

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 vrši obradu nad nizom celih brojeva. Program najpre učitava pozitivan broj M ($1 \leq M \leq 5$) koji određuje koliko najmanje cifara svaki broj u nizu mora da sadrži. Program zatim učitava dužinu niza N ($N \leq 100$), a nakon toga N nenegativnih celih brojeva sa standardnog ulaza. Ukoliko korisnik unese u niz negativan broj ili broj koji ima manje cifara od M , učitavanje celog niza treba ponoviti. Nakon učitavanja niza se vrši njegova obrada. Ukoliko je broj paran, treba ga zameniti najvećim brojem koji se dobije kada se iz originalnog broja izdvoji bilo kojih M uzastopnih cifara. Ukoliko je broj neparan, treba ga zameniti najmanjim brojem koji se dobije kada se iz originalnog broja izdvoji bilo kojih M uzastopnih cifara. Nakon obrade treba ispisati tako obrađeni niz. Učitavanje brojeva, obradu i ispis realizovati kao zasebne potprograme koji sa glavnim programom komuniciraju isključivo putem argumenata i povratne vrednosti.

Primer za $M = 3$ i $N = 4$

Ulazni niz: 343 205346 3428 205341

Obradjeni niz: 343 534 428 53

2) Napisati program na programskom jeziku Pascal koji pronalazi najpovoljniju ponudu za kupovinu đačke opreme. U svakom redu datoteke `ponude.txt` se nalaze podaci o ponuđačima i jediničnim cenama pet đačkih artikala po sledećem formatu: naziv ponuđača (jedna reč od najviše 30 karaktera), cena ranca (realan broj), cena sveske (realan broj), cena pernice (realan broj), cena olovke (realan broj), cena gumice (realan broj). Svi podaci su odvojeni po jednim blanko znakom. Program treba da učitava podatke o ponudi đačke opreme u jednostruko ulančanu listu. Nakon toga, korisnik treba da za svaki od pet artikala unese količinu koju je potrebno kupiti, a program da ispiše na standardnom izlazu nazive jednog ili više najpovoljnijih ponuđača i cenu za taj paket opreme. Voditi računa o korektnoj upotrebi korišćenih resursa.

II. PITANJA

1) Koji je red funkcije složenosti sledećeg segmenta koda?

```
k:=4;
for i:=1 to n do
begin
  j:=0;
  repeat
    k:=k+4; j:=j+8
  until k=j
end
```

A) $\log(2^n \cdot n)$ B) n^4 (C) 2^n

2) Šta ispisuje sledeći program napisan na programskom jeziku Pascal za unos 26 65?

```
program septembar17(input, output);
var a, b, c: integer;
function paran(a: integer): Boolean;
begin
  paran := ((a mod 2) = 0);
end;
function fun(a, b: integer): integer;
var g, d: integer;
begin
  d := 1;
  while paran(a) and paran(b) do begin
    a := a div 2; b := b div 2; d := d * 2;
  end;
```

```
while a <> b do
  if paran(a) then a := a div 2
  else if paran(b) then b := b div 2
  else if a > b then a := (a - b) div 2
  else b := (b - a) div 2;
g := a;
fun := g * d;
end;
begin
  readln(input, a, b);
  c := fun(a, b); writeln(output, c);
end.
```

A) 17

(B) 13

C) 19

3) Šta treba da stoji umesto ##### i &&&& da bi sledeća procedura na programskom jeziku Pascal ispravno uređivala jednostruko ulančanu listu celih brojeva po neopadajućem poretku? Smatrati da jedan element ulančane liste sadrži ceo broj `broj` i pokazivač na sledeći element `sled`.

```
procedure uredi (prvi: pok);
var tek1, tek2, tek: pok;
  t: integer;
begin
  tek1:= prvi;
  while tek1 <> nil do begin
    #####
```

```
while tek2 <> nil do begin
  if tek1^.broj > tek2^.broj then begin
    &&&&
  end;
  tek2:=tek2^.sled;
end;
tek1:= tek1^.sled;
end;
end;
```

(A) #####

tek2:= tek1^.sled;

&&&&:

t:= tek1^.broj;

tek1^.broj:= tek2^.broj;

tek2^.broj:= t;

B) #####:

tek2:= prvi;

&&&&:

t:= tek1^.broj;

tek1^.broj:= tek2^.broj;

tek2^.broj:= t;

C) #####:

tek2:= tek1^.sled;;

&&&&:

tek:= tek1;

tek1:= tek2;

tek2:= tek;

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

`<start> ::= <a>1 | <a>0``<a> ::= | 1<a>0 | <a><c>1`` ::= 00 | 10 | <c>1``<c> ::= <c>0 | 1`

A) 10100100010

(B) 110010111001

C) 001101010111