành list lst
ten=lst[len(lst)-1]
print("Tên sinh viên là:",ten)
```

Kết quả thực hiện chương trình:

```
Nhập họ tên sinh viên: Nguyễn Văn Minh
Tên sinh viên là: Minh
```

2. Viết chương trình trò chơi ra búa, kéo, lá chơi với máy tính. Người chơi có 5 lượt chơi với máy tính, sau 5 lượt chơi thì thống kê người chơi đã thắng, hòa, thua bao nhiêu lượt với máy tính.

   Gợi ý:

   Cách 1: Nhập số từ bàn phím

```
import random
ngươi_choi = int(input("Búa(0), Lá(1),Kéo(2) "))
computer = random.randint(0, 2)
```

   Cách 2: Nhập chữ từ bàn phím

```
import random
nguoi_choi = ["Búa","Lá","Kéo"]
computer = random.choice(options)
```

Hướng dẫn cách 1:

```
import random
```

```
print("CHÀO MỪNG BẠN ĐẾN VỚI TRÒ CHƠI: ra Búa, Lá, Kéo !")
print("Bạn có 5 lượt chơi với computer")
print('Xin mời bạn lựa chọn ra: Búa/Lá/Kéo??')
print("Giá trị 0 đại diện cho ra búa ")
print("Giá trị 1 đại diện cho ra kéo ")
print("Giá trị 2 đại diện cho ra lá ")
print('---------------------------')
# lưu kết quả trò chơi
thang = 0            #Thắng
hoa = 0              #Hòa
thua = 0             #Thua

# Rock, Paper, Scissors game
for i in range(5):
  computer = random.randint(0, 2)
  print("ván thứ ",i+1,'mời bạn chọn ra ',end='')
  nguoi_choi = input("Búa (0), Kéo(1), Lá (2): ")
  if computer==0:
    print('Computer đã chọn ra búa')
  elif computer==1:
    print('Computer đã chọn ra kéo')
  else:
    print('Computer đã chọn ra lá')
  if nguoi_choi not in ['0', '1', '2']:
    print("Sai luật, Mời nhập lại")
    continue

  nguoi_choi = int(nguoi_choi)
  if nguoi_choi==0:
    kq="búa"
```

```
  elif nguoi_choi==1:
    kq="kéo"
  else:kq="lá"
  if computer == nguoi_choi:
    print("Ván này bạn đã ra ", kq, "nên bạn hòa Computer")
    hoa += 1
  elif (computer == 0 and nguoi_choi == 1) or (computer == 1 and nguoi_choi == 2) or (computer == 2 and nguoi_choi == 0):
    print("Bạn đã ra ",kq,"nên bạn đã thua Computer!")
    thua += 1
  else:
    print("Bạn đã ra",kq, "nên bạn đã thắng Computer!")
    thang += 1
print("-----------------------------------")
print('Tổng kết trò chơi')
print("Bạn đã thắng Computer", thang, "lần !")
print("Bạn đã hòa Computer", hoa, "lần !")
print("Bạn đã thua Computer", thua, "lần !")
```

Kết quả thực hiện chương trình:

```
CHÀO MỪNG BẠN ĐẾN VỚI TRÒ CHƠI: ra Búa, Lá, Kéo !
Bạn có 5 lượt chơi với computer
Xin mời bạn lựa chọn ra: Búa/Lá/Kéo??
Giá trị 0 đại diện cho ra búa
Giá trị 1 đại diện cho ra kéo
Giá trị 2 đại diện cho ra lá
---------------------------
ván thứ  1 mời bạn chọn ra Búa (0), Kéo(1), Lá (2): 1
Computer đã chọn ra búa
Bạn đã ra  kéo nên bạn đã thua Computer!
ván thứ  2 mời bạn chọn ra Búa (0), Kéo(1), Lá (2): 0
```

```
Computer đã chọn ra búa
Ván này bạn đã ra  búa nên bạn hòa Computer
ván thứ  3 mời bạn chọn ra Búa (0), Kéo(1), Lá (2): 2
Computer đã chọn ra búa
Bạn đã ra lá nên bạn đã thắng Computer!
ván thứ  4 mời bạn chọn ra Búa (0), Kéo(1), Lá (2): 1
Computer đã chọn ra lá
Bạn đã ra kéo nên bạn đã thắng Computer!
ván thứ  5 mời bạn chọn ra Búa (0), Kéo(1), Lá (2): 1
Computer đã chọn ra lá
Bạn đã ra kéo nên bạn đã thắng Computer!
-------------------------------------
Tổng kết trò chơi
Bạn đã thắng Computer 3 lần !
Bạn đã hòa Computer 1 lần !
Bạn đã thua Computer 1 lần !
```

4. Viết chương trình chuẩn hóa một chuỗi ký tự được nhập từ bàn phím. Chuỗi chuẩn hóa là chuỗi không có các khoảng trăng(dấu cách) ở đầu và cuối, các từ trong chuỗi chỉ cách nhau đúng một dấu cách.

Hướng dẫn:

Cách 1

```
1  input_string = input("Nhập chuỗi ký tự: ")
2
3
4  #Loại bỏ các khoảng trắn ở đầu và cuối chuỗi ký tự
5  while len(input_string) > 0 and input_string[0] == " ":
6      input_string = input_string[1:]
7  while len(input_string) > 0 and input_string[-1] == " ":
8      input_string = input_string[:-1]
9
10
11 # Chuyển tất cả các ký tự trong chuỗi sang chữ thường
12 input_string = input_string.lower()
```

```
13
14
15
16
17
18
19
```

```
    #Thay thế nhiều khoảng trắng thừa ở giữa các từ bằng một khoảng trắng
    normalized_string = ""
    for i in range(len(input_string)):
        if input_string[i] != " " or (input_string[i] == " " and\
                                      input_string[i-1] != " "):
            normalized_string += input_string[i]
    # Print the normalized string
    print("Chuỗi ký tự sau khi được chuẩn hóa là:", normalized_string)
```

Kết quả thực hiện chương trình:

```
Nhập chuỗi ký tự:   Đại    học  kinh tế   kỹ   thuật
Chuỗi ký tự sau khi được chuẩn hóa là: đại học kinh tế kỹ thuật
```

Cách 2. Sử dụng các hàm strip() để loại bỏ khoảng trắng đầu và cuối của chuỗi, lower() chuyển tất cả các ký tự sang chữ thường, hàm split() tách chuỗi thành một danh sách các từ, cuối cùng dùng hàm join() nối các từ trong danh sách lại với nhau bằng một dấu cách,

```
1
2
3
4
5
6
7
8
9
10
11
12
13
```

```
    input_string = input("Nhập vào một chuỗi ký tự: ")

    # Dùng hàm strip() để loại bỏ khoảng trắng ở đầu và cuỗi chuỗi
    input_string = input_string.strip()

    # Chuyển tất cả các ký tự sang chữ thường
    input_string = input_string.lower()

    # "Thay thế các khoảng trắng bằng một khoảng trắng duy nhất"
    input_string = " ".join(input_string.split())

    # In ra màn hình chuỗi sau khi đa chuẩn hóa
    print("Normalized string:", input_string)
```

5. Viết chương trình tạo một list với các giá trị nhập từ bàn phím, việc nhập liệu kết thúc khi gõ phím ‘q’.In các giá trị đã nhập ra màn hình.

Hướng dẫn:

```
1  list_values = []
2
3  while True:
4      value = input("Nhập giá trị (nhập 'q' để dừng): ")
5      if value == 'q':
6          break
7      list_values.append(value)
8
9  print("Danh sách các giá trị:", list_values)
```

Kết quả thực hiện chương trình:

```
Nhập giá trị (nhập 'q' để dừng): 4
Nhập giá trị (nhập 'q' để dừng): c
Nhập giá trị (nhập 'q' để dừng): 7
Nhập giá trị (nhập 'q' để dừng): 3
Nhập giá trị (nhập 'q' để dừng): 2
Nhập giá trị (nhập 'q' để dừng): q
Danh sách các giá trị: ['4', 'c', '7', '3', '2']
```

6. Viết chương trình tạo một list với các giá trị là số nguyên nhập từ bàn phím, việc nhập liệu kết thúc khi gõ phím ‘0’. Thực hiện các công việc sau đây:

a. Tìm kiếm phần tử có giá trị x trong list với x được nhập từ bàn phím, nếu có hãy in ra vị trí xác định của x trong list.

b. Chỉnh sửa lại giá trị của một phần tử x được tìm thấy ở câu a, in danh sách sau khi chỉnh sửa ra màn hình.

c. Thêm một phần tử có giá trị là y với y được nhập từ bàn phím (chú ý: y có thể thêm vào đầu, cuối hoặc giữa list)

d. Không sử dụng hàm sort(), sắp xếp các phần tử trong list theo giá trị giảm dần. In ra màn hình danh sách trước và sau khi săp xếp.

e. Xóa(loại bỏ) một phần tử bất kỳ trong list. In ra màn hình danh sách list trước và sau khi loại bỏ.

Hướng dẫn: Trong bài này, chúng ta thực hiện các thuật toán cơ bản nhập dữ liệu, tìm kiến tuyến tính, chỉnh sửa/cập nhật/ thêm mới, sắp xếp, xóa phần tử trên cấu trúc list mà không sử dụng các hàm : sorted(), remove(). Để giúp sinh viên nắm vững giải thuật cơ bản

Sinh viên và giảng viên có thể sử dụng các hàm đã xây dựng sẵn trong python để giải bài tập theo các phương pháp đơn giản hơn.

```
    numbers = []

    while True:
        number = int(input("Nhập vào một số nguyên (nhập 0 để kết thúc nhập): "))
        if number == 0:
            break
        numbers.append(number)

    print("Danh sách số nguyên ban đầu:", numbers)

    #a. Tìm kiếm
    search_number = int(input("Nhập vào số cần tìm: "))
    if search_number in numbers:
        print("Tìm thấy số", search_number, "tại vị trí", numbers.index(search_number) + 1)
    else:
        print("Không tìm thấy số", search_number, "trong danh sách.")

    #b. Chỉnh sửa
    new_value = int(input("Nhập vào giá trị mới: "))
    if search_number in numbers:
        index = numbers.index(search_number)
        numbers[index] = new_value
        print("Danh sách sau khi chỉnh sửa:", numbers)
```

```
26
27      #c.Thêm phần tử
28      new_number = int(input("Nhập vào số nguyên mới: "))
29      position = int(input("Nhập vào vị trí muốn thêm (1 - đầu, 2 - giữa, 3 -
30  cuối): "))
31      if position == 1:
32          numbers.insert(0, new_number)
33      elif position == 2:
34          numbers.insert(len(numbers)//2, new_number)
35      else:
36          numbers.append(new_number)
37      print("Danh sách sau khi thêm:", numbers)
38
39      #d. Sắp xếp
40      for i in range(len(numbers)):
41          for j in range(i+1, len(numbers)):
42              if numbers[i] < numbers[j]:
43                  numbers[i], numbers[j] = numbers[j], numbers[i]
44      print("Danh sách sau khi sắp xếp:", numbers)
45
46      #e.Xóa phần tử
47      y = int(input("Nhập phần tử cần xóa?"))
48      for i in range(len(numbers)):
49          if numbers[i] == y:
50              #Di chuyển các phần tử sau phần tử cần xóa y về trước
51              for j in range(i, len(numbers)-1):
52                  numbers[j] = numbers[j+1]
53              #n -= 1
54              break
56
```

```
57      print("Trước khi xóa",numbers)
58      # Giảm list về kích thước mới
59      numbers = numbers[:len(numbers)-1]
60
61      # In list sau khi xóa
        print('Sau khi xóa: ',numbers)
```

Kết quả thực hiện chương trình

```
Nhập vào một số nguyên (nhập 0 để kết thúc nhập): 1
Nhập vào một số nguyên (nhập 0 để kết thúc nhập): 0
Danh sách số nguyên ban đầu: [5, 3, 8, 9, 2, 1]
Nhập vào số cần tìm: 2
Tìm thấy số 2 tại vị trí 5
Nhập vào giá trị mới: 7
Danh sách sau khi chỉnh sửa: [5, 3, 8, 9, 7, 1]
Nhập vào số nguyên mới: 6
Nhập vào vị trí muốn thêm (1 - đầu, 2 - giữa, 3 - cuối): 2
Danh sách sau khi thêm: [5, 3, 8, 6, 9, 7, 1]
Danh sách sau khi sắp xếp: [9, 8, 7, 6, 5, 3, 1]
Nhập phần tử cần xóa?5
Trước khi xóa [9, 8, 7, 6, 3, 1, 1]
Sau khi xóa:  [9, 8, 7, 6, 3, 1]
```

7. Viết chương trình sinh dãy (list)A là biểu diễn của ma trận đơn vị. Chương trình nhập số n từ bàn phím và sinh ra ma trận đơn vị bậc n, sau đó hiện kết quả trên màn hình.
8. Ma trận mxn (m hàng, n cột) có thể được mô tả bởi danh sách như sau:

   A=[[a11, a12, …a1n], [a21, a22,…,a2n], …,[am1, am2, …, amn]].

   Thực hiện các công việc sau:

   a. Viết chương trình nhập vào ma trận A với các phần tử a_{ij} là các số tự nhiên được nhập từ bàn phím.

   b. Tính tổng các phần tử của Ma trận A.

BÀI TẬP TUPLE

9. Cho trước một tuple chứa các số tự nhiên sau (1,2,3,4,5,6,7,8,9,10,11). Viết code lưu nửa đầu và nửa cuối của tuple trên vào các tuple tp1 và tp2. In kết quả ra màn hình.

Hướng dẫn:

```
tp = (1,2,3,4,5,6,7,8,9,10,11)
tp1 = (tp[:5])
tp2= tp[5:]
print('Tuple nửa đầu', tp1)
print('Tuple nửa cuối', tp2)
```

Kết quả chạy chương trình

```
Tuple nửa đầu (1, 2, 3, 4, 5)
Tuple nửa cuối (6, 7, 8, 9, 10, 11)
```

10. Viết chương trình tạo một tuple chứa toàn các số lẻ được lọc ra từ tuple cho trước:
tp =(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

```
tp=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
        ds=list()
        for i in tp:
            if tp[i-1]%2!=0:
                ds.append(tp[i-1])
        tp_le=tuple(ds)
print("Tuple số lẻ là : ",tp_le)
```

Kết quả chạy chương trình:

```
Tuple số lẻ là :  (1, 3, 5, 7, 9, 11, 13, 15)
```

11. Giả sử có hai danh sách, mỗi danh sách có nội dung như sau:

danh_sach_test = [(4,5), (7,6), (1,0), (3,4)]
cap_tim_kiem = [(3,4), (8,9), (7,6), (1,2)]

Viết chương trình tìm chỉ mục của các cặp tuple danh sách cap_tim_kiem xuất hiện trong danh_sach_test.

```
# initializing list
test_list = [(4, 5), (7, 6), (1, 0), (3, 4)]


# printing original list
```

```
print("The original list is : " + str(test_list))

# initializing search tuple
search_tup = [(3, 4), (8, 9), (7, 6), (1, 2)]

res=[]
for i in search_tup:
    if i in test_list:
        res.append(test_list.index(i))

# printing result
print("The match tuple indices : " + str(res))
```

Kết quả thực hiện chương trình:

```
The original list is : [(4, 5), (7, 6), (1, 0), (3, 4)]
The match tuple indices : [3, 1]
```

Cách 2. Sử dụng list comprehension + enumerate()

Cú pháp hàm enumurate()

enumerate(iterable, start=0)

Trong đó

- *iterable:* chuỗi, list, tuple, iterator hoặc bất cứ đối tượng hỗ trợ iteration nào.
- *start: enumerate()* **bắt đầu bộ đếm từ số này. Nếu tham số start bị bỏ qua thì 0 sẽ là giá trị mặc định được gán.**

Giá trị trả về từ enumerate()

Hàm enumerate() thêm vào một bộ đếm vào trước mỗi *iterable* và trả về kết quả dưới dạng đối tượng liệt kê. Các đối tượng *enumerate* này sau đó có thể được sử dụng trực tiếp trong các vòng lặp hoặc được chuyển đổi thành một danh sách, một tuple bằng phương thức list() và tuple().