

NASP - 1. laboratorijska vježba

U ovom se dokumentu nalaze zadaci koji će biti u 1.laboratorijskoj vježbi iz kolegija Napredni algoritmi i strukture podataka. Cilj je ove vježbe upoznati se s programskim jezikom Python kojeg ćete koristiti u narednim vježbama.

Preporučamo izvođenje vježbe prije samog termina laboratorijske vježbe unutar kojeg ćete imati priliku postavljati pitanja putem *ticketing* sustava.

1. Zadatak

Definirajte listu imena koja sadrži duplikate, primjerice:

```
names = ["Ana", "Petar", "Ana", "Lucija", "Vanja", "Pavao", "Lucija"]
```

Sortirajte listu prema imenima obrnuto abecednim redom u funkciji čija je deklaracija definirana na sljedeći način (naziv funkcije i broj ulaznih argumenata moraju bit isti):

```
def reverse_sort(names :list) -> list:  
    ...
```

Rezultat funkcije `reverse_sort` zapišite u varijablu `names_desc`.

U varijablu `selected_names` zapišite imena sortirana obrnutim abecednim redom od prvog (računajući od nule) do predzadnjeg elementa (uključivo). Koristite sintaksu za dohvaćanje podliste unutar liste.

U varijablu `unique_selected_names` zapišite jedinstvene vrijednosti koje se nalaze unutar liste `selected_names`. Kao strukturu podataka koristite `set`.

Pomoću *for* petlje iterirajte po setu `unique_selected_names`. Svakom jedinstvenom imenu konkatenerajte niz znakova "- pass" te rezultat zapišite u listu `pass_names`.

2. Zadatak

Definirajte riječnik `person_data` (python klasa *dict*) koji kao ključ sadrži imena osoba i godinu rođenja. Definirajte sljedeće podatke *Ana : 1995, Zoran : 1978, Lucija: 2001, Anja : 1997*.

U *for* petlji iterirajte riječnikom te svakom članu umanjite godinu rođenja za jedan. Tako će, primjerice, *Ana : 1995* postati *Ana : 1994*.

Definirajte listu `year_age`. Iterirajte po svim vrijednostima (godinama rođenja) u riječniku `person_data` te unutar petlje definirajte n-torku (python klasa *tuple*) koja sadrži (godinu rođenja, dob). Dob izračunajte kao *2022 - godina rođenja*. U svakoj iteraciji dodajte pripadajuću n-torku listi `year_age`.

3. Zadatak

Koristeći biblioteku *numpy* definirajte dva vektora na sljedeći način

```
import numpy as np

vector_a = np.array([[1], [3], [5]])
vector_b = np.array([[2], [4], [6]])
```

Napravite vanjski umnožak `vector_a` i `vector_b` te rezultat zapišite u varijablu `mat_mul`.

Skalarno pomnožite `vector_a` i `vector_b` te rezultat zapišite u varijablu `vect_dot`.

Potencirajte svaki član matrice `mat_mul` na drugu potenciju te rezultat zapišite u varijablu `mat_exp`.

U varijablu `sub_mat` zapišite podmatricu veličine 2x2 koja se nalazi u četvrtom kvadrantu matrice `mat_exp`.

Napomena: za umnoške možete koristiti operator `@` (umnožak matrica, gdje se vektore promatra kao matrice), a npr. transponirana verzija vektora `vector_a` se dobije sa `vector_a.T`.

Napomena: Za više o vanjskom umnošku vektora pogledajte na [poveznici](#).

4. Zadatak

Definirajte klasu *Person* koja sadrži attribute *name* i *age*. Pritom u klasi definirajte konstruktor koji prima vrijednosti *name* i *age*.

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

Unutar klase *Person* definirajte metodu *increase_age(self)* unutar koje ćete uvećati vrijednost godina za 1.

Definirajte varijable `first_person` i `second_person` kao instance klase *Person* s vrijednostima *Marko*, 39 i *Ivan*, 17. Vrijednosti predajte preko konstruktora.

Pozovite metodu `increase_age` nad instancom `second_person`.

Definirajte klasu *PersonDetail* koja nasljeđuje klasu *Person*. Pritom u klasi *PersonDetail* definirajte atribut *address* koji se uz nasljeđene attribute postavlja preko konstruktora.

Definirajte varijablu `first_person_detail` kao instancu klase *PersonDetail* s vrijednostima *Ana*, 25, *Unska* 3.

Nad varijablom `first_person_detail` pozovite metodu *increase_age*.