# ox01. Caching

**ck-end**My Planning(/planning/me)

Weight: 1

盲

Projects(/projects/current)

₱ Project over - took place from Jul 23, 2024 6:00 AM to Jul 25, 2024 6:00 AM

An auto **CPA** i Previebres alvo a me that the flow or lieue tions / to review)

? Evaluation quizzes(/dashboards/my current evaluation quizzes) In a nutshell...

Auto QA review: 89.0/89 mandatory & 15.0/18 optional

Altogether: 183.33%

Curriculums)

Optional: 83.33%

Calculation: 100.0% + (100.0% \* 83.33%) == 183.33% Concepts(/concepts)

Conference rooms(/dashboards/video\_rooms)

### Background Context

In this project, you learn different caching algorithms.

Sandboxes(/user\_containers/current) Kesources

**Bead or watch**: 1001s(/dashboards/my\_tools)

- Cache replacement policies FIFO (/rltoken/fjhr6EvFeF3mWwsPQXUKdQ)
- Cache replacement policies LIFO (/rltoken/U44RQjXp8xBtsbNlyhHlyw)
- Calciner on demand (dashboards/vinters) (dashboards/vinters)
  - Cache replacement policies MRU (/rltoken/Tmk4qEBZ7QTknvbpKabWfQ)
  - Cache replacement policies LFU (/ritoken/8PEJ8L34bxhL2y--BW5zGQ)

# **Learning Objectives**

At the end of this project, you are expected to be able to explain to anyone (/rltoken/-gpAdRQTx1Rbmaz9Jzhoj, without the help of Google:

#### General

- What a caching system is
- What FIFO means My Profile(/users/my\_profile)
- What LIFO means

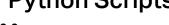




- What LRU means What MRU means () What LFU means What the purpose of a caching system What limits a caching system have Home(/) Requirements

## My Planning(/planning/me)

**Python Scripts** 



- 🗣 Alþrourdikawildæinterpreted/compiled on Ubuntu 18.04 LTS using python3 (version 3.7)
  - · All your files should end with a new line
  - The first line of all your files should be exactly #!/usr/bin/env python3
- A REARRY PAR THE THE TOUTH OF THE POLICE POPULATION OF THE PROPERTY OF THE P • Your code should use the pycodestyle style (version 2.5)
- All your files must be executable Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)
  The length of your files will be tested using wc

  - All your modules should have a documentation (python3 -c

```
'print(__import__("my_module").__doc__)')
```

- All your classes should have a documentation (python3 -c 'print(\_\_import\_\_("my\_module").MyClass.\_\_doc\_\_)')
  - All your functions (inside and outside a class) should have a documentation (python3 -c
- 首 'pြောက<u>eepts</u>Mpone<u>epts</u>my\_module").my\_function.\_\_doc\_\_)' and python3 -c 'print(\_\_import\_\_("my\_module").MyClass.my\_function.\_\_doc\_\_)')
- A documentation is not a simple word, it's a real sentence explaining what's the purpose of the Conference rooms(/dashboards/video\_rooms) module, class or method (the length of it will be verified)

#### TO reversife vers)

### >Parent class Base Caching

All your classes must inherit from BaseCaching defined below:

ע Tools(/dashboards/my tools)

冊 Video on demand(/dashboards/videos)

Peers(/users/peers)

0

Discord(https://discord.com/app)

```
$ cat base_caching.py
  #!/usr/bin/python3
  ""("/) BaseCaching module
Tclass Basser aching():
       """ BaseCaching defines:
         - constants of your caching system
        My Manaing Manaing / and stored (in a dictionary)
       \Pi \Pi \Pi
       MAX_ITEMS = 4
        Projects(/projects/current)
       def __init__(self):
           """ Initiliaze
        QA 'Reviews I can make(/corrections/to review)
           self.cache_data = {}
       detvantation quizzes (Alas) hooards/my current evaluation quizzes)
           """ Print the cache
           print("Current cache:")
           for key in sorted(self.cache_data.keys()):
Curriculums(/dashboards/my_curriculums)
       def put(self, key, item):
        Concepts (concepts) in the cache
           raise NotImplementedError("put must be implemented in your cache class")
        Conference rooms(/dashboards/video_rooms)
       def get(self, key):
           """ Get an item by key
        Servers(/servers)
           raise NotImplementedError("get must be implemented in your cache class")
        Sandboxes(/user_containers/current)
```

# Tools(/dashboards/my\_tools)

# Video on demand(/dashboards/videos) Basic dictionary

mandatory

Score: 100.0% (Checks completed: 100.0%)

Peers(/users/peers)

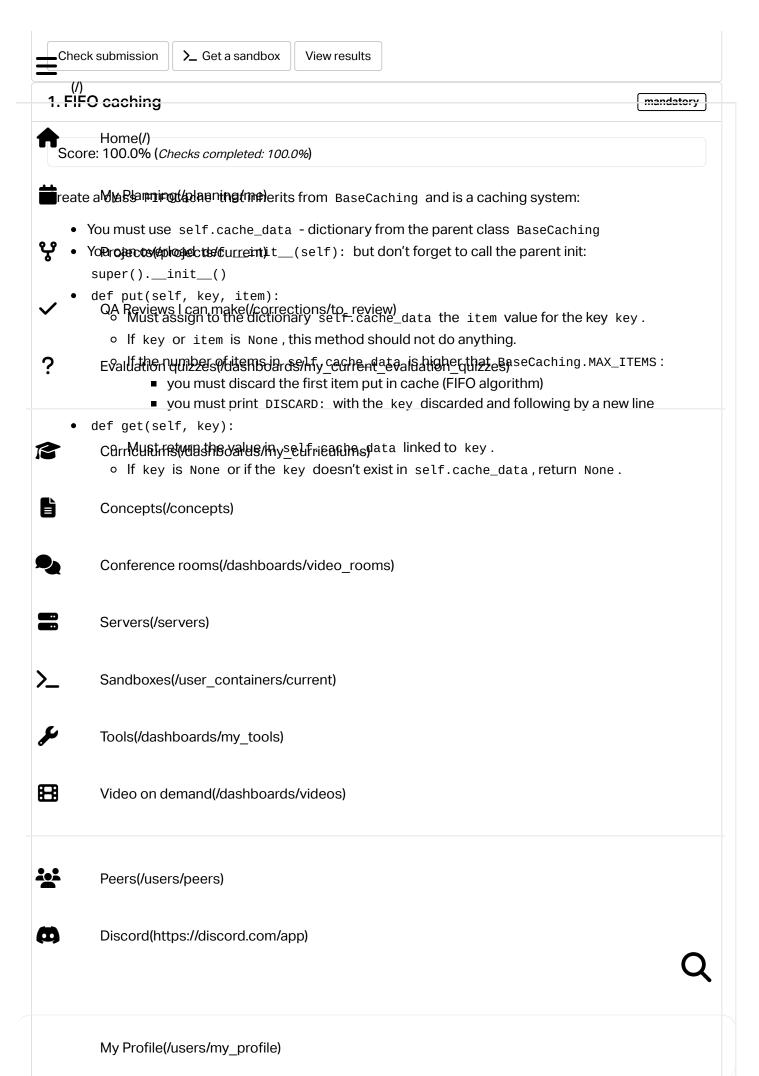
Create a class BasicCache that inherits from BaseCaching and is a caching system:

- You igust ប្រទេស វិបាន ខេត្ត ប្រាស់ You igust ប្រទេស Base Caching
  - This caching system doesn't have limit
  - def put(self, key, item):
    - Must assign to the dictionary self.cache\_data the item value for the key key.
    - If key or item is None, this method should not do anything.
  - def get(self, key): My Profile(/users/my\_profile)
    - Must return the value in self.cache\_data linked to key.

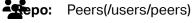
```
If key is None or if the key doesn't exist in self.cache_data, return None.
   gujllaume@ubuntu:~/0x01$ cat 0-main.py
   #l/usr/bin/python3
   """ 0-main """
___Basic������(片 __import__('0-basic_cache').BasicCache
   my_cache = BasicCache()
  my_cachie marining(Astalening/me)
   my_cache.put("A", "Hello")
   my_cache.put("B", "World")
Ymy_caphojects(/pGojects/clarection")
   my_cache.print_cache()
   print(my_cache.get("A"))
 print @x Reviews 9 can make (/corrections/to_review)
   print(my_cache.get("C"))
   print(my_cache.get("D"))
? my_cache print cache (/dashboards/my_current_evaluation_quizzes) my_cache.put("D", "School")
   my_cache.put("E", "Battery")
   my_cache.put("A", "Street")
   my_cache.print_cache()
Frint Guyrricad wens yetas haroayds/my_curriculums)
   guillaume@ubuntu:~/0x01$ ./0-main.py
CurrerCtoraepts(/concepts)
   Current cache:
   A: Hello
 B: Wordonference rooms(/dashboards/video_rooms)
  C: Holberton
   Hello
world Servers(/servers)
  Holberton
   None
\__Currentand66xes(/user_containers/current)
   A: Hello
   B: World
  C: Holbert (Mashboards/my_tools)
   Current cache:
   A: Street
B: Worldeo on demand(/dashboards/videos)
   C: Holberton
   D: School
   E: Battery
  Street
quillares (/bsets/peess)01$
         Discord(https://discord.com/app)
```

- GitHub repository: alx-backend
- Directory: 0x01-caching
- File: 0-basic\_cache.py

My Profile(/users/my\_profile)



```
guillaume@ubuntu:~/0x01$ cat 1-main.py
#!/usr/bin/python3
  ""(//) 1-main """
   FIF0Cache = __import__('1-fifo_cache').FIF0Cache
my_cacheme() IFOcache()
   my_cache.put("A", "Hello")
  my_cache.put("B", "World")
 my_cadN&Polanning(/plannoing/medn")
   my_cache.put("D", "School")
  my_cache.print_cache()
my_cadrojects(/projectscurenyt))
   my_cache.print_cache()
   my_cache.put("C", "Street")
wy_cacone Reviews declare hake (/corrections/to review)
   my_cache.put("F", "Mission")
  my_cache.print_cache()
        Evaluation quizzes(/dashboards/my current evaluation quizzes)
   guillaume@ubuntu:~/0x01$ ./1-main.py
   Current cache:
   A: Hello
  B: World
C: Ho Curriculums(/dashboards/my_curriculums)
   D: School
   DISCARD: A
current on gents (/concepts)
   B: World
   C: Holberton
🗫 : scif@nference rooms(/dashboards/video_rooms)
  E: Battery
  Current cache:
 සි: Worපිලrvers(/servers)
   C: Street
  D: School
__E: Bat<del>S</del>andboxes(/user_containers/current)
  DISCARD: B
   Current cache:
C: Strients(/dashboards/my tools)
  D: School
  E: Battery
F: MisAadeo on demand(/dashboards/videos)
   guillaume@ubuntu:~/0x01$
```



Check submission

• GitHub repository: alx-backend

• Diffsforki(https://disbord.com/app)

>\_ Get a sandbox

• File: 1-fifo\_cache.py

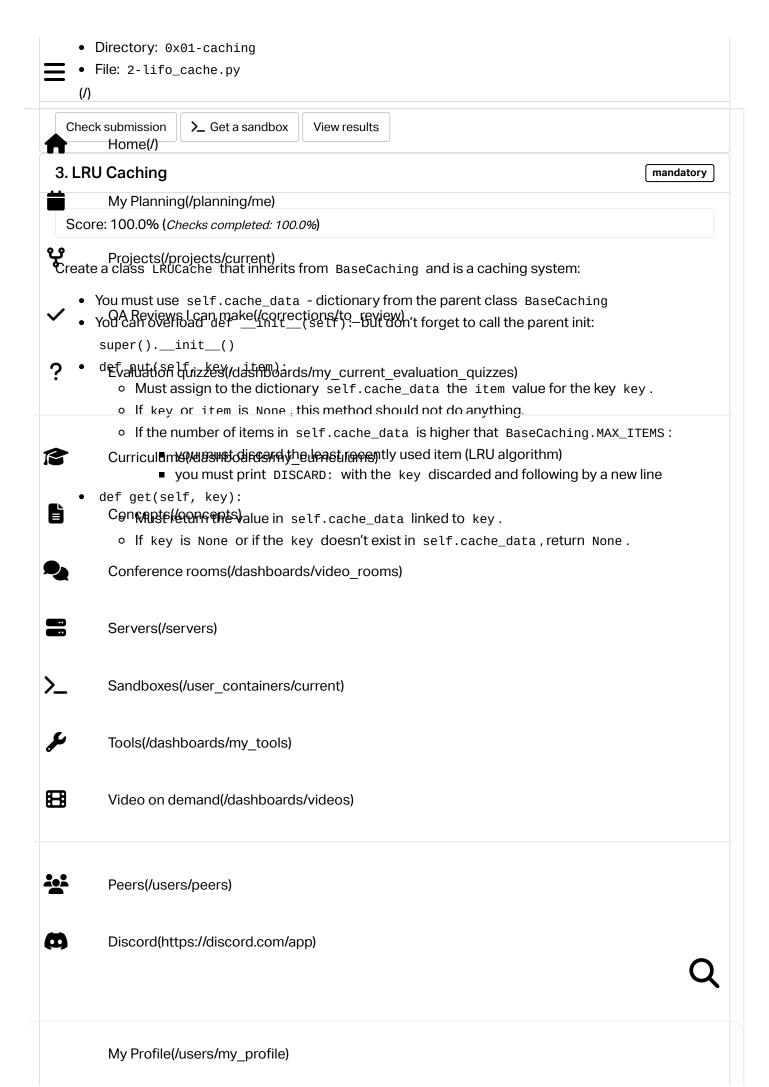
View results

My Profile(/users/my\_profile)

```
guillaume@ubuntu:~/0x01$ cat 2-main.py
#!/usr/bin/python3
  ""(//) 2-main """
   LIFOCache = __import__('2-lifo_cache').LIFOCache
my_cacheme(1) IFO cache()
   my_cache.put("A", "Hello")
  my_cache.put("B", "World")
 my_cadMy Phantoing(/plannoing/medn")
   my_cache.put("D", "School")
  my_cache.print_cache()
my_cadrojects(/projects/curenyt))
   my_cache.print_cache()
   my_cache.put("C", "Street")
wy_cacone Reviews declare hake (/corrections/to review)
   my_cache.put("F", "Mission")
  my_cache.print_cache()
🤁 my_caœាvalinatto/h ឲ្យแរzz២៩/៧៨ទាវាលាខា៎៧៩៧កាំ) current evaluation quizzes)
  my_cache.print_cache()
   guillaume@ubuntu:~/0x01$ ./2-main.py
  Current cache:
A: He Gurriculums(/dashboards/my_curriculums)
   B: World
   C: Holberton
D: Schooncepts(/concepts)
   DISCARD: D
   Current cache:
Pa: He Qonference rooms(/dashboards/video_rooms)
   B: World
  C: Holberton
 E: Bat Serwers (/servers)
   Current cache:
  A: Hello
__B: worSandboxes(/user_containers/current)
  C: Street
   E: Battery
FDISCAR®pls(/dashboards/my tools)
  Current cache:
  A: Hello
B: WorWideo on demand(/dashboards/videos)
  E: Battery
   F: Mission
   DISCARD: F
Current cache:
  A: He l lors(/users/peers)
   B: World
   E: Battery
G: San Discord(https://discord.com/app)
   guillaume@ubuntu:~/0x01$
```

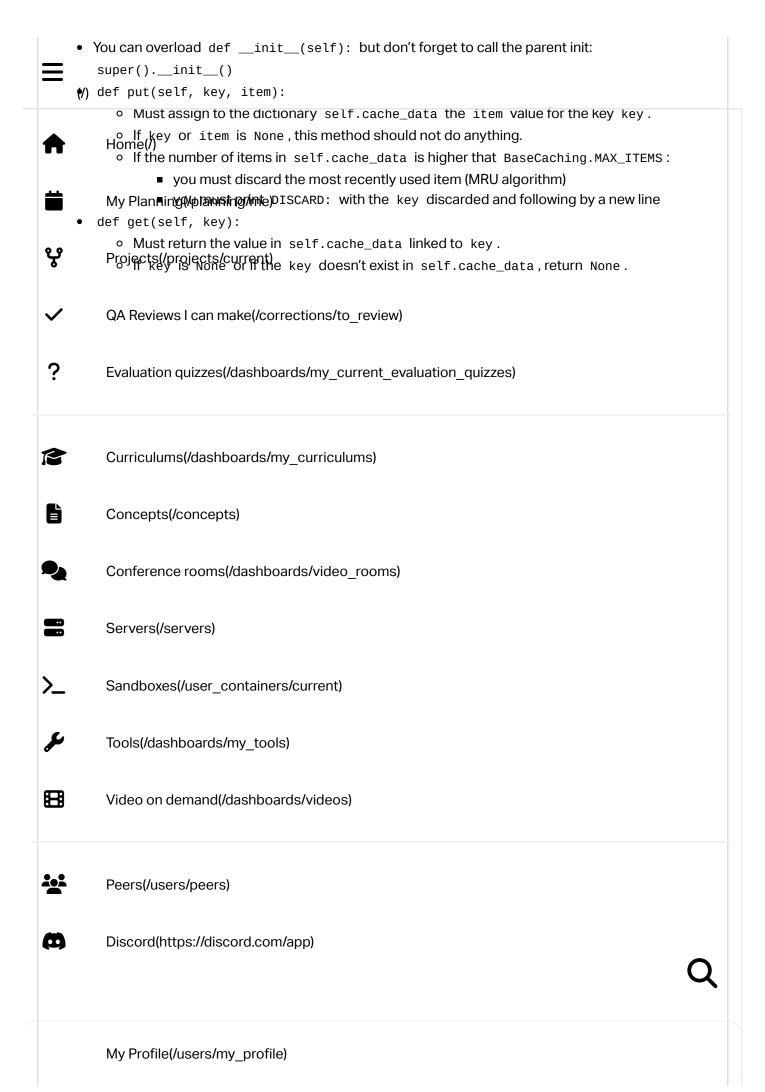
#### Repo:

My Profile(/users/my\_profile)
 GitHub repository: a tx-backend

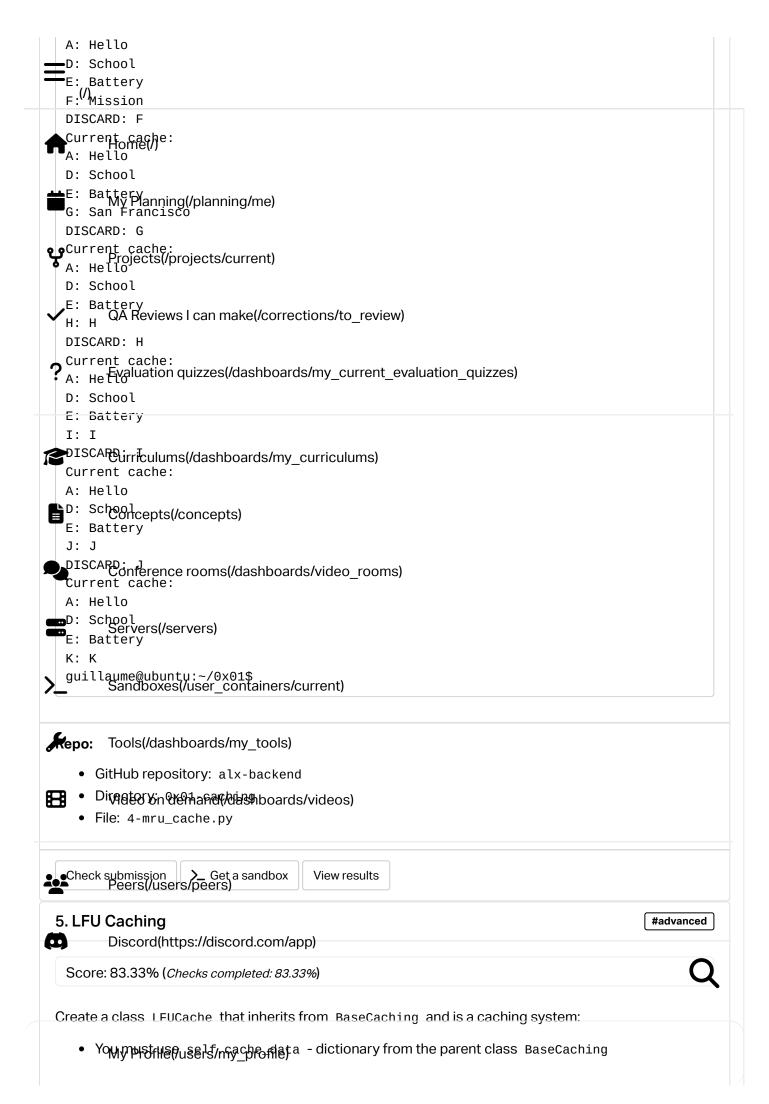


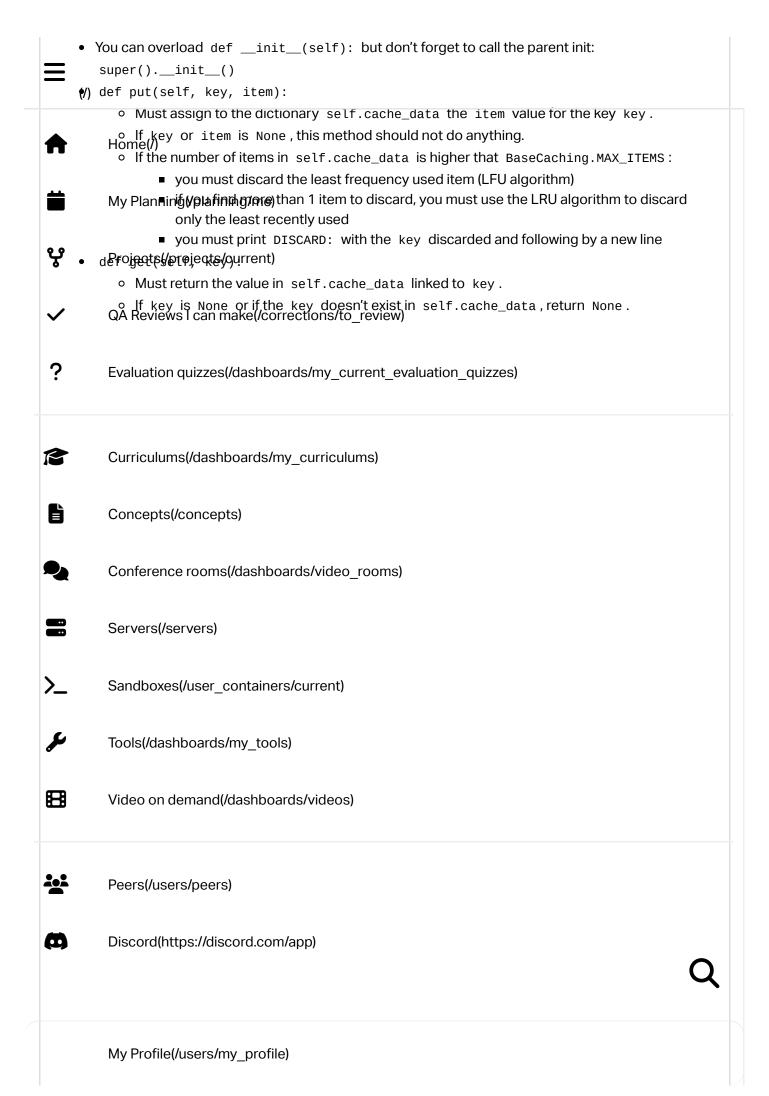
```
guillaume@ubuntu:~/0x01$ cat 3-main.py
#!/usr/bin/python3
  ""("/) 3-main """
  LRUCache = __import__('3-lru_cache').LRUCache
my_cacheme(1) RUCache()
  my_cache.put("A", "Hello")
  my_cache.put("B", "World")
 my_cadvie Polanthing(/plannoingemedn')
  my_cache.put("D", "School")
  my_cache.print_cache()
print (Arpicats/Aprojects/current)
  my_cache.put("E", "Battery")
  my_cache.print_cache()
wy_cacon Reviews | carstnaket/corrections/to review)
  my_cache.print_cache()
  print(my_cache.get("A"))
? print (म्राप्टाप्टकाप्टेन व्यक्टेर्यर्टिं अ/da)shboards/my_current_evaluation_quizzes)
  print(my_cache.get("C"))
  my_cache.put("F", "Mission")
  my_cache.print_cache()
  my_cache.put("G", "San Francisco")
my_cache print_cache/pards/my_curriculums)
  my_cache.put("H", "H")
  my_cache.print_cache()
my_cachoncepts)
  my_cache.print_cache()
  my_cache.put("J", "J")
ֈ
  my_cache.put("K", "K")
  my_cache.print_cache()
        Servers(/servers)
  guillaume@ubuntu:~/0x01$ ./3-main.py
  Current cache:
A: He Sandboxes(/user_containers/current)
  B: World
  C: Holberton
▶D: Schlowols(/dashboards/my tools)
  World
  DISCARD: A
Eurren/tideache demand(/dashboards/videos)
  B: World
  C: Holberton
  D: School
E: Battery
Peers(/users/peers)
  B: World
  C: Street
: Schoocord(https://discord.com/app)
  E: Battery
  None
  World
  Street
  DISCARD: D
  CurrenMy Profile (/users/my_profile)
```





```
guillaume@ubuntu:~/0x01$ cat 4-main.py
#!/usr/bin/python3
  ""("/) 4-main """
  MRUCache = __import__('4-mru_cache').MRUCache
my_cacheme(MRUCache()
  my_cache.put("A", "Hello")
  my_cache.put("B", "World")
 my_cadvie Polanthing(/plannoingemedn')
  my_cache.put("D", "School")
  my_cache.print_cache()
print (Arpicats/Aprojects/current)
  my_cache.put("E", "Battery")
  my_cache.print_cache()
wy_cacon Reviews | carstnaket/corrections/to review)
  my_cache.print_cache()
  print(my_cache.get("A"))
? print (म्राप्टाप्टकाप्टेन व्यक्टेर्यर्टिं अ/da)shboards/my_current_evaluation_quizzes)
  print(my_cache.get("C"))
  my_cache.put("F", "Mission")
  my_cache.print_cache()
  my_cache.put("G", "San Francisco")
my_cache print_cache/pards/my_curriculums)
  my_cache.put("H", "H")
  my_cache.print_cache()
my_cachoncepts)
  my_cache.print_cache()
  my_cache.put("J", "J")
ֈ
  my_cache.put("K", "K")
  my_cache.print_cache()
        Servers(/servers)
  guillaume@ubuntu:~/0x01$ ./4-main.py
  Current cache:
A: He Sandboxes(/user_containers/current)
  B: World
  C: Holberton
▶D: Schlowols(/dashboards/my tools)
  World
  DISCARD: B
Eurren/tideache demand(/dashboards/videos)
  A: Hello
  C: Holberton
  D: School
E: Battery
Peers(/users/peers)
  A: Hello
  C: Street
: Schoocord(https://discord.com/app)
  E: Battery
  Hello
  None
  Street
  DISCARD: C
  CurrenMy Profile (/users/my_profile)
```





```
guillaume@ubuntu:~/0x01$ cat 100-main.py
#!/usr/bin/python3
  ""(//) 100-main """
   LFUCache = __import__('100-lfu_cache').LFUCache
my_cacheme(/) Fucache()
   my_cache.put("A", "Hello")
   my_cache.put("B", "World")
 my_cadvie Polanthing(/plannoingemedn')
   my_cache.put("D", "School")
  my_cache.print_cache()
print (Arpicats/Aprojects/curijent)
   my_cache.put("E", "Battery")
   my_cache.print_cache()
wy_cacon Reviews | carstnaket/corrections/to review)
   my_cache.print_cache()
   print(my_cache.get("A"))
? print (म्राप्टाप्टकाप्टेन व्यक्टेर्यर्टिं अ/da)shboards/my_current_evaluation_quizzes)
   print(my_cache.get("C"))
   my_cache.put("F", "Mission")
   my_cache.print_cache()
  my_cache.put("G", "San Francisco")
my_cache print_cache/pards/my_curriculums)
   my_cache.put("H", "H")
   my_cache.print_cache()
my_cachencepts)
   my_cache.print_cache()
   print(my_cache.get("I"))
ுorint (ந<u>ொ</u>égéநéégééḿśl/dashboards/video_rooms)
   print(my_cache.get("I"))
  print(my_cache.get("H"))
print (Sig/_ve/sel/se/g/e/s)'I"))
   print(my_cache.get("H"))
   my_cache.put("J", "J")
_my_cacfandboxes(basere containers/current)
   my_cache.put("K", "K")
   my_cache.print_cache()
≠my_cacifoeols(Wota(s'hb'oar'ds/n)ny tools)
   my_cache.print_cache()
   my_cache.put("M", "M")
my_cachideprointenand((dashboards/videos)
   guillaume@ubuntu:~/0x01$ ./100-main.py
   Current cache:
   A: Hello
  B: World Peers (/users/peers)
   C: Holberton
   D: School
World Discord(https://discord.com/app)
   DISCARD: A
   Current cache:
   B: World
  C: Holberton
   D: School
   E: BatMy_Profile(/users/my_profile)
```

```
Current cache:
  B: World
  C: Street
   D: (/)School
   E: Battery
None Home(/)
   Street
 DISCARD: D
My Planning(/planning/me)
Current cache:
   B: World
C: Street
Projects(/projects/current)
E: Battery
   F: Mission
  DISCARD: E
QA Reviews I can make(/corrections/to_review)
Current cache:
   B: World
C: Street
F: Mission quizzes(/dashboards/my_current_evaluation_quizzes)
   G: San Francisco
 DISCARD: F
   Current cache:
B: Worddriculums(/dashboards/my_curriculums)
   C: Street
   G: San Francisco
H: H Concepts(/concepts)
   DISCARD: G
   Current cache:
   B: World Conference rooms(/dashboards/video_rooms)
   c: Street
   H: H
⊸I∶ I
         Servers(/servers)
   Н
  Ι
         Sandboxes(/user_containers/current)
H Tools(/dashboards/my_tools)
   Current cache:
C: Street Video on demand(/dashboards/videos)
   I: I
  J: J
  DISCARD: J
currenteers/Nisers/peers)
   C: Street
Discord(https://discord.com/app)
   K: K
   DISCARD: K
   Current cache:
   C: Street
   H: H
         My Profile(/users/my_profile)
```

