

VEŽBA 6

Pregled

Kao i mnogi drugi programski jezici i Python podržava razbijanje programa u sitnije delove. Podela najnižeg nivoa je podela na funkcije. Posebna vrsta funkcije koja ima sposobnost da generiše sekvencu (npr. listu) naziva se *generatorom*.

Definicija funkcije

Funkcija se u Pythonu definiše na sledeći način:

```
def ime_funkcije(lista_argumenata):  
    blok_tela_funkcije
```

Vraćanje vrednosti iz funkcije

Svaka funkcija može da vrati rezultat u vidu povratne vrednosti. Povratak iz funkcije i vraćanje vrednosti postiže se naredbom `return`. U Pythonu povratna vrednost može da bude i n-torka. U tom slučaju naredba `return` ima više parametara, a zagrade za obeležavanje n-torke nisu neophodne.

```
return x                # vraca jednu vrednost  
return a, b, c         # vraca 3 vrednosti (uredjenu trojku)
```

Zadaci

1. Napisati funkciju koja izračunava i ispisuje prvih n elemenata Fibonačijevog niza.
2. Napisati funkciju koja izračunava prvih n elemenata Fibonačijevog niza i vraća taj rezultat u obliku liste. Testirati funkciju.
3. Napisati program koji sortira elemente liste u neopadajući redosled. Sortiranje treba obaviti nalaženjem i uklanjanjem najmanjeg elementa nesortirane liste koji se zatim ubacuje u novu (na početku praznu listu). Rešenje izvesti kao funkciju koja sortira listu koja je prosleđena kao prvi i jedini parametar, a sortiranu vraća. Algoritam je dat na kraju teksta.
4. Napisati funkciju koja obavlja sortiranje liste bez korišćenja dodatne liste, promenama na listi koja je prvi argument funkcije. Algoritam treba da bude *selection sort*. Kriterijum sortiranja treba zadati funkcijom koja se prenosi kao drugi argument. Pomoć za razvoj algoritma dat je na kraju ovog teksta.

Dodaci

Funkcija koja implementira selection sort korišćenjem nove liste

Sledeća funkcija implementirana je u *Python-u*:

```
def sort(ls, kt):  
    """ Sortira listu zadatu kao prvi argument,
```

```

rezultat vraca kao novu sortiranu listu,
kriterijum sortiranja zadaje se kao funkcija"""
dl = len(ls)
xdl = dl
i = 0
nls = []
while i<dl:
    me = ls[0]
    xx = 0
    j = 1
    while j<xdl:
        if kt(ls[j], me):
            me = ls[j]
            xx = j
        j += 1
    xdl -= 1
    nls.append(ls.pop(xx))
    i += 1
return nls

```

Kriterijumska funkcija

Kao i prethodna, i ova funkcija je implementirana u *Python*-u.

```

def krit(a, b):
    """ Funkcija koja zadaje kriterijum neopadajuće liste"""
    return a<b

```

Selection sort algoritam implementiran zamenom mesta elementima

Sledeća funkcija data je u *pseudo-kodu*. On svojom formom najviše podseća na jezik C, za uslove i petlje čak su i korišćene ključne reči iz tog jezika.

```

// SELECTION-SORT ALGORITAM BEZ KREIRANJA NOVE LISTE
dl = dužina_list(lista)
i = 0
while ( i < (dl-1) )
{
    me = ls[i]
    xx = i
    k = i + 1
    while ( k < dl )
    {
        if ( ls[k] |*|1 me)
        {
            me = ls[k]
            xx = k
        }
    }
}

```

1 Znak |*| označava proizvoljnu operaciju poređenja na osnovu koje će se lista sortirati. Za sortiranje u neopadajućem redosledu ovu oznaku treba zameniti znakom <

```
    k = k + 1
}
if (i != xx)
{
    ls[xx] = ls[i]
    ls[i] = me
}
i = i + 1
}
```

Fibonačijev niz

Za definiciju Fibonačijevog niza uzeti:

$$F_i = F_{i-1} + F_{i-2}$$

$$F_0 = 0; F_1 = 1$$

Primer: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Količnik $\frac{F_i}{F_{i-1}}$ konvergira vrednosti tzv. „zlatnog preseka“.