Before modifying

prefixes = 'JKLMNOPQ'  
suffix = 'ack'  
  
for letter in prefixes:  
 print(letter + suffix)

>>> Jack

Kack

Lack

Mack

Nack

Oack

Pack

Qack

After modifying

prefixes = 'JKLMNOPQ'  
suffix = 'ack'  
  
for letter in prefixes:  
 if letter == 'O' or letter == 'Q':  
 print(letter+ 'u' + suffix)  
 else:  
 print(letter + suffix)

>>> Jack

Kack

Lack

Mack

Nack

Ouack

Pack

Quack

In this modifying, I put an if condition into for loop, once meet one of the condition, variable letter=’O’ or letter=’Q’, print statement will add ‘u’ between letter and suffix.

Part 2

Let’s try an example print single letter from string, now we give a string to variable str, and print the first and second letter.

str="today is a beautiful day"  
print(str[0])  
print(str[1])

>>>t

o

Then make another example.

str="today is a beautiful day"  
print(str[0:2])  
print(str[1:2])

>>>to

o

In this example, we can see str[a:b] is to slice the substring from string from character a until b (not include b).

str="today is a beautiful day"  
print(str[0:4:2])  
print(str[1:4:2])  
print(str[0:22:3])  
print(str[1:19:2])

>>> td

oa

taiaetud

oa sabatf

this example show step slice from a string, in str[a:b:c], the first letter we get is str[a] until str[b], the step-length is c.

str="today is a beautiful day"  
  
print(str[-1:-19:-2])  
print(str[1:19:2])

>>>ydlftabas

oa sabatf

if we make a negative number as flag, there no syntax, we can see the result is an inverted order.

Another interesting example for negative number:

str="today is a beautiful day"  
  
print(str[-1:-19:2])  
print(str[-1:-19:-2])  
print(str[1:-19:2])  
print(str[1:-19:-2])

>>>null

ydlftabas

oa

null

lastly, we try str[a:b:c:d]

str="today is a beautiful day"  
  
print(str[-1:-19:2:1])

>>> SyntaxError: invalid syntax

That is mean str[a:b:c:d] is invalid.

References  
Downey, A. (2015). Think Python: How to think like a computer scientist. Green Tree Press.