

```
#####
## EXAMPLE: for loops over strings
#####
#s = "demo
loops"
#for index in range(len(s)):
#    if s[index] == 'i' or s[index] == 'u':
#
print("There is an i or u")
#
#for char in s:
#    if char == 'i' or char == 'u':
#
    print("There is an i or u")

#####
## EXAMPLE: while loops and
strings
## CHALLENGE: rewrite while loop with a for loop
#####
#an_letters =
"aefhilmnorsxAEFHILMNORSX"
#word = input("I will cheer for you! Enter a word:
")
#times = int(input("Enthusiasm level (1-10): "))
#
#i = 0
#while i <
len(word):
#    char = word[i]
#    if char in an_letters:
#        print("Give me an
" + char + "! " + char)
#    else:
#        print("Give me a " + char
+ "! " + char)
#    i += 1
#print("What does that spell?")
#for i in
range(times):
#    print(word, "!!!")

#####
## EXAMPLE: perfect
cube
#####
#cube = 27
##cube = 8120601
#for guess in range(cube+1):
#    if
guess**3 == cube:
#        print("Cube root of", cube, "is", guess)
#
# loops keeps going even after found the cube root

#####
## EXAMPLE:
guess and check cube root
#####
#cube = 27
##cube = 8120601
#for guess in
range(abs(cube)+1):
#    # passed all potential cube roots
#    if guess**3 >= abs(cube):
#
#        # no need to keep searching
#        break
#if guess**3 != abs(cube):
#    print(cube,
```

```

'is not a perfect cube')
#else:
#     if cube < 0:
#         guess = -guess
#     print('Cube
root of ' + str(cube) + ' is ' + str(guess))

#####
## EXAMPLE: approximate
cube root
#####
#cube = 27
##cube = 8120601
##cube = 10000
#epsilon =
0.1
#guess = 0.0
#increment = 0.01
#num_guesses = 0
## look for close enough answer and make
sure
## didn't accidentally skip the close enough bound
#while abs(guess**3 - cube) >=
epsilon and guess <= cube:
#     guess += increment
#     num_guesses += 1
#print('num_guesses
=', num_guesses)
#if abs(guess**3 - cube) >= epsilon:
#     print('Failed on cube root of',
cube, "with these parameters.")
#else:
#     print(guess, 'is close to the cube root
of', cube)

#####
## EXAMPLE: bisection cube root (only positive
cubes!)
#####
#cube = 27
##cube = 8120601
## won't work with x < 1 because
initial upper bound is less than ans
##cube = 0.25
#epsilon = 0.01
#num_guesses = 0
#low =
0
#high = cube
#guess = (high + low)/2.0
#while abs(guess**3 - cube) >= epsilon:
#     if
guess**3 < cube:
#         # look only in upper half search space
#         low = guess
#     else:
#         # look only in lower half search space
#         high = guess
#     # next guess
is halfway in search space
#     guess = (high + low)/2.0
#     num_guesses +=
1
#print('num_guesses =', num_guesses)
#print(guess, 'is close to the cube root of', cube)

```