

الوي ب برمجة في مقدمة

للكائنات، موجهة نسخة إلى في بوناشري متتالية وظيفية تحويل لفي في عن تحدثنا السابقة، المقالة في طرفية. واجهة بتنفيذ وقمنا

```
# server.py
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/echo', methods=['POST'])
def echo():
    data = request.json
    return jsonify(data)

if __name__ == '__main__':
    app.run(debug=True)

class BaseHandler:
    def handle(self, request:str):
        pass

class Server:
    def __init__(self, handlerClass):
        self.handlerClass = handlerClass

    def run(self):
        while True:
            request = input()
            self.handlerClass().handle(request)

fib_handle.py:

from fib import f
from server import BaseHandler, Server

class FibHandler(BaseHandler):
    def handle(self, request:str):
```

```

n = int(request)
print('f(n)=', f(n))
pass

server = Server(FibHandler)
server.run()

```

بسيط ويب خادم

Web؟ واجهة إلى تحويله يمكنه كيف

عمل كيفية على نظرة نلقي أولاً دعونا HTTP. بـ الخا Server بـ أعلاه Server باستبدال سنقوم
HTTP في .

مدمج. ويب خادم القياسي مكتبة توفر

```
python -m http.server
```

الطرفية. في بتشغيله قم

```
$ python -m http.server
      HTTP      8000 (http://[::]:8000/) ...
```

النتيجة. لرؤية فتحه يمكنك الممتصفح، في

سيكون المرة، هذه الطرفية. إلى ارجع الصفحة، هذه تصفح عند ذلك، بعد الحالي. الدليل محتويات يعرض هذا
للأهم. مثيلاً الأمر

```
$ python -m http.server
      HTTP      8000 (http://[::]:8000/) ...
::1 - - [07/Mar/2021 15:30:35] "GET / HTTP/1.1" 200 -
::1 - - [07/Mar/2021 15:30:35]      404
1:: - - [07/Mar/2021 15:30:35] "GET /favicon.ico HTTP/1.1" 404 -
::1 - - [07/Mar/2021 15:30:35]      404
1:: - - [07/Mar/2021 15:30:35] "GET /apple-touch-icon-precomposed.png HTTP/1.1" 404 -
::1 - - [07/Mar/2021 15:30:35]      404
1:: - - [07/Mar/2021 15:30:35] "GET /apple-touch-icon.png HTTP/1.1" 404 -
::1 - - [07/Mar/2021 15:30:38] "GET / HTTP/1.1" 200 -
```

الإنترنت، على اللكودنم وذج عن ابحت أولاً، بنا. الاختاصة في بوناتشي متتالية خدمة لبناء استخدامه لكي فية بسيّطاً: وي ب خادم الكتب ثم قلليلاً، بت عديله وقم

```
class Handler(SimpleHTTPRequestHandler):  
    def do_GET(self):  
        self.send_response(200)  
        self.send_header('Content-type', 'text')  
        self.end_headers()  
        self.wfile.write(bytes("    ", "utf-8"))  
  
server = HTTPServer(("127.0.0.1", 8000), Handler)  
  
server.serve_forever()
```

نفهم كيف المحلّي. الجهاز على □□□□□□□□ الممنفذ يمثل 8000 والمحلّي، الجهاز عنوان يمثّل 127.0.0.1 هنا،
تعتبرني bytes الخارجيّ. والعالَم المنزل بين اتصال نقطة فهو المنزل، في بنافذة تشبّهه يُمكّن الممنفذ؟
`send_headers` و `send_header` و `send_response` للخصوص. ترميز طريقة هي utf-8 بايتات. إلى النص تحويل
وبهذه فهمه. من المتصفح يتمكن حتى HTTP بروتوكول مع يتوافق محتوى لإخراج تُستخدّم وظائف كلها
الويب. صفحة في hi كلّمة نرى الطريقة،

```
from http.server import SimpleHTTPRequestHandler, HTTPServer
from fib import f
from urllib.parse import urlparse, parse_qs
```

```
from http.server import SimpleHTTPRequestHandler, HTTPServer
from fib import f
from urllib.parse import urlparse, parse_qs
```

عادةً. تترجم هذه التي تم لا وال تي ، بلغة ووظائف مكتبات أسماء على يحتوي لأنه يتغير لم الكود ملاحظة:

```
class Handler(SimpleHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type', 'text')
        self.end_headers()
        parsed = urlparse(self.path)
        qs = parse_qs(parsed.query)
        result = ""
        if len(qs) > 0:
            ns = qs[0]
            if len(ns) > 0:
                n = int(ns)
                result = str(f(n))
            self.wfile.write(bytes(result, "utf-8"))
```

إلى: أعلاه الكود تترجم تم

```
class Handler(SimpleHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type', 'text')
        self.end_headers()
        parsed = urlparse(self.path)
        qs = parse_qs(parsed.query)
        result = ""
        if len(qs) > 0:
            ns = qs[0]
            if len(ns) > 0:
                n = int(ns)
                result = str(f(n))
            self.wfile.write(bytes(result, "utf-8"))
```

لما عليها الحفاظ ويجب بالإنجليزية، ومتغيرات دوال أسماء على يحتوي لأنه يتغيره يتم لم الكود ملاحظة:
هي.

```
server = HTTPServer(("127.0.0.1", 8000), Handler)
```

```
server.serve_forever()
```

المعاملات. بعض تحليلي يتم هنا كذلك؟ أليس الشيء، بعض معقد

```
self.path=?n=3
parsed=ParseResult(scheme='', netloc='', path='/', params='', query='n=3', fragment='')
qs={'n': ['3']}
ns=['3']
n=3
```

الذاتي [] [] الاس تدعاء التكرار مع التعامل في التقديم

للكود. بسيسة هيكلية بإعادة نقوم دعونا

```
from http.server import SimpleHTTPRequestHandler, HTTPServer
from fib import f
from urllib.parse import urlparse, parse_qs
```

```
class Handler(SimpleHTTPRequestHandler):
```

ترجمة:

```
class Handler(SimpleHTTPRequestHandler):
```

هذه SimpleHTTPRequestHandler ال فئدة من تراث وال تي Handler باسم جديدة فئدة تعريفي يتم الكود، هذا في ال بسيسة. الوي بخوادم في [] [] [] [] طلبات لمعالجة عادة تُستخدم ال فئدة

```
def parse_n(self, s):
    parsed = urlparse(s)
    qs = parse_qs(parsed.query)
    if len(qs) > 0:
        ns = qs['n']
        if len(ns) > 0:
            n = int(ns[0])
            return n
    return None
```

```
def do_GET(self):
    self.send_response(200)
    self.send_header('Content-type', 'text')
    self.end_headers()
```

أي إلى بحاجة كنت إذا هي. كما ال إنجليزية الأسماء على الحفاظ مع العربية اللغة إلى أعلاه الكود ترجمة تمت
إعلامي! في تتردد فلا إضافية، تعديلات

```
result = ""
n = self.parse_n(self.path)
if n is not None:
    result = str(f(n))

self.wfile.write(bytes(result, "utf-8"))
self.wfile.write(bytes(result, "utf-8"))
```

إلى: أعلاه الكود ترجمة تمت

```
result = ""
n = self.parse_n(self.path)
if n is not None:
    result = str(f(n))

self.wfile.write(bytes(result, "utf-8"))
self.wfile.write(bytes(result, "utf-8"))
```

ترجمته. يتم لا والتي بال إنجليزية، ومتغيرات دوال أسماء على يحتوي لأن ي تغير لم الكود ملاحظة:

```
server = HTTPServer(("127.0.0.1", 8000), Handler)

server.serve_forever()
```

الطلب. مسار من n القديمة استخراج عملية لتغلي في parse_n دالة إدخال تم

وبعد في بوناتي، متسلسلة في 10000 رقم العدد 0000 0000 طلب التالية: المشكلة ليه البرنامج
انتظر المرتين، كلتا في في بوناتي. متسلسلة في 10000 رقم العدد أيضاً 0000 0000 طلب أيام، بضعة
كفاءة تحسني يمكننا لكي في النتيجة. على الحصول قبل طويلة لفترة 0000 و 0000 0000 من كل
هذه؟

10000 رقم العدد حساب من بدلاً. الموقتات التخزين تقنيات استخدام يمكننا الكفاءة، لتحسين ذاكرة في وتخزينها واحدة مرة النتيجة حساب يمكننا الطلب، في هذا يتم مرة لكل في في بوناتشي متسلسلة في الموقتات التخزين من النتيجة استرجاع ببساطة يمكننا أخرى، مرة العدد نفس طلب يتم عندما الموقتات التخزين حسابها. إعادة من بدلاً

و: @lru_cache باستخدام بسيط مثال إليكم

```
from flask import Flask, jsonify
from functools import lru_cache

app = Flask(__name__)

@lru_cache(maxsize=None)
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

@app.route('/fibonacci/<int:n>', methods=['GET'])
def get_fibonacci(n):
    result = fibonacci(n)
    return jsonify({'n': result})

if __name__ == '__main__':
    app.run(debug=True)
```

عند أنه يعني هذا مؤقتًا. fibonacci الدالة نتائج لتخزين @lru_cache الذي لوراتور نستخدم المثال، هذا في يحسن مما حسابها، إعادة من بدلاً الموقتات التخزين من النتيجة استرجاع سيتم أخرى، مرة العدد نفس طلب الخدمة. كفاءة من كابير بشكل

سيتم في بوناتشي، متسلسلة في 10000 رقم العدد أو 10000 10000 يطلب عندما الطريقة، بهذه أكبر. بسرعة النتيجة على الحصول

التجارب. بعض إجراءات من نفسها. دائماً ستكون $f(n)$ قيمة فإن متشابهة، n قيمة كانت إذا أنه لاحظنا

```
127.0.0.1 - - [10/Mar/2021 00:33:01] "GET /?n=1000 HTTP/1.1" 200 -
```

```
-----
) '127.0.0.1', 50783)
```

```
Traceback ) : (
```

```
...
```

```
if v[n] != -1:
```

```
IndexError:
```

10000 إلى v مصفوفة فلنغير الحجم، في كافيّة غير المصفوفة لكانت إذا

```
v = []
```

```
for x in range(10000):
```

```
    v.append(-1)
```

إلى: أعلاه الكود ترجمة تمت

```
v = []
```

```
for x in range(10000):
```

```
    v.append(-1)
```

for حلقة باستخدام مرة 10000 القائمة إلى -1 القائمة إضافة يتم ثم v، فارغة قائمة إنشاء يتم الكود، هذا في -1. هو منها عنصر لكل عنصر، 10000 على تحتوي قائمة هي النهائية النتيجة

الذاتي الاستدعاء عمق تجاوز في خطأ ظهر، 2000 تساوي n تكون عندهم ذلك، ومع
الذاتي الاستدعاء عمق تجاوز في خطأ ظهر، 2000 تساوي n تكون عندهم ذلك، ومع

```
127.0.0.1 - - [10/Mar/2021 00:34:00] "GET /?n=2000 HTTP/1.1" 200 -
```

```
-----  
( '127.0.0.1', 50821)
```

```
Traceback ) : (
```

```
...
```

```
if v[n] != -1:
```

```
RecursionError:
```

ما. حد إلى سرياً كان هذا كل ذلك، ومع

قد، f(1000) حساب عند أنه يعني هذا فقط. واحدة مرة ثحسب أن إلى تحتاج كلها، f(1000) إلى f(1) لأن لم إذا؟ أنه يعني وهذا. 1000 حوالي بلغ Python في التكرار عمق أن نعلم فقط. مرة 1000 حوالي + عمليّة تنفيذي يتم حدوث إلى يؤدي قد هذا ل، 1000 أولاً فسنحسب، 2000 حساب أردنا إذا الطريقة: بهذه البرنامج تحسبن يمكننا 400 أولاً فلنحسب، 1200 حساب أردنا وإذا 1200 أولاً فلنحسب، 2000 حساب أردنا إذا .

خطأ يحدث ولن، 800 حوالي الذاتي الاستدعاء عمق سيكون، 2000 حساب ثم الطريقة، بهذه و1200 400 حساب بعد الذاتي. الاستدعاء عمق تجاوز في


```

v = []
for x in range(1000000):
    v.append(-1)

def fplus(n):
    if n > 800:
        fplus(n-800)
    return f(n)
else:
    return f(n)

def f(n):
    if v[n] != -1:
        return v[n]
    else:
        a = 0
        if n < 2:
            a = n
        else:
            a = f(n-1) + f(n-2)
        v[n] = a
    return v[n]

```

fplus دالة إضافة تمت

800 1000 ؟ مرة 1000 متكرر بشكل fplus الدالة استدعاء تم لو ماذا يتساءل، أن إلا المرء يسرع لا ذلك، ومع الاستمرار بعد الاستدعاءات. عمق في مشكلة أخرى مرة ظهرت ألف، 800 إلى n قيمة بتعيني قيمت عن دما. 800000. سهلاً أمراً 2000 حساب أصبح التحسينات، هذه بعد ذلك، ومع تعقيداً. أكثر الأمور أن اكتشفت التجربة، في للغاية.

الملفات وكتابة قراءة

الطلب وفي، f(400) الأول الطلب في الوبي. تطويع موضوع إلى فلنعد قليلاً. انحراف قد الموضوع أن يبدو ومع الأول. الطلب في إنشاؤها تم التي v مصفوفة قيم استخدمنا الثاني، الطلب عند f(600) الثاني متسلسلة حساب لطريقتنا، وفقاً استخدمنا. من نتمكن لن تشغله، ونعيد البرنامج من نخرج عن دما ذلك، الكثير هناك وكان، v مصفوفة أدخلنا قد نكن لم عن دما خاصة بطيئاً. كان إذا تخيل ولكن جذاً، سرع في بوناتي بصعوبة. عليها حصلنا التي النتائج حفظ في نرغب الحالة، هذه في المتكررة. الحسابات من

موجودة فهي ذلك، ومع مؤقت. تخزين بمثابة تعتبر هنا v المصفوفة. `[[[[[[]]]]]]` مفهوماً دورياً هنا
نفسه الحال، بطبيعة الحال؟ ما إذن، الـ بيانات. هذه تختفي الـ برنامج، إغلاق بمجرّد البرنامج. حياة دورة خلّال فقط
ملف. في تخزينها في

ملف؟ إلى v مصفوفة حفظ يمكن لأي

```
0 0
1 1
2 1
3 2
4 3
...
```

ربما طبيعي، بشكل يتزايد n لأن نظراً $f(n)$ لـ صف لكل حفظ يتم الطريقة. بهذه v مصفوفة حفظ يمكننا
 $f(n)$ قيم حفظ فقط يمكننا

```
0
1
1
2
3
...
```

الآن. جربها

```
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()
```

الكود: ترجمة

```
f = open("demofile2.txt", "a") # "demofile2.txt" (append)
f.write("Now the file has more content!") # "Now the file has more content!"
f.close() #
```

إضافة سيتم أنه يعني ممّا، `"a"` إلحاق وضع في `demofile2.txt` باسم نصيّاً ملفّاً يفتح الكود ملحوظة:
الملف. إغلاق يتم ذلك، بعد الحالّي. المحتوى حذف دون الملف نهاية إلى الجديد النص

```
#
:
f = open("demofile2.txt", "r")
print(f.read())
```

مما w، أو الملف؛ نهاية في المحتوى إضافة سيتم أنه يعني مما a، يكون أن يمكن open للدلالة الثاني الوسيط
الحالي. الملف محتوى استبدال سيتم أنه يعني

```
file = open('fib_v', 'a')
file.write('hi')
file.close()
```

fib_v باسم ملفاً بالفعل وجدت التشغيل، بعد

```
`fib_v`
```

هكذا. يصبح أخرى، مرة بتشغيله نقوم عندم

```
hihi
```

جديد؟ سطر إدراج يمكنني كيف

```
file = open('fib_v', 'a')
file.write('hi\n')
file.close()
```

أخرى، مرة الطباعة عند ذلك، ومع جديد. سطر أي ملاحظة يتم ولم hihhi، وستظهر واحدة، مرة طباعته سيتم هذا
في كان ولكنه الأولى، المرة في بالفعل طباعته تم قد الجديد السطر أن على يدل وهذا جديد. سطر إدراج يتم
مرئياً. يكن ولم النهاية

القراءة؟ يمكنني كيف

```
file = open('fib_v', 'r')
print(file.read())
```

محتوى يطبع ثم 'r' للقراءة fib_v باسم ملف بفتح وي قوم بلغة مكتوب أعلاه الكود ملاحظة:
ترجمة. إلى يحتاج ولا المدي برمجية بلغة مكتوب لأن الكود في تغيري وجد لا الملف.

```
$ python fib.py
hihihi
hi
```

في بوناتشي. مدت الية لحساب برنامجنا نعدل دعونا ذلك، بعد

```
v = []
for x in range(1000000):
    v.append(-1)

def read():
    file = open('fib_v', 'r')
    s = file.read()
    if len(s) > 0:
        lines = s.split('\n')
        if (len(lines) > 0):
            for i in range(len(lines)):
                v[i] = int(lines[i])
```

العربية: إلى الكود ترجمة

```
def read():
    file = open('fib_v', 'r') # 'fib_v'
    s = file.read() #
    if len(s) > 0: #
        lines = s.split('\n') #
        if (len(lines) > 0): #
            for i in range(len(lines)): #
                v[i] = int(lines[i]) # v
```

الملف. من قراءتها يتم التي القيم لتخزين مسبقًا مُعرفة v قائمة وجود يفترض الكود ملأحظة:

```
def save():
    file = open('fib_v', 'w')
    s = ''
    start = True
    for vv in v:
```

```

        if vv == -1:
            break
        if start == False:
            s += '\n'
            start = False
        s += str(vv)
    file.write(s)
    file.close()

def fcache(n):
    x = fplus(n)
    save()
    return x

def fplus(n):
    if n > 800:
        fplus(n-800)
        return f(n)
    else:
        return f(n)

def f(n):
    if v[n] != -1:
        return v[n]
    else:
        a = 0
        if n < 2:
            a = n
        else:
            a = f(n-1) + f(n-2)
        v[n] = a
        return v[n]

read()
fcache(10)
save()

```

التالي. الانح و على fib_v ملف أصبح ال برن امج، تشغيل بعدد ال برن امج. كتابة من ان تهين اخيرا،

fib_v:

0
1
1
2
3
5
8
13
21
34
55

توحيّدًا؟ وأكثر أبسط تحليل طريقة هناك هل جديّد. سطر رمز هو \n الشيء. بعض مَعقد أعلاه التحليل أن نرى JSON. البيانات تنسيق الناس ابتكر لقد

□□□□

JSON على مثال يلي فيم JavaScript Object Notation. هو □□□□ لـ الكامل الاسم

```
{"name": "John", "age": 31, "city": "New York"}
```

الطريقة. بهذه التعيين عن التّعبيريمكن

التالية: الأساسية العنصر على □□□□ تحتوي

1. النصرية السلسلة أو الأرقام
2. القوائم
3. الخرائط □□ التعيينات

ويمكن أخرى. قائمة على القوائم تحتوي أن يمكن أي تعسفي. بشكل الأساسية العنصر هذه تداخل أيضًا ويمكن وهكذا. أيضًا. قائمة على الخريطة تحتوي أن

```
{  
  "name": "John",  
  "age": 30,  
  "cars": [ "Ford", "BMW", "Fiat" ]  
}
```

XX

json بتنسيق نصية سلسلة إلى v مصفوفة تحوّل إلى نحتاج ذلك، بعد

```
import json

v = []
for x in range(1000000):
    v.append(-1)

def fplus(n):
    if n > 800:
        fplus(n-800)
    return f(n)
    else:
        return f(n)

def f(n):
    if v[n] != -1:
        return v[n]
    else:
        a = 0
        if n < 2:
            a = n
        else:
            a = f(n-1) + f(n-2)
        v[n] = a
        return v[n]

fplus(100)
s = json.dump(v)
file = open('fib_j', 'w')
file.write(s)
file.close()
```

الطريقة. بهذه الدالة تعرّف رؤية يمكننا، في 'fp'.
الطريقة. بهذه الدالة تعرّف رؤية يمكننا، في 'fp'.
الطريقة. بهذه الدالة تعرّف رؤية يمكننا، في 'fp'.

مريحاً؟ ذلك أليس فقط. dump فوق الماوس تحريك يمكنك

```

def f(n):
    (function) dump: (obj: Any, fp: IO, *, skipkeys: bool = ..., ensure_ascii: bool = ...,
    if v[n check_circular: bool = ..., allow_nan: bool = ..., cls: Type | None = ..., indent: int | str |
    ret None = ..., separators: Tuple | None = ..., default: (_p0: Any) -> Any | None = ..., sort_keys:
    else: bool = ..., **kwds: Any) -> None
    a =
    if Serialize obj as a JSON formatted stream to fp (a .write() -supporting file-like object).
    els If skipkeys is true then dict keys that are not basic types ( str , int , float , bool , None ) will be skipped instead
    of raising a TypeError .
    v[n If ensure_ascii is false, then the strings written to fp can contain non-ASCII characters if they appear in strings
    ret contained in obj . Otherwise, all such characters are escaped in JSON strings.
    If check_circular is false, then the circular reference check for container types will be skipped and a circular reference
    fplus(100 will result in an OverflowError (or worse).
    s = json.dump(v)
    file = open('fib_j', 'w')
    file.write(s)
    file.close()

```

000000 1: 0000

```

fplus(10)
file = open('fib_j', 'w')
json.dump(v, file)
file.close()

```

تم ريدي مكن ال أصل، في 10. إلى بتغيي يدها سنقوم لذا الأرقام، من كبي ر عدد عرض سي تم، 100 حتى ال أرقام لحساب
dump. للدالة ثاني كمعامل file ال كائن

الطريقة: بهذه الملاحظات رؤية ي مكنك

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, -1, -1, -1]
```

لاحقًا. حذفها تم -1 من ال عديدهن الك أن لاحظ

```

def read():
    file = open('fib_j', 'r')
    s = file.read()
    sv = json.loads(s)
    for i in range(len(sv)):
        if sv[i] != -1:
            v[i] = sv[i]
def save():
    file = open('fib_j', 'w')
    json.dump(v, file)
    file.close()

```


ترجمة:

```
def read():
    file = open('fib_j', 'r')
    s = file.read()
    sv = json.loads(s)
    for i in range(len(sv)):
        if sv[i] != -1:
            v[i] = sv[i]
def save():
    file = open('fib_j', 'w')
    json.dump(v, file)
    file.close()
```

قراءة

```
for vv in v:
    if vv != -1:
        print(vv)
```

تظهر: ال طباعة أن نرى أن يمكننا هذا، يحدث عن دم

0
1
1
2
3
5
8
13
21
34
55

مغاً: الدوال هذه من نتحقق دعونا

```
def read():
    file = open('fib_j', 'r')
```

```

s = file.read()
sv = json.loads(s)
for i in range(len(sv)):
    v[i] = sv[i]

```

إلى: الكود مترجمة تمت

```

def read():
    file = open('fib_j', 'r')
    s = file.read()
    sv = json.loads(s)
    for i in range(len(sv)):
        v[i] = sv[i]

```

ترجمته. يتم ولا `print` برمجة بلغة مكتوب لأنه هو كما يبقى الكود ملاحظة:

```

def save():
    sv = []
    for i in range(len(v)):
        if v[i] != -1:
            sv.append(v[i])
        else:
            break
    file = open('fib_j', 'w')
    json.dump(sv, file)
    file.close()

```

fplus(100)

save()

مرتب. وبشكل الصحيح، القيم حفظ تم وبالفعل العرض، ملف إلى انتقلت ثم

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711]

البيانات قاعدة

بطيء! سيصبح الملفات حفظ طريقة استخدام نفع؟ ماذا الـ كل، في ومعدة جدًا لكبيرة البيانات كانت إذا وحذفها البيانات إضافة بسهولة يمكن للبرمجة. قابل Excel تعادل وهي . دور يأتي هنا ومعدًا. Excel مثل الكود، باستخدام فيها والبحث وتعديلها.

```
import sqlite3
con = sqlite3.connect('example.db')
cur = con.cursor()
```

000.0000000000000000 00000 000000 00000 0000, 00000 0000, 000000 0000, 000 0000,
00000 000000 00

000.0000000000000000 000000 000000 000000 0020060105,00000,000000,100,35.1400

□□□□.□□□□□□□□

حسب مرتبة جدول من الصفوف بطباعة وي قوم بلغة مكتوب أعلاه المذكور الكود ملحظة:
هي. كما تبقى أن يجب برمجية أوامر على يحتوي لأنه الكود ترجمة يتم لم السعر.

```
import sqlite3
```

```

v = []
for x in range(1000000):
    v.append(-1)

def create_table(cur: sqlite3.Connection):
    cur.execute('CREATE TABLE vs(v text)')

```

إلى: أعالاه الكود ترجمة تمت

```

def create_table(cur: sqlite3.Connection):
    cur.execute('CREATE TABLE vs(v text)')

```

ال نص نوع من واحد عمود على يحتوي vs باسم جديد جدول إنشاء يتم الدالة، هذه في `text`.

```

def read():
    pass

def save():
    con = sqlite3.connect('fib.db')
    cur = con.cursor()
    create_table(cur)
    for vv in v:
        if vv != -1:
            cur.execute('INSERT INTO vs VALUES(' + str(vv) + ')')
        else:
            break
    con.commit()
    con.close()

```

save()

إلى: أعالاه الكود ترجمة سيتم

save()

قد وال ترجمة معينة، برمجة بلغة خاصة تكون قد برمجة أوامر على يحتوي لأنه ترجمته يتم لم الكود ملاحظه:
الكود. وظيفة في تغيري إلى تؤدي

الآن. جربها الكتابة، تمت

بي. الخاص الكمبيوتري جهاز على sqlite3 بالفعل لدي

```

$ sqlite3
SQLite version 3.32.3 2020-06-18 14:16:19
    ".help"
    ".open FILENAME"

sqlite> .help
.auth ON|OFF
.backup ?DB? FILE          DB          ) "main")          FILE
.bail on|off                .          OFF
.binary on|off              .          OFF
.cd DIRECTORY              DIRECTORY
.changes on|off            SQL
.check GLOB                .testcase
.clone NEWDB              NEWDB
.databases
.dbconfig ?op? ?val?      sqlite3_db_config()
.dbinfo ?DB?
.dump ?TABLE?            SQL
.echo on|off
.eqp on|off|full|...      EXPLAIN QUERY PLAN
.excel
.exit ?CODE?              CODE
.expert
.explain ?on|off|auto?    EXPLAIN. :      auto
.filectrl CMD ...         sqlite3_file_control()
.fullschema ?--indent?    sqlite_stat
.headers on|off
.help ?-all? ?PATTERN?    PATTERN
.import FILE TABLE       FILE          TABLE
.imposter INDEX TABLE    TABLE          INDEX
.indexes ?TABLE?
.limit ?LIMIT? ?VAL?      SQLITE_LIMIT
.lint OPTIONS
.log FILE|off             .          stderr/stdout
.mode MODE ?TABLE?
.nullvalue STRING         STRING          NULL

```

```

.once ?OPTIONS? ?FILE?          SQL          FILE
.open ?OPTIONS? ?FILE?          FILE
.output ?FILE?                  FILE  stdout  FILE
.parameter CMD ...              SQL
.print STRING...                STRING
.progress N                     N
.prompt MAIN CONTINUE
.quit
.read FILE                      FILE
.recover
.restore ?DB? FILE              DB          ) "main")  FILE
.save FILE                      FILE
.scanstats on|off               sqlite3_stmt_scanstatus()
.schema ?PATTERN?              CREATE        PATTERN
.selftest ?OPTIONS?            SELFTEST
.separator COL ?ROW?
.session ?NAME? CMD ...
.sha3sum ...                    SHA3
.shell CMD ARGS...             CMD ARGS...  shell
.show
.stats ?on|off?
.system CMD ARGS...            CMD ARGS...  shell
.tables ?TABLE?                LIKE TABLE
.testcase NAME                  'testcase-out.txt'
.testctrl CMD ...              sqlite3_test_control()
.timeout MS                     MS
.timer on|off                   SQL
.trace ?OPTIONS?               SQL
.vfsinfo ?AUX?                 VFS
.vfslist                       VFSes
.vfsname ?AUX?                 VFS
.width NUM1 NUM2 ...           "column"

```

الخروج. يعني الانني quit. بينها من الاوامر. من العديد رؤية يمكنك

brew install sqlite الأمر تشغيل أو الرسم، الموقع من تنزيله يمكنك مثبتًا، لديك يمكن لم إذا
ل تثبت به.

```
$ sqlite3 fib.db
```

```
sqlite> show tables
```

```
...> ;
```

```
:          "show":
```

```
sqlite> tables;
```

```
:          "tables":
```

```
sqlite> .schema
```

```
CREATE TABLE vs(v text);
```

لكن المتوفرة. الجداول لمعرفة `show tables` استخدم يمكنك. MySQL يشبه الأمر أن اعتقدت البداية في
في سنتعلمه ما أيضاً وهو البيانات، قواعد من آخر نوع هو MySQL مختلف. SQLite في الأمر أن اكتشفت لاحقاً
المستقبل.

```
sqlite> select * from vs;
```

```
0
```

```
1
```

```
1
```

```
2
```

```
3
```

```
5
```

```
8
```

```
13
```

```
21
```

```
34
```

```
55
```

نوع يتمكن لا وقد كبيرة أرقامنا لأن `text` استخدمنا أننا لاحظ صحيح. بشكل البيانات بكتابة قمننا بالفعل،
حفظها. من البيانات قاعدة في الصحيحة الأعداد

```
import sqlite3
```

```
v = []
```

```
for x in range(1000000):
```

```
    v.append(-1)
```

```
def fplus(n):
```

```
    if n > 800:
```

```

        fplus(n-800)
        return f(n)
    else:
        return f(n)

def f(n):
    if v[n] != -1:
        return v[n]
    else:
        a = 0
        if n < 2:
            a = n
        else:
            a = f(n-1) + f(n-2)
        v[n] = a
        return v[n]

def create_table(cur: sqlite3.Connection):
    cur.execute('CREATE TABLE vs(v text)')

def read():
    con = sqlite3.connect('fib.db')
    cur = con.cursor()
    create_table(cur)
    i = 0
    for row in cur.execute('SELECT * from vs'):
        v[i] = int(row)
    con.close()

```

إلى: الكود تترجمه تمت

```

def read():
    con = sqlite3.connect('fib.db')
    cur = con.cursor()
    create_table(cur)
    i = 0
    for row in cur.execute('SELECT * from vs'):

```



```

        v[i] = int(row)
    con.close()

```

وعدم استخداًه، قبل `v` المتغير تعريفي عدم مثل المحتمة، الأخطاء بعض على يحتوي الأصلي الكود ملاحظة:
الترجمة. في تصحيحها يتم لم الأخطاء هذه الحلقة. داخل `i` قيمة زيادة

```

def save():
    con = sqlite3.connect('fib.db')
    cur = con.cursor()
    create_table(cur)
    for vv in v:
        if vv != -1:
            cur.execute('INSERT INTO vs VALUES(' + str(vv) + ')')
        else:
            break
    con.commit()
    con.close()

read()
for i in range(10):
    print(v[i])

```

خطأ. ظهر التشغيل، بعد ذلك، ومع `read` دالة بإضافة نس تمر

```

$ python fib_db.py
...
File "fib_db.py", line 27, in create_table
    cur.execute('CREATE TABLE vs(v text)')
sqlite3.OperationalError:      vs

```

قليلاً. الصيغة بتعديّل سنقوم بالفعل. موجد الجدول أخرى، مرة الجدول إنشاء يمكننا لا

```

def create_table(cur: sqlite3.Connection):
    cur.execute('CREATE TABLE IF NOT EXISTS vs(v text)')

```

الترجمة:

```

def create_table(cur: sqlite3.Connection):
    cur.execute('CREATE TABLE IF NOT EXISTS vs(v text)')

```

الكود نفس هي هنا والترجمة، `int()` برمجة بلغة مكتوب أنه حيث ترجمة إلى يحتاج لا المقدم الكود ملاحظة:
الأصلي.

خطأ. حدث ذلك، ومع

```
v[i] = int(row)
TypeError: int() argument must be string, bytes or bytearray, not 'tuple'
```

نطبعها. دعنا `tuple`. تُرجع `row` الدالة أن يعنى `tuple` في البيانات أنواع من نوع هو `tuple`

```
for row in cur.execute('SELECT * from vs'):
    print(row)
    v[i] = int(row)
```

هي: النتيجة

```
('0',)
```

مختلفة تكون أن يمكن `tuple` عن اصر أن هو الرئي سي الفرق المصفوفة. كبيرحد إلى يشبه `tuple` الواقع، في
النوع. نفس من عن اصرها جميع تكون أن يجب التي المصفوفة عكس على البعض، بعضها عن

```
def read():
    con = sqlite3.connect('fib.db')
    cur = con.cursor()
    create_table(cur)
    i = 0
    for row in cur.execute('SELECT * from vs'):
        v[i] = int(row[0])
    con.close()
```

تغييري تم لا `tuple` بيانات قاعدة من البيانات بقراءة ويقوم `tuple` بلغة مكتوب أعلاه الكود ملاحظة:
الترجمة. أثناء الكود

هكذا: الينات يكون أن الغريب من ذلك، ومع التعتدي. هذا

55

-1

-1

-1

-1
-1
-1
-1
-1
-1

زيادته. يتم لم i الم تغيّر أن اتضح

```
for row in cur.execute('SELECT * from vs'):
    v[i] = int(row[0])
    i += 1
```

صحيح. هذا

0
1
1
2
3
5
8
13
21
34

الشكل: بهذا ال بيانات قاعدة في حفظها يتم كبيّرة، الأرقام تكون عندما أنه لاحظنا ذلك، ومع

4660046610375530309
7540113804746346429
1.22001604151219e+19
1.97402742198682e+19
3.19404346349901e+19

كالتالي. النتيجة ستكون التشغيّل، إعادة عن

```
$ python fib_db.py
```

```
Traceback (most recent call last):
```

```
File "fib_db.py", line 35, in read
    v[i] = int(row[0])
ValueError: invalid literal for int() with base 10: '1.22001604151219e+19'
```

يلي: كما العربيه إلى أعلاه النص ترجمه تمت

```
$ python fib_db.py
Traceback (most recent call last):
  File "fib_db.py", line 35, in read
    v[i] = int(row[0])
ValueError: int() :10 '1.22001604151219e+19'
```

عذل أو غيّر

```
cur.execute("INSERT INTO vs VALUES('" +str(vv) + "'")")
```

تحويل يتم INSERT INTO. أمر باسخدام vs يسمى جدول في vv المتغير قيمة إدراج يتم أعلاه، الكود في الاستعمال. في إدراجها قبل str() الدالة باسخدام نصية سلسلة إلى vv قيمة

وأضفنا مزدوجة، اقتباسات إلى INSERT عبارة حول الفردية الاقتباسات بتغيير قمنها أننا لاحظ قاعدة لكانت قبل، من الطريقة بهذه كتبنا قد لنا إذا بنا. الخاصة الأرقام سلسلة حول اقتباسات أيضاً مع هذا التعامل يتم الاقتباس، علامات باسخدام الآن، ولكن أرقام، أنها على السلسلة مع تعاملت قد البيئات نصية. سلسلة أنها على

السابقة؟ الخاطئة البيئات مسح يمكنه ذلك، ومع صحيحاً. أصبح ثم

```
$ sqlite3 fib.db
SQLite version 3.32.3 2020-06-18 14:16:19
    ".help"
sqlite> delete * from vs;
```

على أمثلة هنا قدمنا لقد . ذلك. بعد أخرى عبارات تجربة يمكنك

التمرين

أعلاه. ورد لما مشابه بشكل الأمور استكشف الطلاب على يجب