

Hướng dẫn chi tiết

1. Cho chuỗi ký tự Str (có thể được nhập từ bàn phím), hãy đếm số các ký tự là số trong xâu trên và in kết quả ra màn hình.

Hướng dẫn:

```
1 Str=input('Nhập vào một chuỗi ký tự:')
2 print('Chuỗi ký tự vừa nhập: ',Str)
3 dem=0
4 for c in Str:
5     if "0" <= c <= "9":
6         dem+=1
7 print('Số các ký tự là số trong chuỗi đã nhập =',dem)
```

Kết quả thực hiện chương trình:

Nhập vào một chuỗi ký tự:abc0 e1

Chuỗi ký tự vừa nhập: abc0 e1

Số các ký tự là số trong chuỗi đã nhập = 2

2. Cho xâu S, hãy tạo xâu S' là nghịch đảo của S. ví dụ: S="TAN", S'="NAT"

Hướng dẫn

```
1 S=input("Nhập chuỗi ký tự S: ")
2 S1=""
3 n=len(S)
4 for i in range(1,n+1,1):
5     S1+=S[len(S)-i]
6 print("Chuỗi ký tự nghịch đảo của ",S," là: ",S1,"")
```

Kết quả chạy chương trình:

Nhập chuỗi ký tự S: ADEF

Chuỗi ký tự nghịch đảo của ' ADEF ' là: ' FEDA '

3. Cho xâu ký tự S1, S2, hãy in ra vị trí xuất hiện sau cùng của xâu S2 trong S1.

Hướng dẫn:

```
1 S1=input("Nhập chuỗi ký tự S1: ")
2 S2=input("Nhập chuỗi ký tự S2: ")
3 vtc=S1.rfind(S2)
4 print("Vị trí xuất hiện sau cùng của chuỗi/xâu ký tự ", S2, " trong chuỗi ",S1, "là: %d"%vtc)
```

Kết quả chạy chương trình:

Nhập chuỗi ký tự S1: Trường Đại học Công nghệ - Đại học Quốc Gia Hà Nội

Nhập chuỗi ký tự S2: Đại học

Vị trí xuất hiện sau cùng của chuỗi/xâu ký tự Đại học trong chuỗi Trường Đại học Công nghệ - Đại học Quốc Gia Hà Nội là: 27

4. Cho một xâu ký tự chỉ gồm các ký tự '(' và ')' hãy kiểm tra xem xâu ký tự đó có là một biểu thức đúng hay không. Biết rằng các biểu thức đúng có dạng sau: (), (biểu thức đúng)(biểu thức đúng), (biểu thức đúng).

Hướng dẫn:

1	Str = input('Nhập biểu thức ')
2	n=len(Str)
3	kt=int(0)
4	for i in range(0,n):
5	if Str[i]=='(': kt+=1
6	if Str[i]==')': kt-=1
7	if kt < 0 : break
8	if kt!=0 :
9	print('Biểu thức không đúng! ')
10	else:
11	print('Biểu thức đúng!')

Kết quả chạy chương trình:

Nhập biểu thức (a+b)*(c-d)/(2*a+c)

Biểu thức đúng!

Nhập biểu thức (a+b)-(c+d

Biểu thức không đúng!

5. Cho hai xâu ký tự S1, S2, hãy kiểm tra xem có thể xóa đi một số các ký tự của xâu S1 thì ta được xâu S2 hay không.

Hướng dẫn:

1	S1 = input('Nhập chuỗi S1: ')
2	S2 = input('Nhập chuỗi S2: ')
3	
4	i, j = 0, 0
5	while i < len(S1) and j < len(S2):
6	if S1[i] == S2[j]:
7	j += 1
8	i += 1
9	

```

10 if j == len(S2):
11     print("Yes, có thể nhận được S2 từ S1 bằng cách xóa đi một số ký tự.")
12 else:
13     print("No, không thể nhận được S2 từ S1 bằng cách xóa đi một số ký tự..")

```

Kết quả thực hiện chương trình:

Nhập chuỗi S1: abcdef

Nhập chuỗi S2: acf

Yes, có thể nhận được S2 từ S1 bằng cách xóa đi một số ký tự.

Nhập chuỗi S1: abcdef

Nhập chuỗi S2: kkk

No, không thể nhận được S2 từ S1 bằng cách xóa đi một số ký tự..

6. Cho họ và tên một sinh viên, hãy trích ra tên của sinh viên đó. Ví dụ: S="Nguyễn Van Minh", in ra màn hình "Ten sinh vien: Minh".

Hướng dẫn.

Cách 1: Code 7.6.a.py

```

1 str=input('Nhập họ tên sinh viên: ')
2 ten=""
3 pos = len(str)
4 while str[pos-1]!=" ": pos-=1
5 for i in range(pos,len(str)):
6     ten+=str[i]
7 print("Tên sinh viên là:",ten)

```

Kết quả chạy chương trình:

Nhập họ tên sinh viên: Nguyễn Văn Minh

Tên sinh viên là: Minh

Cách 2: code 7.6b.py, sử dụng hàm **rfind()** trả về vị trí của ký tự trắng đầu tiên trong chuỗi từ bên phải.

```

1 str=input('Nhập họ tên sinh viên: ')
2 pos = str.rfind(" ")
3 print("Tên sinh viên là:",str[pos+1:])

```

Cách 3: code 7.6c.py

```

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

```

4	<code>print("Tên sinh viên là:",ten)</code>
---	---

7. Một xâu ở dạng chuẩn nếu đầu và cuối xâu không có dấu cách, trong xâu không có hai dấu cách liền nhau. Viết chương trình chuẩn hóa một xâu ký tự.

Hướng dẫn:

1	<code>s=input('Nhập chuỗi ký tự: ')</code>
2	<code>s= " ".join(s.split())</code>
3	<code>print("Chuỗi sau khi chuẩn hóa là:",s,"")</code>

Kết quả chạy chương trình:

Nhập chuỗi ký tự: Hà Nội là Thủ Đô của Việt Nam !

Chuỗi sau khi chuẩn hóa là:'Hà Nội là Thủ Đô của Việt Nam!'

8. Nhập vào một chuỗi ký tự có độ dài 10 ký tự (lần ca chữ cái và chữ số)

Thực hiện các yêu cầu sau:

- Trích ra chuỗi ký tự con từ vị trí thứ 3 đến vị trí thứ 7 (Chú ý: Khái niệm “vị trí “ của người Việt khác với khái niệm index trong Python)
- Trích ra chuỗi ký tự con gồm 06 vị trí kể từ vị trí đầu tiên.
- Trích ra chuỗi ký tự con từ cuối chuỗi gồm 4 ký tự
- Đảo ngược chuỗi ký tự nhập vào.

9. Viết chương trình mã hóa một chuỗi/xâu ký tự `s_in` nhập vào từ bàn phím. Kết quả mã hóa được xuất ra vào chuỗi ký tự `s_out`.

Nguyên tắc mã hóa như sau: với mỗi ký tự của `s_in` sẽ được mã hóa bằng ký tự với số thứ tự trong bảng mã Unicode cộng thêm 2.

Hướng dẫn:

1	<code># Khai báo chuỗi s_in để lưu trữ chuỗi đầu vào</code>
2	<code>s_in = input("Nhập vào chuỗi cần mã hóa: ")</code>
3	
4	<code># Khai báo chuỗi s_out để lưu trữ chuỗi sau khi mã hóa</code>
5	<code>s_out = ""</code>
6	
7	<code># Duyệt từng ký tự trong chuỗi s_in</code>
8	<code>for char in s_in:</code>
9	<code> # Lấy ra mã Unicode của ký tự</code>
10	<code> char_code = ord(char)</code>
11	<code> # Mã hóa ký tự</code>
12	<code> char_code = char_code + 2</code>

```

13     # Chuyển mã Unicode về ký tự
14     char = chr(char_code)
15     # Thêm ký tự vào chuỗi s_out
16     s_out += char

```

Kết quả thực hiện chương trình:

Nhập vào chuỗi cần mã hóa: Ngân hàng đề thi học phần tin cơ sở

Chuỗi sau khi mã hóa: Piăp"jâpi"ēē"vjk"jôe"rjăp"vkp"eq"uở

10. Viết chương trình giải mã chuỗi ký tự xuất ra từ bài tập 9.

Hướng dẫn:

```

1     # Khai báo chuỗi s_in để lưu trữ chuỗi cần giải mã
2     s_in = input("Nhập vào chuỗi cần giải mã: ")
3
4     # Khai báo chuỗi s_out để lưu trữ chuỗi sau khi giải mã
5     s_out = ""
6
7     for char in s_in:
8         char_code = ord(char) # Chuyển ký tự sang mã Unicode
9         char_code -= 2 # Trừ 2 từ mã Unicode
10        s_out += chr(char_code) # Chuyển mã Unicode sang ký tự và thêm vào
11        s_in
12    print('Chuỗi sau khi giải mã:', s_out)

```

Kết quả thực hiện chương trình:

Nhập vào chuỗi cần giải mã: Piăp"jâpi"ēē"vjk"jôe"rjăp"vkp"eq"uở

Chuỗi sau khi giải mã: Ngân hàng đề thi học phần tin cơ sở

11. Cho trước một chuỗi ký tự (có thể nhập từ bàn phím) S. Viết chương trình đếm số chữ cái tiếng Anh (bao gồm cả chữ hoa, chữ thường) trong xâu S.

Hướng dẫn:

Cách 1. Sử dụng phương pháp so sánh từng ký tự trong xâu s với các ký tự chữ cái tiếng Anh, nếu ký tự đó thuộc về khoảng 'a' đến 'z' hoặc 'A' đến 'Z' thì tăng biến đếm count lên
Code 11a.

```

1     # Nhập chuỗi s từ bàn phím
2     s = input("Nhập chuỗi: ")
3
4     # Khởi tạo biến đếm số chữ cái

```

```

5 count = 0
6
7 # Duyệt từng ký tự trong chuỗi s
8 for char in s:
9     # Kiểm tra xem ký tự đó có phải là chữ cái không
10    if 'a' <= char <= 'z' or 'A' <= char <= 'Z':
11        count += 1
12
13 # Xuất ra kết quả
14 print("Số chữ cái trong chuỗi là:", count)

```

Cách 2: Sử dụng hàm `isalpha()` trả về True nếu ký tự đó là chữ cái tiếng Anh (hoặc chữ hoa hoặc chữ thường), và False nếu ký tự đó không phải là chữ cái tiếng Anh.

Code 11.b

```

1 # Nhập chuỗi
2 S = input("Nhập chuỗi: ")
3
4 # Khởi tạo biến đếm
5 count = 0
6 # Duyệt từng ký tự của chuỗi
7 for char in S:
8     # Nếu ký tự là chữ cái tiếng Anh (bao gồm cả hoa, thường)
9     if char.isalpha():
10        count += 1
11 # In kết quả
12 print("Số chữ cái tiếng Anh trong chuỗi:", count)

```

Kết quả thực hiện chương trình:

Nhập chuỗi: sndsansanflkksaeiwe798798asda~!@#%\$#%abc

Số chữ cái tiếng Anh trong chuỗi: 26

12. Cho trước 2 chuỗi ký tự S1, S2 (giá trị có thể được nhập vào từ bàn phím) Viết chương trình thực hiện các yêu cầu sau:

- Kiểm tra xem chuỗi ký tự S2 có nằm trong chuỗi ký tự S1 không?
- Nếu có thì chuỗi S2 được xuất hiện (có thể không chồng lên nhau) trong chuỗi S1 bao nhiêu lần?
- Tương tự câu b nhưng chỉ tính số lần xuất hiện chuỗi S2 trong chuỗi S1 không chồng lên nhau.

Hướng dẫn:

```
1  # Nhập chuỗi S1 và S2 từ người dùng
2  S1 = input("Nhập chuỗi S1: ")
3  S2 = input("Nhập chuỗi S2: ")
4
5  # Tính số lần xuất hiện của S2 trong S1 (có chồng lẫn)
6  count_overlap = 0      #biến đếm số lần xuất hiện S2 trong S1(có
7  chồng lẫn)
8  start = 0
9  while True:
10     #trả về vị trí xuất hiện của S2 trong S1 từ vị trí start
11     start = S1.find(S2, start)
12     if start == -1:
13         break
14     count_overlap += 1
15     start+=1
16
17 # Tính số lần xuất hiện của S2 trong S1 (không chồng lẫn)
18 count_non_overlap = 0  #biến đếm số lần xuất hiện S2 trong S1(có
19 chồng lẫn)
20 start = 0
21 while True:
22     start = S1.find(S2, start)
23     if start == -1:
24         break
25     count_non_overlap += 1
26     start = start + len(S2)
27
28 # In kết quả
29 if count_overlap == 0:
30     print("Chuỗi S2 không nằm trong chuỗi S1")
31 else:
32     print("Chuỗi S2 có nằm trong chuỗi S1")
33     print("S2 xuất hiện trong S1 (có chồng lẫn) ", count_overlap, "lần")
34     print("S2 xuất hiện trong S1 (không chồng lẫn) ", count_non_overlap, "lần")
```

Kết quả chạy chương trình:

Nhập chuỗi S1: abababababab

Nhập chuỗi S2: aba

Chuỗi S2 có nằm trong chuỗi S1

S2 xuất hiện trong S1 (có chồng lấn) 5 lần

S2 xuất hiện trong S1 (không chồng lấn) 3 lần

13. Nhập một chuỗi ký tự Str từ bàn phím. Kiểm tra Str có phải là chuỗi nhị phân (chỉ gồm các ký tự số 0 và 1) hay không? Nếu đúng hãy chuyển chuỗi ký tự số này sang số thập phân. In kết quả ra màn hình.

Hướng dẫn

```
1  # Nhập chuỗi ký tự str
2  str = input("Nhập một chuỗi ký tự nhị phân: ")
3
4  # Kiểm tra chuỗi str có phải là nhị phân hay không
5  binary = True
6  for char in str:
7      if char != '0' and char != '1':
8          binary = False
9          break
10
11 # Nếu chuỗi str là nhị phân, chuyển sang số thập phân và in kết quả ra
12 # màn hình
13 if binary:
14     decimal = 0
15     for i in range(len(str)):
16         decimal += int(str[i]) * (2**(len(str) - i - 1))
17     print("Số thập phân tương ứng là:", decimal)
18 else:
19     print("Chuỗi ký tự bạn nhập không phải là chuỗi nhị phân.")
```

Kết quả thực hiện chương trình:

Nhập một chuỗi ký tự nhị phân: 11111111

Số thập phân tương ứng là: 255

14.