

**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 tekstualne datoteke `maps.txt` koja sadrži podatke o jednoj mapi u igrici Minesweeper. U prvom redu datoteke se nalaze dva cela broja M i N koji predstavljaju dimenzije mape ( $M \leq 100$ ,  $N \leq 100$ ). U narednih M redova se nalazi N nula ili jedinica koje predstavljaju prisustvo (1) odnosno odsustvo (0) mine na mapi, tako da je svaka vrsta prikazana u zasebnom redu. Program treba da pročita sadržaj navedene datoteke, a zatim da na osnovu učitane mape generiše novu mapu identične veličine čija će polja sadržati broj mina koje se nalaze u okružujućim poljima odgovarajućeg polja originalne mape. Novoformiranu mapu upisati u tekstualnu datoteku `bombs.txt` po istom formatu kao kod ulazne datoteke. Na kraju na standardnom izlazu ispisati najveći broj mina MAX koji okružuje neko polje na mapi ( $0 \leq \text{MAX} \leq 8$ ). Učitavanje mape, obradu i ispis rezultujuće mape u datoteku kao i određivanje najvećeg broja mina koje okružuju neko polje na mapi 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 određuje zbir dva polinoma predstavljena pomoću uređene jednostruko ulančane liste. Jedan element liste predstavlja jedan član polinoma i sadrži stepen (ceo broj) i nulti koeficijent (realan broj) uz odgovarajući član. Lista je uređena opadajuće po stepenu člana, a čuvaju se samo članovi sa nenultim koeficijentom. Program treba da učita dva polinoma sa standardnog ulaza, izvrši proveru da li su uređeni na traženi način i ukoliko jesu, odredi njihov zbir, takođe predstavljen u obliku jednostruko ulančane liste. Prilikom unosa, najpre uneti red polinoma, a zatim odgovarajuće članove. Rezultujući polinom ispisati na standardnom izlazu. Voditi računa o korektnoj upotrebi korišćenih resursa.

**II. PITANJA**

**1)** Koji je red funkcije složenosti sledećeg segmenta koda?

```
t := 2;
for i := 1 to n do
begin
  t := 2 * t;
  m := t; k := 1;
  while k < m do begin
    m := 2 * m;
    k := 4 * k
  end
end;
end;
```

A)  $\log(n)$ **(B)**  $n^2$ C)  $2^n$ 

**2)** Šta ispisuje sledeći program napisan na programskom jeziku Pascal, ako se redom unose vrednosti 4 3 8 8 5 3 6?

<pre>program plfeb(input, output); type niz = array[1..10] of integer; var a: niz;     s: set of 1..20;     i: integer; procedure proc(b: niz; x, n: integer); var i,j: integer; begin   i:=1; j:=1;   while i &lt;= n do begin     if b[i] &lt;&gt; x then begin       a[j]:=a[i];       j:=j+1;     end;     i:=i+1;   end end;</pre>	<pre>begin   s:=[];   for i:= 1 to 7 do     read(input, a[i]);   for i:= 1 to 7 do   begin     if not (a[i] in s) then s:= s + [a[i]]     else proc(a, a[i], 7)   end;   for i:= 1 to 7 do     write(output, a[i], ' '); end.</pre>
---	---

**(A)** 4 5 3 6 6 3 6

B) 4 3 8 8 5 3 6

C) 4 5 6 5 6 5 6

**3)** Šta ispisuje sledeći program na programskom jeziku Pascal ukoliko funkcija `ucitaj` ispravno formira jednostruko ulančanu listu koja redom sadrži brojeve 10 3 8 5 1 13 4? Smatrati da funkcije `f1` i `f2` primaju kao prve argumente pokazivače na početak ulančane liste. Takođe, procedura `brisi` ispravno oslobađa memorijski prostor koji lista zauzima, dok procedura `pisi` ispravno ispisuje sadržaj liste od početka.

<pre>program oolpl(input, output); type pokaz = ^elem; elem = record   broj: integer; sledeci: pokaz end; var lista: pokaz; n: integer; function f1 (lst: pokaz; var n: integer) : integer; begin   if (lst &lt;&gt; nil) then   begin     n := n + 1;     f1 := f1 (lst^.sledeci, n) +       lst^.broj   end   else f1 := 0 end;</pre>	<pre>function f2 (lst: pokaz; b: integer): pokaz; var p: pokaz; begin   if (lst &lt;&gt; nil) then begin     if (lst^.broj &gt; b) then       lst^.sledeci := f2(lst^.sledeci,b)     else begin       p := lst;       lst := f2 (lst^.sledeci, b);       dispose (p)     end   end;   f2 := lst end; begin   n := 0; ucitaj(lista);   lista := f2(lista, f1(lista,n) div n);   pisi(lista); brisi(lista) end.</pre>
---	---

**(A)** 10 8 13

B) 3 5 1 4

C) 10 8 4

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

```
<x> ::= 0<y>10|1<x>1|1<x>0
<y> ::= 0<y>1|10<z>
<z> ::= 11|00|<z>1|<z>0
```

A) 0101000011010

**(B)** 0010111101110**(C)** 1100101111010