

Do omówienia po sprawdzianie:

- kiedy lepszy cpp a kiedy py - w kontekście zadań maturalnych,
- kiedy lepsze listy, kiedy tabele - w kontekście algorytmicznym,
- jak samej/samemu zdecydować kiedy i w jaki sposób opakować kod w funkcję,
- jak regulować sobie poziom trudności zadania - logiczna dekompozycja,
- jak testować kolejne porcje kodu,
- odróżniać istotne elementy zadania od pobocznych - w kontekście matury (uniwersalne & parafialne)
- dług technologiczny w życiu i na maturze (kiedy pisać „na szybko” a kiedy „na elegancko”?)
- kiedy używać `for ...` a kiedy `while ...`
`if ...`
`break`
- tworzenie własnych modułów (temat na kiedyś tam) `import my-module`
- czy optymalizacja kodu jest zawsze dobra?
- printowanie tabel może zamulić program

• zmiany w miejscu (in place) czy w nowej

strukturze

np. `zeros_like(tab)`

czy
np. `zeros(tab.shape)`

my_module.py

```
class Utils:
```

```
    def rysuj_kola...
```

```
class Config:
```

```
    MAX_TEMP=32
```


numba jest
tylko do zabawy
w wolnej chwili.
W kroku IV wystarczy
obstuzyc uciekinierów
cwaniaków.

Tips & tricks

- jeżeli wprowadzam własne koncepcje do rozwiązania, warto dać w kodzie stosowny komentarz /uzasadnienie (zwięzły)
- robic' czasem sejiwa (ostatnia działająca wersja)
- Na maturze będzie czasu za mało
- jeżeli boisz się, że sprawdzający czegoś nie skuma, należy wykazać się empatią i wstawić komentarz w kodzie
- CTRL+C CTRL+V to Twoi sprzymierzeńcy
CTRL+ /
- niezbudowane warunki logiczne to podszept szatana

in place w miejscu

siatka = np.array(-----, dtype=...)

mapa = np.zeros_like(siatka)

$\neg$
 $\vee \wedge \exists$

$\neg p \wedge q \Rightarrow$

if $a == 4$ and $b = 2$ or $a + b \geq 8$

$2 < a < 4$

$i = 0$
warunek = true

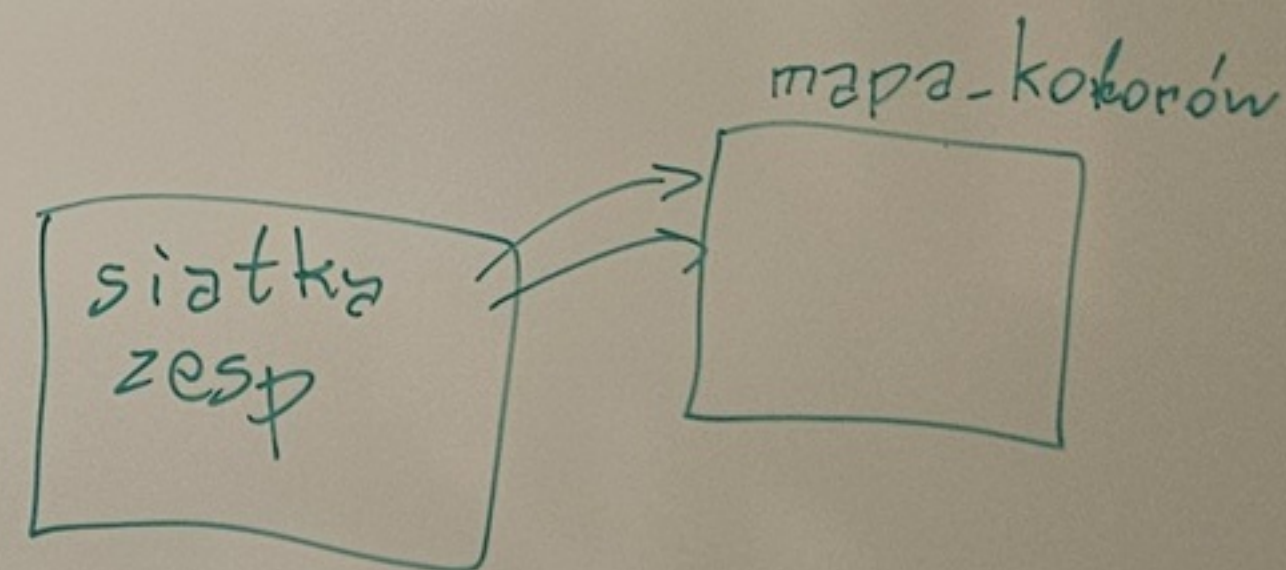
for elem in cos; while warunek:

.....
if „awaria”
break;

.....
 $i += 1$

if - - -

if - - -



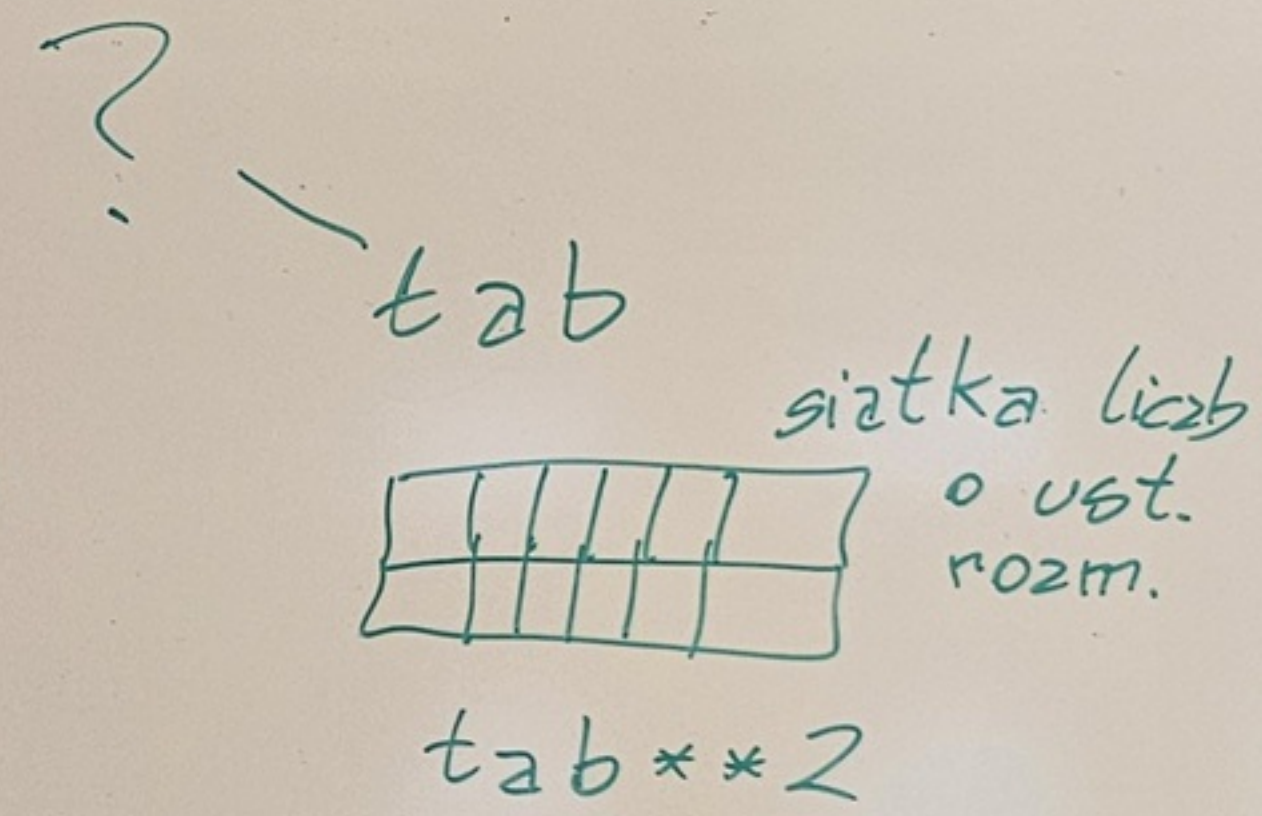
lista
append()

lista**2

[l**2 for l in lista]

lista1 + lista2

MANIAKALNIE
krok po kroku



2
cpp Python
wskaźniki
wektory

my-module.py
MAX_ITER = 48
NIĘSK = 1e10

class Config:

import my-module as i

i.MAX_ITER

print(tab) !

tab[:10, :10]

[-4:, -4:]