```
>>> # 13/02/23
>>> d = {'Jan':1, 'Feb': 2, 'Mar':3}
>>> d
{'Jan': 1, 'Feb': 2, 'Mar': 3}
# Iterating over the dictionary using just keys. ‘i’ assumes the key
>>> for i in d:
     print(i)


Jan
Feb
Mar

# Iterating over the dictionary using both keys and values.
>>> for i, v in d.items():
     print(i, v)


Jan 1
Feb 2
Mar 3


>>> # Gather input(int) from user and print the same in words
>>> 12
12
>>> "Twelve"
'Twelve'
>>> # Constraint the value 0 to 9
>>> num_words = {1 : 'one', 2: 'two', 3:'three', 4:'four', 5:'five',
6 :"six", 7:"seven", 8:"eight" ,9 : "nine"}

>>> num_words
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 6: 'six', 7:
'seven', 8: 'eight', 9: 'nine'}
>>> n = input("Enter a digit")
Enter a digit4
>>> n
'4'
>>> "four"
'four'

>>> num_words.keys()
dict_keys([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> n ==num_words.keys()
```

```
False

# Keys are not subscriptable
>>> num_words.keys()[1]
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    num_words.keys()[1]
TypeError: 'dict_keys' object is not subscriptable

>>> for key in num_words:
     if int(n) == key:
          print(num_words[int(n)])



four

# Convert previous code to a function, ensure to supply an integer

>>> def num_to_words(num):
     """Returns word equivalent for a given digit"""
     for key in num_words:
          if num == key:
               print(num_words[num])



>>> num_to_words(int(n))
four
>>> num_to_words(3)
'three'
>>> num_to_words(9)
'nine'
>>> num_to_words(10)  # returns None as 10 is not a key
>>> x = num_to_words(10)
>>> x
>>> num_words[10]
Traceback (most recent call last):
  File "<pyshell#43>", line 1, in <module>
    num_words[10]
KeyError: 10
>>> num_words.get(9)
'nine'
>>> num_words.get(10)
>>> x = num_words.get(10)
>>> x
```

```
# The same function can be reduced to a single line using the .get
method

>>> def num_to_words_2(num):
     """Returns word equivalent for a given digit"""
     return num_words.get(num)

>>> num_to_words_2(int(input("Enter a digit")))
Enter a digit3
'three'

>>> num_to_words_2(int(input("Enter a digit\n ")))
Enter a digit
 7
'seven'

>>> num_to_words_2(int(input("Enter a digit\n ")))
Enter a digit
 12

# Update the same function by printing appropriate error message
>>> def num_to_words_2(num):
     """Returns word equivalent for a given digit"""
     return num_words.get(num, "Enter a valid digit from 0 to 9")

>>> num_to_words_2(int(input("Enter a digit\n ")))
Enter a digit
 12
'Enter a valid digit from 0 to 9'
>>> # break
>>> # if a number is div by 3 then stop the loop
>>> for i in range(1, 10):
     if (i % 3 == 0):
          break
     print(i)



1
2
# continue
# if a condition is true skip the current iteration and move to next
# iteration
>>> for i in range(1, 10):
     if (i % 3 == 0):
          continue
```

```
    print(i)


1
2
4
5
7
8


----------------------------------------------------------------------
```