

Vsak program lahko naleti na napako. Način, kako se y njimi spopadamo v pythonu je sledeč. Del kode, za katerega sumimo, da se lahko v njem sгоди napaka postavimo v try: Npr da želimo deliti število 10 z nekaj števili

```
stevila = [1, 3, 2, 0, 8, 4]

for stevilo in stevila:
    print(10 / stevilo)
```

Kar lahko naredimo torej je da del kode, ki nam dela probleme postavimo v try blok. tega združimo skupaj z except

```
stevila = [1, 3, 2, 0, 8, 4]

for stevilo in stevila:
    try:
        print(10 / stevilo)
    except:
        print("napaka")
```

To pa nam ne pove prav dosti kaksne vrste napako smo ujeli. zato uproabimo specificne napake ali pa kar generalno exceptions

```
stevila = [1, 3, 2, 0, 8, "dan"]

for stevilo in stevila:
    try:
        print(10 / stevilo)
    except Exception as e:
        print(type(e))
        print(e)
```

Odkrili smo, da sta pri nasem deljenju dve napaki. Sedaj lahko z njima rokujemo kot zelimo

```
stevila = [1, 3, 2, 0, 8, "dan"]

for stevilo in stevila:
    try:
        print(10 / stevilo)
    except ZeroDivisionError:
        print("deljenje z nic pa ne gre!")
    except TypeError:
        print("deliti moras nujno s stevilom")
```

dodajamo lahko tudi else in finally stavka, else se izvede vsakic, ce pred njem ni bila zaznana nobena napaka, finally pa se izvede cisto vsakic

```

stevila = [1, 3, 2, 0, 8, "dan"]
for stevilo in stevila:
    try:
        print(10 / stevilo)
    except ZeroDivisionError:
        print("deljenje z nic pa ne gre!")
    except TypeError:
        print("deliti moras nujno s stevilko")
    else:
        print("uspesno deljenje")
    finally:
        print(f"zelel si delit {10} z {stevilo} ")

```

ce pa imamo neke specificne zelje, kot na primer, da ne zelimo deliti s sodimi stevili, lahko napisemo svoj exception

```

stevila = [1, 3, 2, 5, 8, 7]
for stevilo in stevila:
    try:
        if stevilo % 2 == 0:
            raise ValueError("stevilo je sodo")
        print(10 / stevilo)
    except ValueError as e:
        print(e)

```

(<https://docs.python.org/3/library/exceptions.html>)

(Se pokaže kako ma Python zgrajeno hierarhijo dedovanja Errorjev). Se pravi, če damo kot prvi `except except Exception` bomo z njim prestregl vse, ker vsi dedujejo iz tega classa.

BaseException

- SystemExit
- KeyboardInterrupt
- GeneratorExit
- Exception
- - StopIteration
- - StopAsyncIteration
- - ArithmeticError
- - - FloatingPointError

- - - OverflowError
- - - ZeroDivisionError
- - AssertionError
- - AttributeError
- - BufferError
- - EOFError
- - ImportError
- - - ModuleNotFoundError
- - LookupError
- - - IndexError
- - - KeyError
- - MemoryError
- - NameError
- - - UnboundLocalError
- - OSError
- - - BlockingIOError
- - - ChildProcessError
-

- - ConnectionError
- - - - BrokenPipeError
- - - - ConnectionAbortedError
- - - - ConnectionRefusedError
- - - - ConnectionResetError
- - - FileExistsError
- - - FileNotFoundError
- - - InterruptedError
- - - IsADirectoryError
- - - NotADirectoryError
- - - PermissionError
- - - ProcessLookupError
- - - TimeoutError

- - ReferenceError
- - RuntimeError
- - - NotImplementedError
- - - RecursionError
- - SyntaxError
- - - IndentationError
- - - - TabError
- - SystemError
- - TypeError
- - ValueError
- - - UnicodeError
- - - - UnicodeDecodeError
- - - - UnicodeEncodeError
- - - - UnicodeTranslateError
- - Warning
- -

- - - DeprecationWarning
- - - PendingDeprecationWarning
- - - RuntimeWarning
- - - SyntaxWarning
- - - UserWarning
- - - FutureWarning
- - - ImportError
- - - UnicodeWarning
- - - BytesWarning
- - - ResourceWarning

Naloga: Napišite funkcijo `is_anagram`, ki od uporabnika zahteva, da vnese dve besedi. Funkcija naj vrne `True`, če sta besedi anagrama, v nasprotnem primeru `False`. Anagram je beseda ali fraza, ki se oblikuje z urejanjem črk druge besede ali fraze, običajno z uporabo vseh izvirnih črk točno enkrat.

Če uporabnik vnese samo številke za katero koli od besed, naj funkcija rais-a `ValueError`.

Program naj 3x zažene funkcijo. V kolikor pride do `ValueError`, naj se izpiše sporočilo in izvajanje programa nadaljuje.

```
beseda = input("vnesei")
```

```
Vnesi prvo besedo: listen
Vnesi drugo besedo: silent
Besedi sta ANAGRAMA
```

```
Vnesi prvo besedo: hello
```

Vnesi drugo besedo: world
Besedi NISTA anagrama.

Vnesi prvo besedo: 1234
Vnesi drugo besedo: test
Vnešene so bile samo številke.

```
def is_anagram():  
    beseda1 = input("Vnesi prvo besedo: ")  
    beseda2 = input("Vnesi drugo besedo: ")  
    if beseda1.isnumeric() or beseda2.isnumeric():  
        raise ValueError("Vnešene so bile samo številke.")  
  
    return sorted(beseda1) == sorted(beseda2)  
  
for _ in range(3):  
    try:  
        if is_anagram():  
            print("Besedi sta ANAGRAMA")  
        else:  
            print("Besedi NISTA anagrama.")  
    except ValueError as e:  
        print(e)  
    print()
```