

Ispit iz Programiranja 1

Ispit traje 120 minuta

Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
 b) Vrednost odgovora: tačan = **5**; netačan = **-1.25**; nevažeći (nula ili više zacrnjenih kružića) = **0**.
 c) Na pitanjima se može osvojiti najviše **15** poena.
 d) Prvi zadatak nosi **25** poena, a drugi **30**.

I. ZADACI

1) Napisati program na programskom jeziku Pascal koji utvrđuje broj pojavljivanja slova engleskog alfabeta u tekstu koji se u proizvoljnom broju redova nalazi u tekstualnoj datoteci `clanak.txt`. Prilikom utvrđivanja broja pojavljivanja, zanemariti razliku između malih i velikih slova. Ostale znake zanemariti prilikom obrade. Rezultate obrade ispisati na standardnom izlazu, tako što se najpre ispiše samo slovo, a zatim i broj pojavljivanja, razdvojen razmakom, za svako slovo u posebnom redu. Utvrđivanje broja pojavljivanja slova i ispis rezultata realizovati kao zasebne potprograme koji sa glavnim programom komuniciraju isključivo putem argumenata i povratne vrednosti. Voditi računa o korektnoj upotrebi korišćenih resursa.

2) Napisati program na programskom jeziku Pascal koji sa standardnog ulaza učitava parove: pozitivan broj `bx` (u decimalnoj osnovi) i broj `os` ($1 < os < 17$) koji predstavlja novu osnovu u koju će biti konvertovan broj `bx` i smešta takve parove u jednostruko ulančanu listu. Čitanje se prekida ukoliko se za neki od parova unese nekorektna vrednost. Potrebno je za svaki član liste konvertovati broj `bx` u novu osnovu `os`, (pri čemu se za cifre veće od 9 ukoliko su potrebne, koriste redom slova A, B, C, D, E, F) i odrediti razliku između najveće i najmanje cifre tako konvertovanog broja. Nakon toga treba ispisati članove liste na standardnom izlazu, i to za svaki član u posebnom redu broj u osnovi `bx` i `os`, kao i decimalnu vrednost razlike između najveće i najmanje cifre broja predstavljenog u osnovi `os`. Smatrati da konvertovani broj neće imati više od 10 cifara. Voditi računa o korektnoj upotrebi korišćenih resursa.

II. PITANJA

1) Koji je red funkcije složenosti sledećeg segmenta koda?

```
i := 1;
while (i * i < n) do
begin
  k := i; j := 1;
  while (j < k) do
  begin
    j := j * 2; k := k div 2
  end;
  i := i + 1
end;
```

A) $n \cdot \log(\sqrt{n})$ (B) $\log(\lfloor \sqrt{n} \rfloor!)$ C) $n \cdot \log(n)$

2) Šta ispisuje sledeći program napisan na programskom jeziku Pascal?

<pre>program juli(output); var a, b, c : integer; function f(a : integer) : integer; var i, j : integer; begin i := a; while i > 0 do begin j := 2; while (j > 0) and (j * j < a) do begin if (i mod j = 0) then j := -1; j := j + 1 end; end; end;</pre>	<pre>if (j > 0) then begin f := i; i := -1 end; i := i - 1 end; end; begin a := 127; b := a - 1; while (b > 0) do begin c := f(b); b := a - c; a := b; write(output, c, ' ') end; end.</pre>
--	--

A) 117 7 3

(B) 113 13 1

C) 7 3 3 2

3) Koja od ponuđenih procedura na programskom jeziku Pascal ispravno ispisuje sadržaj nepravne jednostruko ulančane liste celih brojeva u obrnutom poretku? Smatrati da jedan element ulančane liste sadrži ceo broj `b` i pokazivač na sledeći element `sled`.

<pre>procedure pisi1 (prvi: pok); begin if prvi^.sled <> nil then pisi1(prvi^.sled); write(prvi^.broj, ' '); end;</pre>	<pre>procedure pisi2 (prvi: pok); begin if prvi <> nil then begin pisi2(prvi^.sled); write(prvi^.broj, ' '); end; end;</pre>	<pre>procedure pisi3 (prvi: pok); begin write(prvi^.broj); if prvi <> nil then begin pisi3(prvi^.sled, ' '); end; end;</pre>
---	--	--

(A) `pisi1`(B) `pisi2`C) `pisi3`

4) Koja od ponuđenih sekvenci odgovara sintaksnoj definiciji za `<start>` zadatoj u BNF notaciji?

```
<start> ::= AA<a> | <a>A
<a> ::= <b> | <a>BBA | <b><a><d>A | <a><d>A
<b> ::= BA | BB<b>BB | AA<b>BB
<c> ::= BA | AB | BA<c> | AB<c>
<d> ::= AB<c>
```

A) AABABBBAAABBA

(B) AABABBBABABABBA

C) AABABBBAAABBAABBA