upGrad

PLACEMENT REFRESHER PROGRAM

Session 1: Python I

Basics, Data Types, Control Structures

By
Ritesh Kumar Pandey

Agenda

- Basics
- Data Types
- Control Structures

Introduction

upGrad

Python

- High Level
- General Purpose
- Interpreted
- Dynamically Typed

Introduction

upGrad

Python

- High Level
- General Purpose
- Interpreted
- Dynamically Typed

Provides abstractions and simplifications making it Human Understandable language

Introduction

upGrad

Python

- High Level
- General Purpose
- Interpreted
- Dynamically Typed

Provides abstractions and simplifications making it Human Understandable language

Web Development, Data Science, Artificial Intelligence, Machine Learning, Application Development, Data Analytics

Introduction

upGrad

Python

- High Level
- General Purpose
- Interpreted
- Dynamically Typed

Provides abstractions and simplifications making it Human Understandable language

Web Development, Data Science, Artificial Intelligence, Machine Learning, Application Development, Data Analytics

Executed Line by Line

Introduction

upGrad

Python

- High Level
- General Purpose
- Interpreted
- Dynamically Typed

Provides abstractions and simplifications making it Human Understandable language

Web Development, Data Science, Artificial Intelligence, Machine Learning, Application Development, Data Analytics

Executed Line by Line

Variable types are determined at runtime
int x = 10 # not required
x = 10

Variable Assignment

upGrad

Question 1

Which of the following is invalid ?

a) it = 100
b) on = 200
c) in = ‘Hi’
d) apple = mango

upGrad

Answer

Which of the following is invalid ?

a) it = 100
b) on = 200
c) in = ‘Hi’
d) apple = mango

Variable Assignment

upGrad

Answer

Which of the following is invalid ?

a) it = 100
b) on = 200
c) in = ‘Hi’
d) apple = mango

in is a Keyword in Python, hence cannot be used a variable name.

There are a total of **35** keywords in Python.

True, False, None are the only capitalized keywords, rest all are small case

Variable Assignment

upGrad

Answer

Which of the following is invalid ?

a) it = 100
b) on = 200
c) in = ‘Hi’
d) apple = mango

in is a Keyword in Python, hence cannot be used a variable name.

There are a total of **35** keywords in Python.

True, False, None are the only capitalized keywords, rest all are small case

```
import keyword

keywords = keyword.kwlist
print(" The total number of keywords in Python is : ", len(keywords))
print("\n The list of keywords in Python is : ", keywords)
```

Variable Assignment

upGrad

Answer

Which of the following is invalid ?

a) it = 100
b) on = 200
c) in = ‘Hi’
d) apple = mango

in is a Keyword in Python, hence cannot be used a variable name.

There are a total of **35** keywords in Python.

True, False, None are the only capitalized keywords, rest all are small case

```
import keyword

keywords = keyword.kwlist
print(" The total number of keywords in Python is : ", len(keywords))
print("\n The list of keywords in Python is : ", keywords)
```

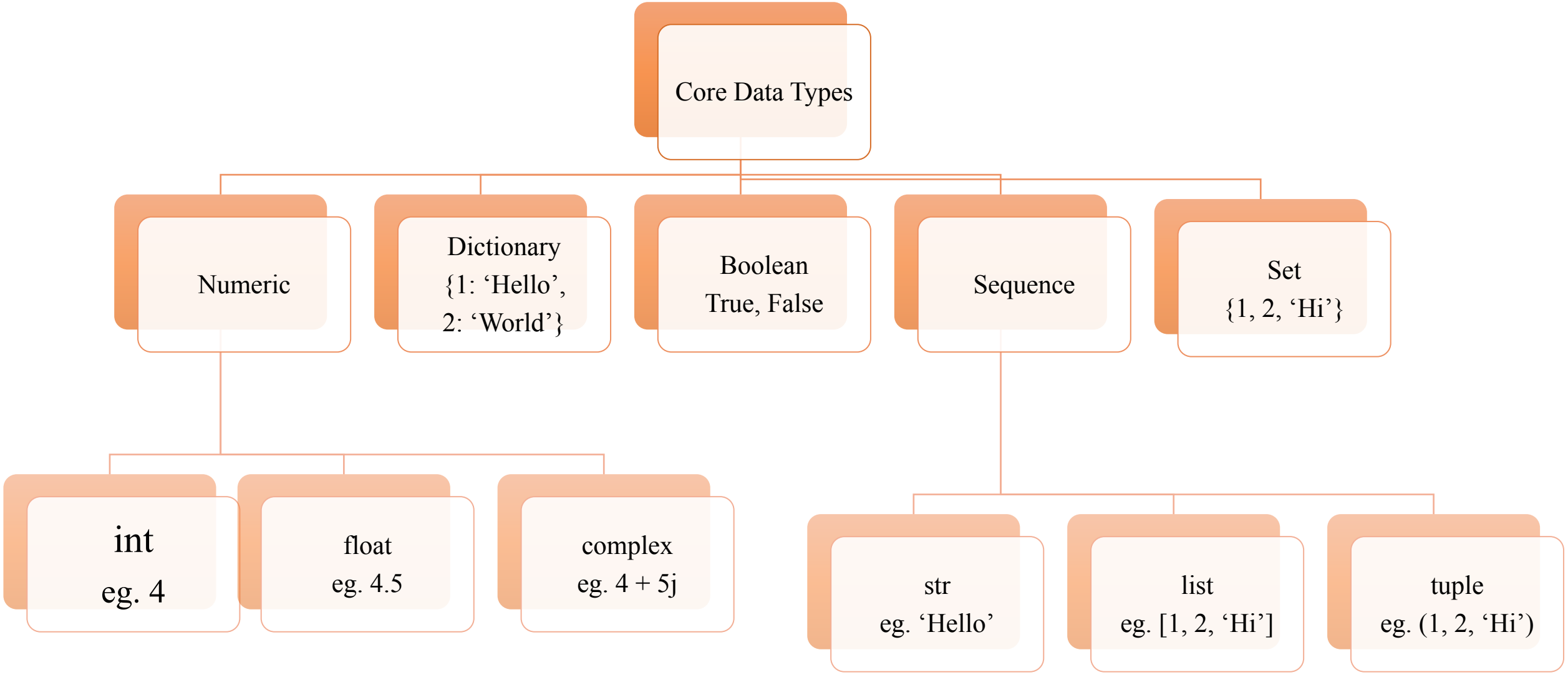
mango is not defined

```
mango = 5
apple = mango
```

```
apple = ‘mango’
```

Data Types

upGrad



Question 2

```
x = input('Enter Something : ')
print(type(x))
```

upGrad
32
32.5
True

Answer

```
x = input('Enter Something : ')
print(type(x))
```

Class <'str'>

upGrad
32
32.5
True

upGrad

Question 3

```
var1 = ['Hi', 'Hello', 23]
print(type(var1))
var1[1] = 'Hey'
print(var1)
```

```
var2 = ('Hi', 'Hello', 23)
print(type(var2))
var2[1] = 'Hey'
print(var2)
```

```
var3 = {'Hi', 'Hello', 23}
print(type(var3))
var3[1] = 'Hey'
print(var3)
```

Data Types - List, Tuple, Set

upGrad

Answer

```
var1 = ['Hi', 'Hello', 23]
print(type(var1))
var1[1] = 'Hey'
print(var1)
```

```
var2 = ('Hi', 'Hello', 23)
print(type(var2))
var2[1] = 'Hey'
print(var2)
```

```
var3 = {'Hi', 'Hello', 23}
print(type(var3))
var3[1] = 'Hey'
print(var3)
```

class <'list'>
['Hi', 'Hey', 23]

Data Types - List, Tuple, Set

upGrad

Answer

```
var1 = ['Hi', 'Hello', 23]
print(type(var1))
var1[1] = 'Hey'
print(var1)
```

class <'list'>
['Hi', 'Hey', 23]

```
var2 = ('Hi', 'Hello', 23)
print(type(var2))
var2[1] = 'Hey'
print(var2)
```

class <'tuple'>
ERROR !

```
var3 = {'Hi', 'Hello', 23}
print(type(var3))
var3[1] = 'Hey'
print(var3)
```

Data Types - List, Tuple, Set

upGrad

Answer

```
var1 = ['Hi', 'Hello', 23]
print(type(var1))
var1[1] = 'Hey'
print(var1)
```

class <'list'>
['Hi', 'Hey', 23]

```
var2 = ('Hi', 'Hello', 23)
print(type(var2))
var2[1] = 'Hey'
print(var2)
```

class <'tuple'>
ERROR !

```
var3 = {'Hi', 'Hello', 23}
print(type(var3))
var3[1] = 'Hey'
print(var3)
```

class <'set'>
ERROR !

upGrad

Question 4

```
var1 = [‘Hi’, ‘Hello’, 23]
var1.append(‘Hey’)
var1.append(‘Hi’)
print(var1)
```

```
var2 = {‘Hi’, ‘Hello’, 23}
var2.add(‘Hey’)
var2.add(‘Hi’)
print(var2)
```

upGrad

Answer

```
var1 = [‘Hi’, ‘Hello’, 23]
var1.append(‘Hey’)
var1.append(‘Hi’)
print(var1)
```

```
var2 = {‘Hi’, ‘Hello’, 23}
var2.add(‘Hey’)
var2.add(‘Hi’)
print(var2)
```

[‘Hi’, ‘Hello’, 23, ‘Hey’, ‘Hi’]

upGrad

Answer

```
var1 = [‘Hi’, ‘Hello’, 23]
var1.append(‘Hey’)
var1.append(‘Hi’)
print(var1)
```

[‘Hi’, ‘Hello’, 23, ‘Hey’, ‘Hi’]

```
var2 = {‘Hi’, ‘Hello’, 23}
var2.add(‘Hey’)
var2.add(‘Hi’)
print(var2)
```

[‘Hi’, 23, ‘Hey’, ‘Hello’]

Data Types - List, Tuple, Set

upGrad

Aspect	List	Set	Tuple
Mutability	Mutable	Mutable	Immutable
Ordered	Ordered	Unordered	Ordered
Duplicates	Allowed	Not allowed	Allowed
Syntax	[ ]	{ }	( )

upGrad

Question 5

```
d = dict()
k = (1, 2, 3, 4)
d[k] = ‘Hey’
print(d)
```

```
d = dict()
k = [1, 2, 3, 4]
d[k] = ‘Hey’
print(d)
```

Data Types - Dictionary

upGrad

Answer

```
d = dict()
k = (1, 2, 3, 4)
d[k] = ‘Hey’
print(d)
```

{ (1, 2, 3, 4): ‘Hey’ }

```
d = dict()
k = [1, 2, 3, 4]
d[k] = ‘Hey’
print(d)
```

Data Types - Dictionary

upGrad

Answer

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
print(d)
```

{ (1, 2, 3, 4): 'Hey' }

```
d = dict()
k = [1, 2, 3, 4]
d[k] = 'Hey'
print(d)
```

ERROR !

upGrad

Question 6

```
d = dict()
k = (1, 2, 3, 4)
d[k] = ‘Hey’
m = (5, 6)
d[m] = ‘Hello’
print(d)
```

```
d = dict()
k = (1, 2, 3, 4)
d[k] = ‘Hey’
m = (1, 2, 3, 4)
d[m] = ‘Hello’
print(d)
```

```
d = dict()
k = (1, 2, 3, 4)
d[k] = ‘Hey’
m = ‘1, 2, 3, 4’
d[m] = ‘Hello’
print(d)
```

upGrad

Answer

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = (5, 6)
d[m] = 'Hello'
print(d)
```

{ (1, 2, 3, 4): 'Hey', (5, 6): 'Hello' }

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = (1, 2, 3, 4)
d[m] = 'Hello'
print(d)
```

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = '1, 2, 3, 4'
d[m] = 'Hello'
print(d)
```

upGrad

Answer

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = (5, 6)
d[m] = 'Hello'
print(d)
```

{ (1, 2, 3, 4): 'Hey', (5, 6): 'Hello' }

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = (1, 2, 3, 4)
d[m] = 'Hello'
print(d)
```

{ (1, 2, 3, 4): 'Hello' }

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = '1, 2, 3, 4'
d[m] = 'Hello'
print(d)
```

Data Types - Dictionary

upGrad

Answer

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = (5, 6)
d[m] = 'Hello'
print(d)
```

{ (1, 2, 3, 4): 'Hey', (5, 6): 'Hello' }

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = (1, 2, 3, 4)
d[m] = 'Hello'
print(d)
```

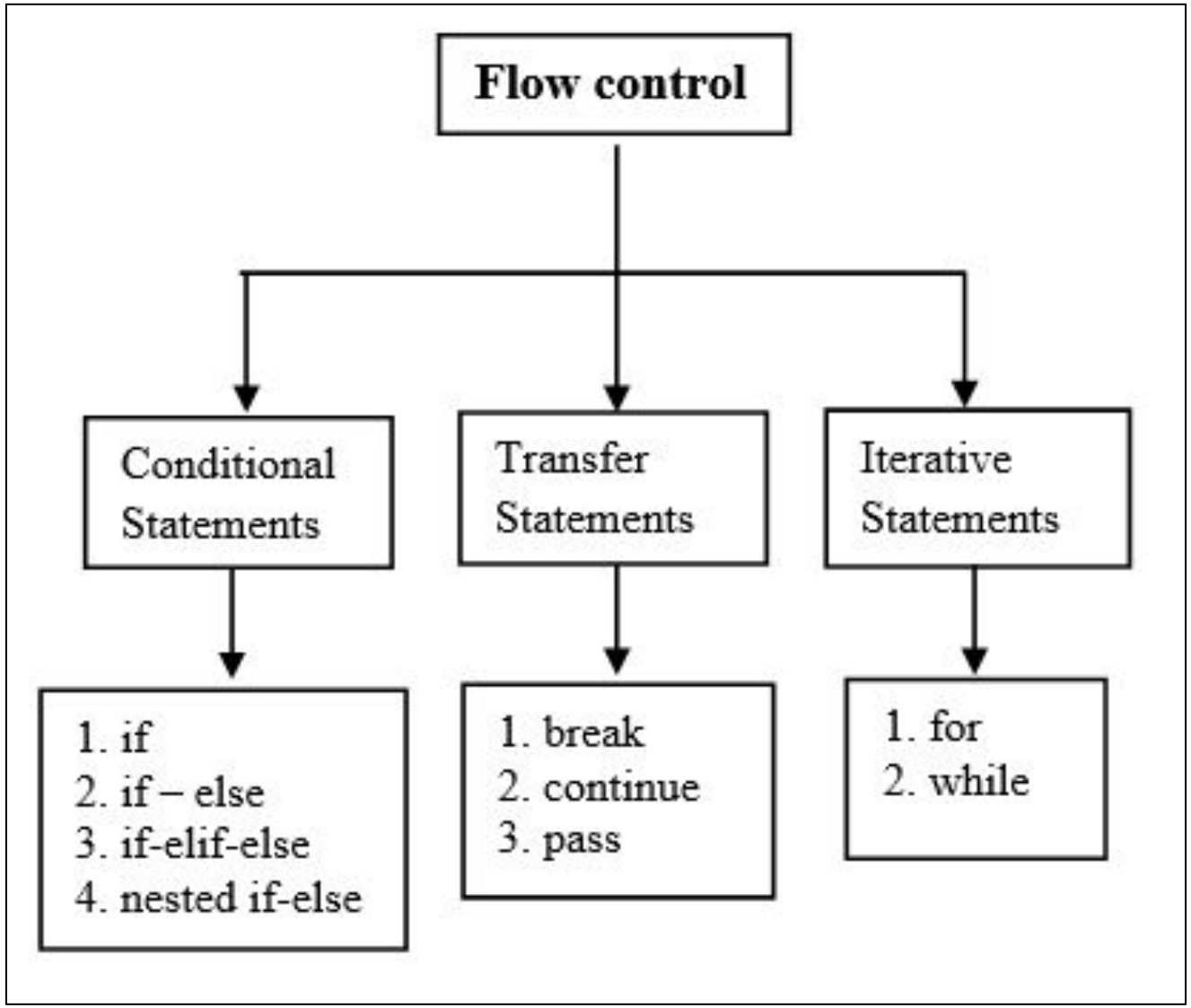
{ (1, 2, 3, 4): 'Hello' }

```
d = dict()
k = (1, 2, 3, 4)
d[k] = 'Hey'
m = '1, 2, 3, 4'
d[m] = 'Hello'
print(d)
```

{ (1, 2, 3, 4): 'Hey', '1, 2, 3, 4': 'Hello' }

Control Statements

upGrad



upGrad

Question 7

```
for i in range(0, 10):
    if i % 2 == 0:
        print(i)
    elif i % 5 == 0:
        continue
        print('Divisible by 5')
    elif i == 7:
        break
    else:
        print('No')

    print('something \n')
```

upGrad

Answer

```
for i in range(0, 10):
    if i % 2 == 0:
        print(i)
    elif i % 5 == 0:
        continue
        print('Divisible by 5')
    elif i == 7:
        break
    else:
        print('No')

    print('something \n')
```

```
0
something

No
something

2
something

No
something

4
something
6
something
```

Question 8

Given the following code, please rewrite it in Python after fixing all syntax errors. Please underline each correction you made.

```
30 = To
for k in (0, To):
    if k / 4 == 0:
        print(k is divisible by 4)
    else:
        print(k is not divisible by 4)
```

upGrad

Answer

Given the following code, please rewrite it in Python after fixing all syntax errors. Please underline each correction you made.

```
30 = To
for k in (0, To):
    if k / 4 == 0:
        print(k is divisible by 4)
    else:
        print(k is not divisible by 4)
```

```
To = 30
for k in range(0, To):
    if k % 4 == 0:
        print(k + 'is divisible by 4')
    else:
        print(k + 'is not divisible by 4')
```

Question 9

Input: [1, 2, 3, 6, 4, 8]
Output: [8, 4, 6, 3, 2, 1]

upGrad

Answer

Input: [1, 2, 3, 6, 4, 8]
Output: [8, 4, 6, 3, 2, 1]

```
list_name = list(reversed(list_name))
```

upGrad

Answer

Input: [1, 2, 3, 6, 4, 8]
Output: [8, 4, 6, 3, 2, 1]

```
list_name = list(reversed(list_name))
```

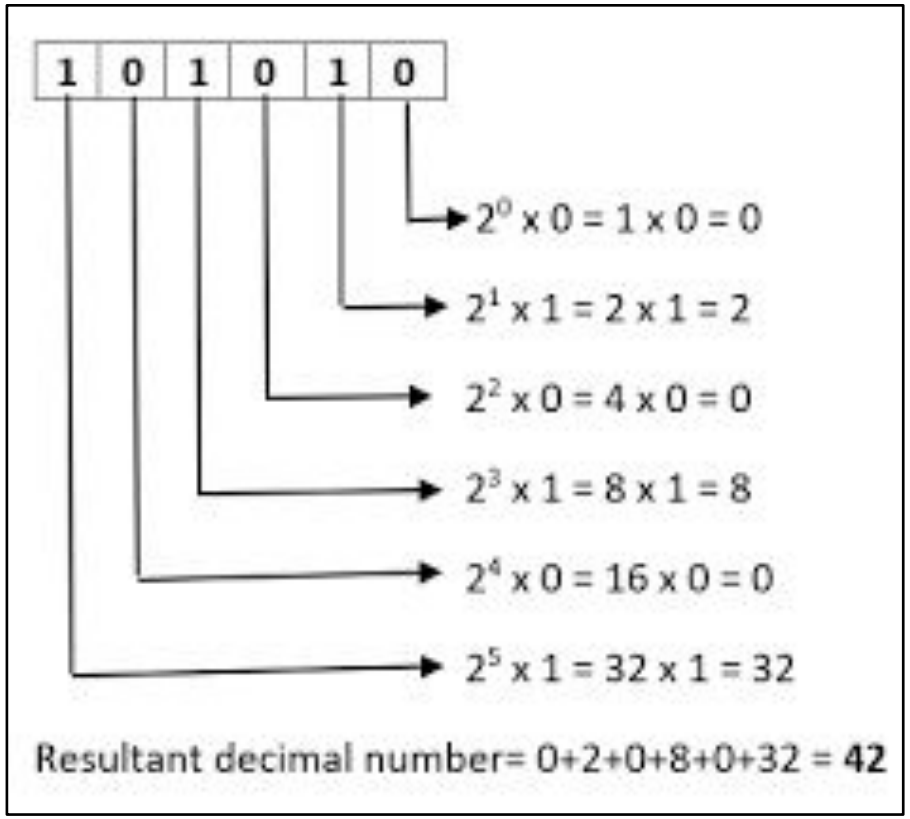
```
left = 0
right = len(list_name) – 1
while left < right:
    ll[left], ll[right] = ll[right], ll[left]
    left += 1
    right -= 1
print(original_list)
```

Question 10

Input: 0100, 0011, 1010, 1001, 1100, 1011, 0101
Output : 1010, 0101

Answer

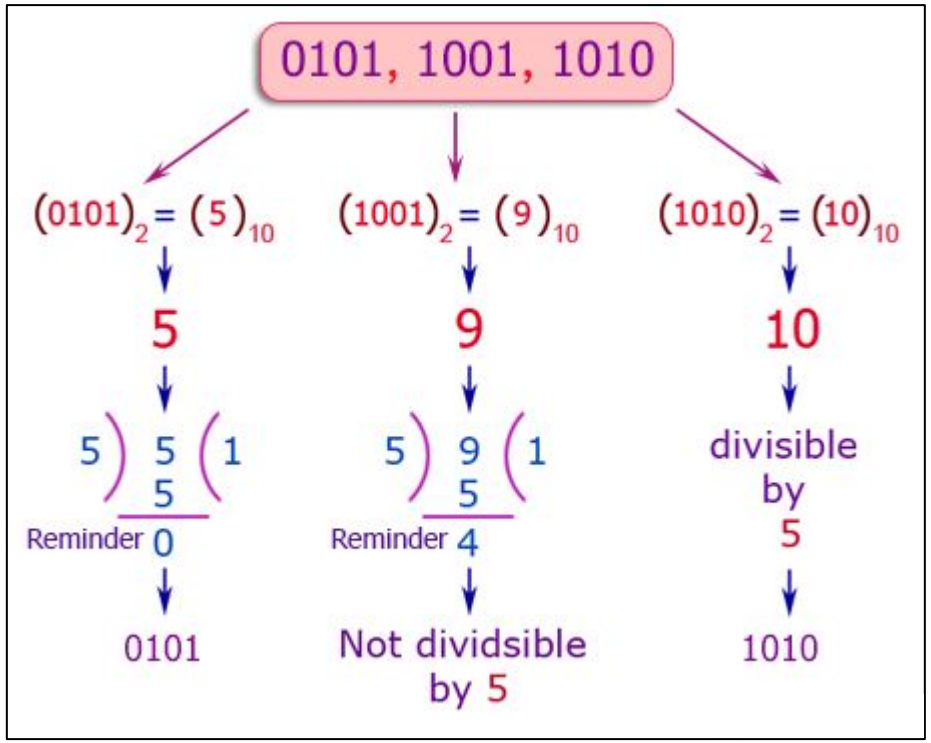
Input: 0100, 0011, 1010, 1001, 1100, 1011, 0101
Output : 1010, 0101



Answer

Binary Number divisible by 5

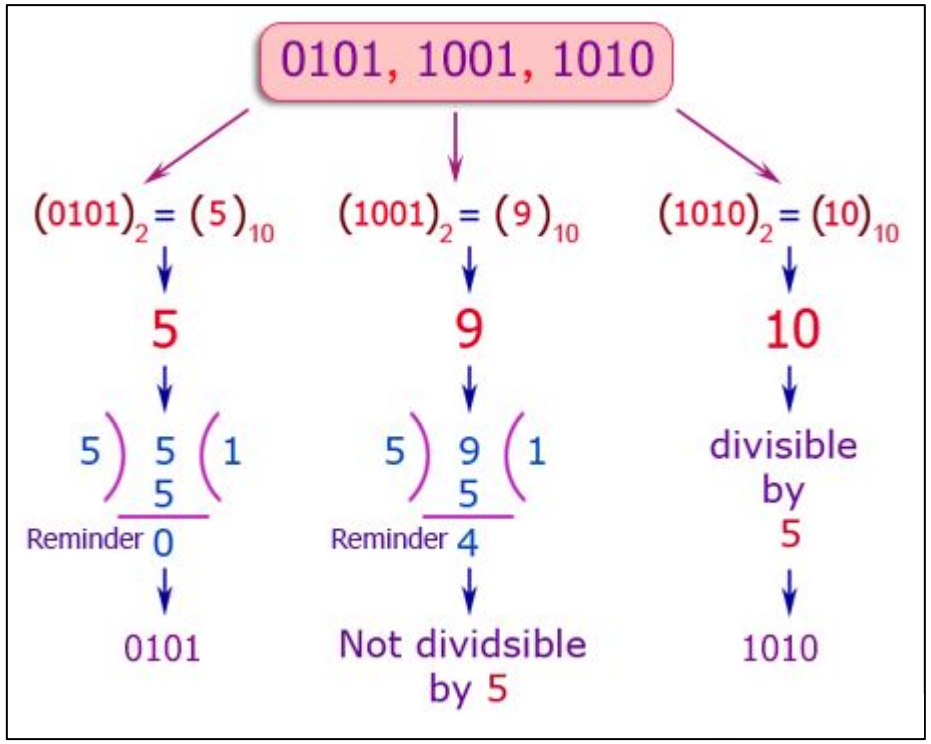
Input: 0100, 0011, 1010, 1001, 1100, 1011, 0101
Output : 1010, 0101



Answer

Binary Number divisible by 5

Input: 0100, 0011, 1010, 1001, 1100, 1011, 0101
Output : 1010, 0101




```
items = []

for p in input_list:
    x = int(p, 2)
    if not x%5:
        items.append(p)

print(','.join(items))
```

THANK YOU