

תשובות:

1.

א.24

ב. את סכום הספרות שבו

ג. הפקודה שנכשלה היא הפיכת הסטרינג מהאינפוט (הביטוי האריתמטי) לאינטגר.

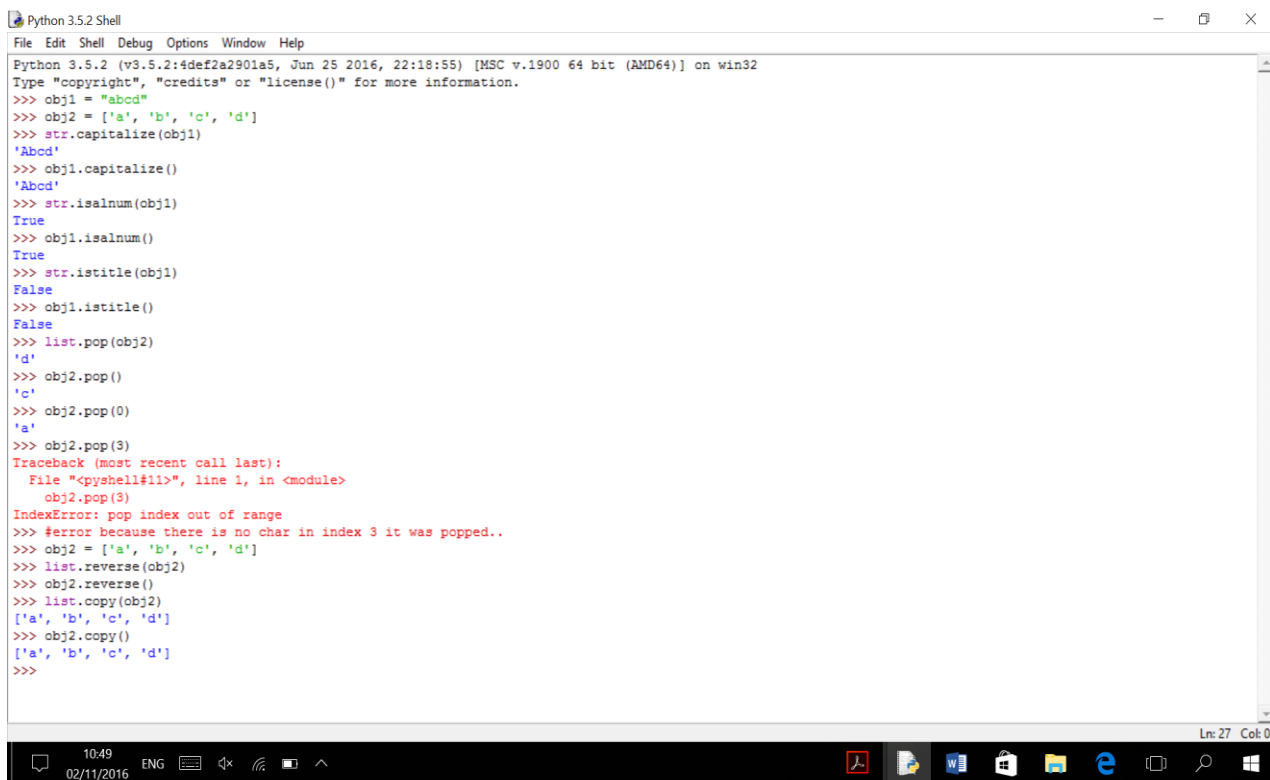
הבעיה - הסטרינג היחיד ש `int` יכול להפוך למספר הוא סטרינג שבתוכו אך ורק מספר שלם.

וסטרינג שבתוכו ביטוי אריתמטי אינו סטרינג שבתוכו מס' שלם בלבד.

ד. 2596148429267413814265248164610136

ה. תווצר לולאה אינסופית כי כש `num` יהפוך ל0 (ע"י החילוק ב10) הלולאה תמשיך שוב במקרה הזה, `num` ימשיך להתחלק ב 10 וישאר 0 וכך הלולאה תפעל בלי סוף.

2.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> obj1 = "abcd"
>>> obj2 = ['a', 'b', 'c', 'd']
>>> str.capitalize(obj1)
'Abcd'
>>> obj1.capitalize()
'Abcd'
>>> str.isalnum(obj1)
True
>>> obj1.isalnum()
True
>>> str.istitle(obj1)
False
>>> obj1.istitle()
False
>>> list.pop(obj2)
'd'
>>> obj2.pop()
'c'
>>> obj2.pop(0)
'a'
>>> obj2.pop(3)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    obj2.pop(3)
IndexError: pop index out of range
>>> #error because there is no char in index 3 it was popped..
>>> obj2 = ['a', 'b', 'c', 'd']
>>> list.reverse(obj2)
>>> obj2.reverse()
>>> list.copy(obj2)
['a', 'b', 'c', 'd']
>>> obj2.copy()
['a', 'b', 'c', 'd']
>>>
```

3.

4.

א.

זמני הריצה : עבור הפקודה הראשונה קודם ואז השנייה, כשגודל המס גודל בכל אחת

ראשונה:

0.00013727704774803634

0.00020655705315358527

0.0005696356000012004

0.002235349310214907

שנייה:

2.009975465469549e-05

3.3784693994065584e-05

4.9607905105206185e-05

8.852445135154596e-05

כל פעם הזמן גודל פי מס' גדול יותר מהקודם

כלומר קצב הגדילה הוא אינפנטסימלי כתלות בגודל הקלטים (שגם גדולים פי מספר גדול יותר מהקודמים להם מאשר אלו-לקודמים להם).

ב. 1.2829630630678546e-05

1.325728498502432e-05

1.5395556756780948e-05

2.6086915615647355e-05

גם כאן הזמן גודל פי מס' גדול יותר מהקודם

אבל פה זה האלגוריתם יותר יעיל כי הזמנים יותר קטנים מ2 האלגוריתמים האחרים.

ג. זמן הריצה ואז מס האפסים בסדר של השיטות שמופיע בתרגיל.

המס' 10^{10} , 10^{100} , 10^{1020} , 10^{10300}

0.23935685916957222

10000

0.1713238938613211

10100

0.17455268423670328

10200

0.17847769590097584

10300
0.0012987862741768996
10000
0.0013253008441469483
10100
0.001362506772975891
10200
0.0013796129471500551
10300
0.0021126125105148663
10000
0.0022486065951998757
10100
0.0021143231279323604
10200
0.00266001008408967
10300

המסקנה היא שיש הבדל מאוד זניח בזמן ריצה בין כל אחד מהמס' באלגוריתם כלשהו.
אבל במקרה הזה לעומת סעיף א זמן הריצה גדול בצורה משמעותית בגלל שמס' האפסים בקלטים
גדול משמעותית.

ד.ראינו ששם לאותו מס' זמן הריצה הוא 2^{10-5} בערך.

ההבדל פה שבלולאה הזאת היא עובדת כמס' הפעמים שהמספר שווה! ו 2^{**200} שווה המון!
בלולאת פור הקודמת היא בכלל עובדת כמס' הספרות שיש במספר בערך 20-30 וסופרת אפסים.
אז בלולאה החדשה התהליך יכול לקחת המון זמן.. החישוב הוא- $(2^{200/20}) * (0.00005)$

$4 * 10^{54}!!$

.6

The maximal length is 16

Sequence starts at 34803

Sequence is 1775199331957997