



Državni izpitni center



M 0 5 2 7 8 1 1 3

JESENSKI ROK

# **RAČUNALNIŠTVO**

## **NAVODILA ZA OCENJEVANJE**

**Torek, 6. september 2005**

**SPLOŠNA MATURA**

**IZPITNA POLA 1****SKLOP ARHITEKTURA IN ORGANIZACIJA RAČUNALNIŠKIH SISTEMOV**  
**NALOGE IZBIRNEGA TIPA**

Obkrožite pravilno trditev.

1. Med osnovne naloge lupine (shell) operacijskega sistema sodi:

(2 točki)

- A razvrščanje procesov,
- B zaščita datotek,
- C upravljanje pomnilnika,
- D postavitve uporabniškega vmesnika.
- E zagotavljanje dostopa do perifernih naprav.

2. Stalno prisotni program (npr. BIOS) je shranjen v pomnilniku, ki ga imenujemo:

(2 točki)

- A SDRAM,
- B SRAM,
- C DRAM,
- D ROM, EPROM.
- E nič od naštetega.

3. V zbirnem programskem jeziku imamo ukaz `lds`, ki v skladovni kazalec (SP) naloži poljubno vsebino. Ukaz `ldaa` naloži vsebino v register A, ukaz `inca` pa poveča vsebino registra A za 1. Kolikšna je vrednost skladovnega kazalca po izvedbi naslednjega zaporedja ukazov? Sklad se širi od višjih naslovov proti nižjim.

```
lds #00ff
ldaa #5
inca
```

(2 točki)

- A A009
- B 00FF
- C 00FD
- D A00D
- E 1000

**4. Časovni multipleks pomeni, da:***(2 točki)*

- A isto fizično prenosno pot uporabimo za prenos več signalov tako, da vsakemu signalu izmenoma pripada določena časovna rezina;
- B več signalov združimo tako, da se prenašajo istočasno;
- C se signali prenašajo z različno hitrostjo;
- D vsak signal odpošljemo takrat, ko ga sprejemna stran potrebuje;
- E vsakemu signalu pripada svoj frekvenčni pas.

**5. Pomnilna hierarhija pomeni razvrstitev pomnilnikov po hitrosti dostopa do podatkov. Obkrožite pravilno razporeditev od najhitrejših do najpočasnejših:***(2 točki)*

- A predpomnilnik, registri, disk;
- B registri, predpomnilnik, primarni pomnilnik, disk;
- C disk, primarni pomnilnik, predpomnilnik;
- D predpomnilnik, primarni pomnilnik, registri, disk;
- E disk, predpomnilnik, primarni pomnilnik.

**6. Sestavni del večopravnega operacijskega sistema je obvezno:***(2 točki)*

- A prevajalnik za več programskih jezikov;
- B programska oprema za časovnik (timer);
- C programska oprema za sinhronizacijo opravil;
- D razvrščevalnik opravil;
- E programska oprema za ravnanje s prekinitvami.

**7. Do smrtnega objema (dead lock) pri večopravnem operacijskem sistemu lahko pride:***(2 točki)*

- A kadar obstaja mehanizem, ki lahko ustavi proces (predkupna večopravnost – preemptive multitasking);
- B kadar je večopravnost odvisna od hitrosti procesorja;
- C kadar so vsi procesi enakovredni (cooperative multitasking);
- D kadar razvrščevalnik opravil ne opravlja svojega dela;
- E do smrtnega objema pri večopravnem operacijskem sistemu sploh ne pride.

8. Zapis s plavajočo vejico se pogosteje uporablja kakor zapis s fiksno vejico, ker: (Obkrožite pravilne odgovore.)

(Vsi pravilni odgovori 2 točki, en pravilen 1 točka, že en nepravilen 0 točk)

- A zapis s fiksno vejico omogoča le zapis celih števil;
- B zapis s fiksno vejico zasede več prostora v pomnilniku;
- C zapis s plavajočo vejico omogoča zapis od zelo majhnih do zelo velikih števil;
- D je napaka pri zaokroževanju manjša, če uporabljamo zapis s plavajočo vejico;
- E je zapis s plavajočo vejico prilagojen desetišemu številskemu sestavu.

### SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE NALOGE IZBIRNEGA TIPA

Obkrožite pravilno trditev.

9. V programskem jeziku imamo deklarirano enodimenzionalno tabelo celih števil.

Pascal:

```
var tab:array[1..100] of integer;
```

C:

```
int tab[100];
```

Kaj izpiše naslednje zaporedje stavkov?

(2 točki)

PASCAL:

```
for i:=1 to 100 do
  if (odd(tab[i]))then
    write(tab[i]);
```

C:

```
for (int i=0;i<100;i++){
  if (tab[i]%2!=0){
    fprintf("%d",tab[i]);}}
```

- A Izpiše vse elemente tabele.
- B Izpiše lihe indekse tabele.
- C Izpiše sode indekse tabele.
- D Izpiše lihe elemente tabele.
- E Izpiše sode elemente tabele.

10. Katera od naslednjih trditev velja za zapise (Pascal) oziroma strukture (C)?

(2 točki)

- A Dostop do komponente zapisa/strukture je mogoč prek indeksa komponente.
- B Komponenta zapisa/strukture je lahko poljubnega enostavnega ali sestavljenega tipa.
- C Komponenta zapisa/strukture je lahko le enostavnega podatkovnega tipa.
- D Vse komponente zapisa/strukture morajo biti enakega tipa.
- E Komponente zapisa/strukture so lahko vseh enostavnih in sestavljenih tipov, ne morejo pa biti spet zapisi/strukture.

**11. Kakšni sta vrednosti spremenljivk  $x$  in  $y$  po izvršitvi naslednjega zaporedja ukazov?***(2 točki)***Pascal:**

```
x:=5; y:=8;
repeat
    x:= x - (y mod x);
    y:= y + 1;
until (x <= 1);
```

**C:**

```
int x=5; int y=8;
do {
    x= x - (y % x);
    y= y + 1;}
while (x>1);
```

- A x=1, y=10
- B x=1, y=1
- C x=-3, y=9
- D x=4, y=10
- E x=1, y=1

**12. Dan je podprogram SpremeniX:****Pascal:**

```
var x:integer;
procedure SpremeniX(var x:integer);
var i:integer;
begin
    for i:=1 to 5 do x:=x+(2*i)
end;
begin
x:=5; SpremeniX(x);
end.
```

**C:**

```
void SpremeniX(int *x){
int i;
for(i=1;i<=5;i++)
    (*x)+=(2*i);
}
int main(){
int x=5; SpremeniX(&x);
}
```

**Kakšno vrednost ima v glavnem programu definirana celoštevilska spremenljivka  $x$  po izvršitvi gornjih stavkov:**

*(2 točki)*

- A 35
- B 115
- C 5
- D nedefinirano
- E 7

**13. Kakšne vrednosti imata spremenljivki a in b ob koncu izvajanja programa?***(2 točki)***Pascal:**

```
var a, b:integer;
procedure prvi(a,b:integer);
begin
    b:=b-3;
    a:=b+a-2;
end;

procedure drugi(a,b:integer);
begin
    a:=a+1;
    dec(b,3);
    prvi(a,b)
end;

begin
    a:=3;b:=2;
    prvi(3,b);
    drugi(a,b)
end.
```

**C:**

```
int a,b;
void prvi(int a, int b){
    b=b-3;
    a=b+a-2;
}

void drugi(int a, int b){
    a+=1;b-=3;
    prvi(a,b);
}

void main(){
    a=3; b=2;
    prvi(3,b);
    drugi(a,b);
}
```

	<b>a</b>	<b>b</b>
A	<u>3</u>	<u>2</u>
B	3	-1
C	3	4
D	2	2
E	2	-1

**14. Programska jezika Pascal in C sodita v skupino:***(2 točki)*

- A strojnih jezikov,
- B zbirnih jezikov,
- C višjih programskih jezikov.
- D povpraševalnih/poizvedbenih (query) jezikov,
- E naravnih jezikov.

15. Orodje, ki ga uporabljamo pri odpravljanju semantičnih napak v programu, imenujemo:

(2 točki)

- A povezovalnik (linker),
- B razhroščevalnik (debugger).
- C nalagalnik (loader),
- D prevajalnik (compiler),
- E orodje RAD (rapid application development tool),
- F orodje CASE (computer aided software engineering tool).

16. Dani sta deklaraciji zapisa in datoteke:

(2 točki)

**Pascal:**

```
Type x=record
    stev:integer;
end;
dat = file of x;

var
    podatek:x;
    f:dat;
    i:integer;
```

**C:**

```
struct x {
    int stev;
};
x podatek;
int i;
FILE *f;
```

**Kateri od navedenih ukazov za branje zapisa iz datoteke je napisan pravilno?**

**Pascal:**

- A read(f,x);
- B readln(f,i);
- C read(f,podatek);
- D readln(f,podatek);
- E nobeden izmed predhodno navedenih.

C:

- A fread(&x,sizeof(x),1,f);
- B fread(x,sizeof(x),1,f);
- C fread(&podatek,sizeof(podatek),1,f);
- D fread(podatek,sizeof(podatek),1,f);
- E vsi ukazi so napačni.

**SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE**  
**NALOGE S KRATKIMI ODGOVORI**

17. Naslednje zaporedje gnezdenih if stavkov prepisite tako, da uporabite stavek case v programskem jeziku Pascal oziroma stavek switch v programskem jeziku C.

(2 točki)

**Pascal:**

```
if (x=10) then
  writeln ('odl')
else begin
  if (x=9) or (x=8) then
    writeln('pd')
  else begin
    if (x=7) then
      writeln('db')
    else begin
      if (x=6) then
        writeln('zd')
      else
        writeln('nzd')
    end;
  end;
end;
```

**C:**

```
if (x==10) {
  printf("odl");}
else {
  if ((x==9)|| (x==8)) {
    printf("pd");}
  else {
    if (x==7) {
      printf("db");}
    else {
      if (x==6) {
        printf("zd");}
      else
        printf("nzd");
    }
  }
}
```



**Rešitev:**

**Pascal:**

```

case (x) of
  10 : writeln('odl');
  8,9 : writeln('pd');
  7 : writeln ('db');
  6 : writeln('zd')
  else writeln('nzd');
end;

```

**C:**

```

switch (x) {
  case 10 : printf("odl"); break;
  case 9: printf("pd"); break;
  case 8: printf("pd"); break;
  case 7 : printf("db"); break;
  case 6 : printf("zd"); break;
  default : printf("nzd");
}

```

18. Iz sledi spreminjanja tabelarične spremenljivke Tabela ugotovite, kateri algoritem je bil uporabljen pri urejanju:

	tabela	korak
začetna vrednost	[2, 8, 9, 3, 1, 6]	
	[2, 8, 9, 3, 1, 6]	1
	[2, 8, 9, 3, 1, 6]	2
	[2, 3, 8, 9, 1, 6]	3
	[1, 2, 3, 8, 9, 6]	4
končna vrednost	[1, 2, 3, 6, 8, 9]	5

(2 točki)

**Rešitev:**

sortiranje z vstavljanjem

19. Napišite glavo podprograma v programskem jeziku Pascal oziroma prototip funkcije v programskem jeziku C, ki kot argumenta dobi celi števili  $a$  in  $b$  ter vrne količnik dobljenih števil  $a/b$  (realno število).

(2 točki)

**Rešitev:**

**Pascal:**

```
function kolicnik (a,b : integer):real;
```

**C:**

```
float kolicnik (int a, int b);
```

**20. Ugotovite izpis naslednjega programa:***(2 točki)**(za 3 vrstice 1 točka,**celoten izpis brez skoka v novo vrstico tudi 1 točka)***Pascal:**

```
var c1,c2 : char;
begin
  for c1:='A' to 'E' do begin
    for c2:='F' downto succ(c1) do
      write(c2);
      writeln;
    end;
  end.
end.
```

**C:**

```
#include <stdio.h>
int main() {
  for (char c1='A'; c1<'F'; c1++)
  {
    for (char c2='F'; c2>c1; c2--)
    {
      printf("%c", c2);
    }
    printf("\n");
  }
  return 0;
}
```

**Rešitev:**

```
FEDCB
FEDC
FED
FE
F
```

**SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE**  
**STRUKTURIRANE NALOGE**

**21. Deklarirajte zapis z naslednjimi podatki: ime in priimek (niza 20 znakov) in točke (celo število). Deklarirajte tabelo 30 zapisov.**

**Napišite podprogram, ki uredi zapise v tabeli po naraščajočih vrednostih točk.**

*Deklaracija podatkovnega tipa 1 točka  
Podprogram za sortiranje 3 točke*

**Rešitev:**

**Pascal:**

```
type zapis = record
    ime : string[20];
    priimek : string[20];
    rezul : integer;
end;
tabela = array [1..30] of zapis;
procedure sort (var t:tabela);
var i,j : integer;
    pom : zapis;
begin
    for i:=1 to 29 do
        for j:= 30 downto i+1 do
            if (t[j-1].rezul>t[j].rezul) then
                begin
                    pom:=t[j-1];
                    t[j-1]:=t[j];
                    t[j]:=pom;
                end;
        end;
    end;
end;
```

**C:**

```
struct zapis {
    char ime[21];
    char priimek[21];
    int rezul;
};
zapis tabela[30];
void sort(tabela t)
{
    int i,j;
    zapis pom;
    for (i=0;i<29;i++)
        for (j=29;j>i;j--) {
            if (t[j-1].rezul>t[j].rezul) {
                pom=t[j-1];
                t[j-1]=t[j];
                t[j]=pom;
            }
        }
}
```

22. V programskem jeziku Pascal ali C napišite program, ki kopira tekstovno datoteko v drugo. Pri tem upoštevajte možnost, da prva datoteka ne obstaja. V tem primeru naj program izpiše opozorilo.

(4 točke)

**Rešitev:**

**Pascal:**

```
Var t,f:file of text;
C:char;
Begin
  Assign(t,'datoteka1.txt');
  Assign(f,'datoteka2.txt');
  {$I-}
  reset(t);
  if IOResult<>0 then writeln('datoteka ne obstaja')
  else
  {$I+}
  rewrite(f);
  while (not eof(t)) do begin
    read(t,c);write(f,c)
  end;
  close(t);
  close(f);
end.
```

**C:**

```
#include <stdio.h>
#include <stdlib.h>
void napaka (char *s1, char *s2=""){
  fprintf(stderr, "%s %s\n", s1, s2);
  exit(2);
}
int main()
{
  FILE *vhod,*izhod;
  char c;
  if ((vhod=fopen("dat1.txt","rt"))==NULL) {
    napaka("Ni vhodne datoteke", "dat1.txt");
  }
  izhod=fopen("dat2.txt","wt");
  fscanf(vhod,"%c",&c);
  while (!feof(vhod)) {
    fprintf(izhod,"%c",c);
    fscanf(vhod,"%c",&c);
  }
  fclose(vhod);
  fclose(izhod);
  return 1;
}
```

23. Dano imamo naslednje zaporedje: 1, 5, 6, 11, 17, 28, 45, ...

Napišite podprogram, ki vrne n-ti element zaporedja. Argument podprograma je n.

(4 točke)

**Rešitev:**

**Pascal:**

```
function vrni(n:integer):integer;
begin
    a:=1;
    b:=5;
    case n of
        1: vrni:=a;
        2:vrni:= b;
        else begin
            for i:=3 to n do
                begin
                    x:=a+b;
                    a:=b;
                    b:=x;
                end ;
            vrni:=x;
        end;
    end;
end;
```

**C:**

```
int vrni(int n)
{int a=1;
 int b=5;
 int x;
 switch (n) {
 case 1 : return a;
 case 2 : return b;
 default : { for (int i=3; i<=n; i++){
                x=a+b;
                a=b;
                b=x;
            }
            return x;}
 }
};
```

24. Deklarirajte tabelo stotih pozitivnih celih števil. Napišite podprogram, ki izpiše tiste elemente tabele, katerih vrednost se razlikuje od povprečja tabele za več kakor 10 %.

(4 točke)

**Rešitev:**

**Pascal:**

```
type tabela = array [1..100] of word;
var t : tabela;
    p : real;
    i : integer;
function povprecje(t:tabela):real;
var vsota,i: word;
begin
    vsota:=0;
    for i:=1 to 100 do
        vsota:=vsota + t[i];
    povprecje:=vsota/100;
end;
begin
    p:=povprecje(t);
    for i:=1 to 100 do
        if (abs(p-t[i]) > (0.1 * p)) then
            writeln(t[i]);
    end.
end.
```

**C:**

```
#include <stdio.h>
#include <math.h>
float povprecje (unsigned int t[100])
{
    unsigned int i;
    float vsota;
    vsota=0;
    for (i=0;i<100;i++)
        vsota += t[i];
    return (vsota/100);
}
void main ()
{
    unsigned int t[100];
    float p;
    int i;
    p=povprecje(t);
    for (i=0;i<100;i++) {
        if (abs(p-t[i])>(0.1*p))
            printf("%d ",t[i]);
    }
}
```

## IZPITNA POLA 2

### SKLOP RAČUNALNIŠKE KOMUNIKACIJE NALOGE IZBIRNEGA TIPA

Obkrožite pravilno trditev.

**1. Za protokol IP na mrežni plasti protokolnega modela TCP/IP rečemo, da**

*(2 točki)*

- A se na prenosni plasti lahko povezuje le s protokolom TCP, z drugimi pa ne;
- B ga je mogoče vključiti tudi v druge protokolne sklade, saj je z vsemi združljiv;
- C je temeljni protokol internetnega omrežja, ki omogoča, da paket prispe na svoj ciljni naslov;
- D je eden od protokolov, ki ga v protokolnem modelu TCP/IP pogosto nadomestijo drugi protokoli;
- E omogoča fizično naslavljanje v omrežju.

**2. Kontrola pretoka je v omrežju potrebna zato, da:**

*(2 točki)*

- A sprejemnik lahko sproti sprejema pakete, ki prihajajo na njegov naslov;
- B se izgubljeni paketi lahko ponovno pošljejo;
- C paketi ne zaostajajo;
- D mogoč je prenos paketov v realnem času;
- E se lahko preveri pot vsakega paketa prek vseh mrežnih vozlišč.

**3. Žeton v omrežju omogoča, da:**

*(2 točki)*

- A paket potuje vedno v istem zaporedju med postajami;
- B vsaka postaja "ve", kdaj lahko odda paket, kdaj pa ga lahko le sprejme;
- C nekatere postaje pridejo na vrsto pogosteje kakor druge;
- D je delovanje omrežja lažje razumljivo;
- E omrežje ni nikoli preobremenjeno.

4. Usmerjevalni postopek je način določanja poti prometa od izvorne do ponorne točke v omrežju. Kaj je značilno za dinamično usmerjanje?

(2 točki)

- A Pri določanju poti upošteva vnaprej določene povezave.
- B Pri določanju poti vedno upošteva najkrajšo povezavo.
- C Pri določanju poti izbira naključne povezave.
- D Pri določanju poti upošteva trenutno obremenjenost povezav.
- E Pred določanjem poti izloči najbolj obremenjene povezave.

5. Petabyte je:

(2 točki)

- A  $2^{10}$  byte
- B  $2^{20}$  byte
- C  $2^{30}$  byte
- D  $2^{40}$  byte
- E  $2^{50}$  byte

6. Dvojiško število 11011,0111 je v osmiškem številske sestavu:

(2 točki)

- A 66,7
- B 33,34
- C 33,07
- D 63,31
- E 63,34

7. Pravica ustvarjanja in spreminjanja shem (opisov podatkov) podatkovne baze je namenjena:

(2 točki)

- A vsem uporabnikom,
- B programerjem,
- C upravitelju PB,
- D upravitelju PB in vodjem projektov,
- E upravitelju PB in programerjem.



**8. Programsko orodje, ki omogoča ustvarjanje, vzdrževanje podatkovne baze in dostop do podatkov, je:**

*(2 točki)*

- A orodje za računalniško podprto programsko inženirstvo (CASE);
- B sistem za upravljanje podatkovne baze (SUPB);
- C grafični uporabniški vmesnik;
- D uporabniška aplikacija;
- E operacijski sistem.

**9. S sekundarnimi indeksi dosežemo:**

*(2 točki)*

- A hitrejši dostop do podatkov;
- B manjši obseg zbirke;
- C optimalno izrabo diska;
- D krajši čas za dodajanje podatkov;
- E uporabo porazdeljenih podatkovnih baz.

**10. Vzdrževanje informacijskega sistema je posledica sprememb na področju:**

*(2 točki)*

- A samo poslovnih procesov;
- B samo strojne opreme;
- C samo programske opreme;
- D samo uporabniških zahtev za nove funkcionalnosti;
- E vseh navedenih sprememb.

**11. Zakaj uporabljamo orodja CASE?**

*(2 točki)*

*(za samo prvo navedbo 1 točka)*

**Odgovor:**

računalniško podprt razvoj programske opreme, izdelava repozitorija, dokumentacije.

**12. Navedite vsaj 4 težave pri razvoju programske opreme:**

(2 točki)  
(za dva odgovora 1 točka)

**Odgovor:**

predolgi razvojni cikli, nizka produktivnost osebja in skupin, nepredvidljiva kakovost  
končnega produkta, pomanjkanje osebja za razvoj programske opreme, visoki stroški razvoja,  
visoki stroški vzdrževanja

**13. Navedite vsaj 3 podatkovne modele**

(2 točki)  
(za dva odgovora 1 točka)

**Odgovor:**

hierarhični, mrežni, relacijski, objektno-relacijski, objektni

**14. Za atomarne (elementarne) procese diagramov toka podatkov izdelamo minispecifikacije, v katerih opišemo transformacijo vhodnih podatkov v izhodne. V ta namen lahko uporabimo tudi:**

(2 pravilna odgovora – 2 točki  
1 pravilen odgovor – 1 točka)

**Odgovor:**

odločitvene tabele  
odločitvena drevesa, psevdokodo, diagram poteka

**15. Podatkovno bazo sestavljajo tabele:**

**Proizvajalec (Sifra\_proizvajalca, Ime\_proizvajalca, Naslov\_proizvajalca, Kraj\_proizvajalca)**

**Proizvaja (Sifra\_proizvajalca, Sifra\_izdelka)**

**Izdelek (Sifra\_izdelka, Ime\_izdelka, Cena\_izdelka, Opis\_izdelka)**

**Napišite stavek SQL, ki vrne imena proizvajalcev iz Postojne.**

(1 točka)

**Rešitev:**

SELECT proizvajalec.ime\_proizvajalca

FROM proizvajalec

WHERE proizvajalec.kraj\_proizvajalca='Postojna' ;

Napišite stavek SQL, ki vrne imena izdelkov, dražjih od 30000, ki jih izdeluje bodisi proizvajalec LANCOM bodisi COMTRON.

(3 točke)

Rešitev:

SELECT izdelek.ime\_izdelka

FROM izdelek, proizvaja, proizvajalec

WHERE (izdelek.sifra\_izdelka=proizvaja.sifra\_izdelka) and

(proizvaja.sifra\_izdelka=proizvajalec.sifra\_izdelka) and

(izdelek.cena\_izdelka>30000) and ((proizvajalec.ime\_proizvajalca='LANCOM') or

(proizvajalec.ime\_proizvajalca='COMTRON'));

16. Narišite entitetno-relacijski diagram za beleženje tehničnih pregledov vozil. Za vsako vozilo beležimo: številko motorja, številko šasije, znamko vozila in registrsko oznako. Tehnični pregled se opravlja v pooblaščenem tehničnem servisu, ki ima davčno številko, ime, naslov in telefonsko številko. Servis ima lahko več telefonskih števil. Za tehnični pregled zapišemo le datum pregleda in status vozila na dan pregleda (ustrezno/neustrezno).

(entitete in razmerja 2 točki;  
entitete, razmerja, atributi, ključi 4 točke)

Rešitev:

