

ISPIT IZ DIZAJNA PROGRAMSKIH JEZIKA – 19.1.2016

1. **(Python 50%)** Korisnik unosi putanju do direktorijuma. Nakon toga program rekurzivno prolazi kroz taj direktorijum i pronalazi sve html datoteke. Svaka html datoteka se otvara i iz nje se izvlače sve putanje linkova. Pod linkovima podrazumevati delove teksta tipa:

```
href="index.html"  
href="https://hg.python.org/cpython/file/2.7/Lib/netrc.py"  
href="..py-modindex.html"  
src="../_static/version_switch.js"
```

Dakle, sve delove teksta oblika href="xyz" ili src="xyz" gde je xyz relativna ili apsolutna putanja. Nakon izvlačenja svaku adresu je potrebno ubaciti u mapu ili ažurirati njen broj pojavljivanja u slučaju da već postoji u mapi. Pritom se adresa uvek ubacuje u formi apsolutne adrese što znači da je relativno zadate adrese potrebno rekonstruisati pre ubacivanja u zavisnosti od toga gde se datoteka koja poziva link nalazi.

Npr. ako se link „../_static/version_switch.js“ nalazi u datoteci sa putanjom:

```
„/home/DPJISPITDOK/python/library/netrc.html“  
onda je rekonstruisana apsolutna putanja pomenutog linka:  
„/home/DPJISPITDOK/python/_static/version_switch.js“
```

Na izlazu ispisati mapu u opadajućem redosledu prema broju pojavljivanja adrese. Testirati program pozivom za direktorijum u kojem se nalazi ispitna dokumentacija.

2. **(C++ 50%)** Sve šablon funkcije testirati nad listom pravougaonika i listom celih brojeva.
- A. **(15%)** Napraviti klasu Pravougaonik koji je opisan: nazivom (string), pozicijom gornjeg levog temena (x i y koordinata), dužinom i širinom. Sve brojevne informacije su tipa float. Napraviti podrazumevani konstruktor, zatim konstruktor koji prihvata sve potrebne informacije, i implementirati operatore >> i <<. Takođe je potrebno implementirati i binarni operator < koji vraća bool i to vrednost „tačno“ ukoliko je površina prvog pravougaonika manja od površine drugog tj. „netačno“ u suprotnom. Redefinisati i operator + koji funkcioniše tako što sabere sve brojevne vrednosti dva pravougaonika, dok se naziv formira tako što se nadovežu nazivi data dva pravougaonika.
- B. **(15%)** Napraviti šablon funkciju za učitavanje u listu proizvoljnog tipa u rastuće uređenom redosledu korišćenjem operatora <. Korisnik najpre unosi broj elemenata, a potom i svaki element redom. Za ubacivanje na odgovarajuću poziciju u listi koristiti funkciju **insert** koja prihvata poziciju iteratora na koju hoćemo da ubacimo kao prvi argument, a vrednost koju ubacujemo kao drugi.

```
template <typename T> void ucitajUredjeno(list<T> &lista);
```

- C. **(5%)** Napraviti funkciju koja ispisuje proizvoljnu listu.

```
template <typename T> void ispisi(list<T> lista);
```

- D. **(15%)** Napraviti šablon funkciju koja prihvata proizvoljnu listu i potom nad njom vrši transformaciju zadatu kao pokazivač na funkciju. Funkcija koja se prosleđuje je binarna i primenjuje se nad posmatranim elementom i njegovim sledbenikom. Transformacija se ne vrši nad poslednjim elementom liste.

```
template <typename T> void transformisi(list<T> lista, T (*f) (T,T));
```

Testirati na funkcijama:

```
T f1(T a, T b){ return a+b; }
```

```
T f2(T a, T b){ if(a<b) return a; else return b; }
```