

Programske paradigme - Oktobar2 2010

1. [20%] Napisati prolog predikat koji vrši eliminaciju svih sekvenci istih brojeva neparne dužine. Npr.

elimnep([1, 1, 1, 2, 2, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5])->[2, 2, 4, 4, 4, 4]

2. [20%] Napisati prolog predikat koji generiše slučajnu permutaciju nad listom npr.

permrnd([a, b, c, d, e])-> [a, c, d, b, e]

3. [25%] Implementirati quick-sort algoritam u prologu. (Quick-sort je baziran na dekompoziciji liste u odnosu na odabrani pivot element. Pivot birati na proizvoljan način, npr. prvi element liste, a zatim particionisati listu u odnosu na pivot. Zatim podnizove na koje taj pivot deli početnu listu rekurzivno sortirati). Pseudokod particionisanja:

```
Partition(A, p, r)
  x = A[p]                # the pivot element
  i = p - 1
  j = r + 1
  while(1){
    j = j - 1;
    while(A[j] <= x)
      j = j - 1;
    i = i + 1;
    while(A[i] >= x)
      i = i + 1;
    if(i < j)
      swap(A[i], A[j]);
    else
      return j;
  }
```

4. [35%] Implementirati u Javascript-u aplikaciju za predstavljanje rasporeda časova.

- a. Od kontrola tu su tekst polje u koje korisnik unosi naziv predmeta, dve padajuće liste (jedna sa radnim danima u nedelji), a druga sa parnim radnim satima (od 8 – 16). Tu je dugme kojim se potvrđuje unos predmeta u raspored. [10%]
- b. Nakon što je korisnik potvrdio unos, podaci se prikazuju na odgovarajućem mestu u tabeli 5 x 5, gde jedna dimenzija predstavlja dan, a druga dvočasovni vremenski slot (8-10, 10-12, 12-14, 14-16 ili 16-18). [10%]
- c. Podatke čuvati u kolačićima, tako da kada se korisnik vrati na stranicu prethodni rad na rasporedu ostane sačuvan. [15%]

(Napomena: da bi se priznali delovi a. i b. ne mora se implementirati nikakvo pohranjivanje podataka, tj. na svaki klik dugmeta dovoljno je prikazati samo poslednju stavku u rasporedu)

SREĆNO!