



Državni izpitni center



M 0 7 1 7 8 1 1 3

SPOMLADANSKI ROK

RAČUNALNIŠTVO

NAVODILA ZA OCENJEVANJE

Sobota, 9. junij 2007

SPLOŠNA MATURA

Popravljená moderirana različica

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

Obkrožite pravilno trditev.

1. V računalnikih imamo tudi pogone za zgoščenke (CD) in (DVD). Pogone skupaj z mediji prištevamo:

(2 točki)

- A samo med pomnilnike;
- B samo med vhodno/izhodne enote;
- C mesto v zgradbi računalnika ni določeno;
- D pogone za CD prištevamo samo med vhodno/izhodne enote, za DVD samo med pomnilnike;
- E med pomnilnike in tudi med vhodno/izhodne enote.

2. Ukaz je sestavljen iz operacijske kode in (obkrožite vse pravilne trditve):

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

- A operacije;
- B operandov;
- C naslovov operacije;
- D kode;
- E naslovov operandov.

3. Kolikšna je vrednost skladovnega kazalca (SP), ko se izvede vrstica 5? Sklad se širi proti nižjim pomnilniškim lokacijam.

(2 točki)

```
1    org    $A000
2    lds    #$00ff
3    ldaa   #5
4    pula
5    pula
```

- A 0100
- B 00FD
- C 00FF
- D 00FE
- E 0101

4. Kaj določa obseg naslovnega prostora pri posrednem naslavljanju?

(2 točki)

- A Dolžina akumulatorja.
- B Dolžina podatkovnega registra primarnega pomnilnika.
- C Organizacija primarnega pomnilnika.
- D Dolžina registrov v procesorju.
- E Dolžina programskega števca.

5. Za prenos podatkov med CPE, vhodno/izhodnimi napravami in pomnilnikom imamo različne načine. Eden izmed njih je neposredni dostop do pomnilnika (DMA – Direct Memory Access). Pri tem načinu:

(2 točki)

- A CPE izvaja program za V/I, vsi podatki gredo prek CPE. Enostavna izvedba, za manjše količine podatkov, CPE mora neprestano preverjati stanje vhodno/izhodne naprave.
- B CPE pošlje ukaz za prenos in nadaljuje delo. Ko je podatek pripravljen, pošlje naprava zahtevo za prekinitev. Imamo boljše izkoriščenost računalnika.
- C Podatki ne gredo skozi CPE, ampak po neposredni povezavi med V/I in glavnim pomnilnikom. Krmilnik DMA ima nadzor nad prenosi. Naslove in vrsto zahteve mu zagotovi CPE.
- D V/I procesorji izvajajo program za svojo V/I enoto in ponavadi še funkcijo krmilnika DMA.
- E Nič od naštetega.

6. V nekem trenutku je aktivnih več procesov. Kako lahko izpišemo procese, ki pripadajo nekemu uporabniku. To naredimo s pomočjo:

(2 točki)

- A whoami,
- B cat,
- C ps,
- D sed,
- E ps in grep.

Priznata se odgovora C in E.

7. Računalniški sistemi vsebujejo poleg procesorja in glavnega pomnilnika tudi množico perifernih enot (vhodno/izhodnih enot). Dele sistemske programske opreme – programe, ki omogočajo uporabniku računalnika dostop do teh perifernih naprav, imenujemo:

(2 točki)

- A vmesniki (interfaces);
- B služnostni programi (utilities);
- C razvrščevalniki (schedulers);
- D gonilniki (drivers);
- E ravnalniki (handlers).

8. Nepredznačeno binarno število 10011010011011 je v šestnajstiškem sistemu:

(2 točki)

- A 115154
- B 9A6C
- C 39532
- D 269B
- E Nič od naštetega.

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
NALOGE IZBIRNEGA TIPA

Obkrožite pravilno trditev.

9. Dana je deklaracija spremenljivke x.

Pascal:

```
var x : integer;
```

C:

```
int x;
```

Kateri od navedenih pogojev je resničen, če je vrednost spremenljivke x sodo število?

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

Pascal:

A `if ((x div 2) = 0)`

B `if ((x mod 2) = 0)`

C `if ((x / 2) = 0)`

D `if ((x and 1) = 0)`

E `if ((x - 2) = 0)`

C:

A `if ((x / 2) == 0)`

B `if ((x % 2) == 0)`

C `if ((x && 1) == 0)`

D `if ((x & 1) == 0)`

E `if ((x-2) == 0)`

10. Dan je izsek programa:

Pascal:

```
var x : real;
begin
x:=14.863;
write(x:8:1);
```

C:

```
float x;
x=14.863;
printf("%8.1f",x);
```

Posledica izvedbe stavka za izpis je:

(opomba: vsak izpisani znak je predstavljen v eni celici)

(2 točki)

A						1	4	.	8	6
B				1	4	.	8			
C				<u>1</u>	<u>4</u>	.	<u>9</u>	✓		
D						1	4	.	9	
E					1	4	.	8	6	3

11. Imamo naslednjo zanko for:**Pascal:**

```
for i:=7 to 9 do write(i);
```

C:

```
for (int i=7;i<=9;i++)
  { printf("%d",i);}
```

Zapisati jo želimo z enakovredno zanko while. Ta je:

(2 točki)

Pascal:

A while i<=9 do write(i);

B i:=7;
while i<=9 do begin
write(i); inc(i); end;

C i:=7;
while i<=9 do write(i);

D i:=7;
while i<9 do begin
write(i); inc(i); end;

E i:=7;
while i<=9 do begin
inc(i); write(i); end;

C:

A while (i<=9) do { printf("%d",i);}

B int i=7;
while (i<=9) {
printf("%d",i); i++; }

C int i=7;
while (i<=9) printf("%d",i);

D int =7;
while (i<9) {
printf("%d",i); i++;}

E int i=7;
while (i<=9) {i++; printf("%d",i);}

12. Med osnovne zahteve za algoritem sodi:

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

- A nedvoumnost,
- B optimalnost,
- C končnost,
- D prijaznost do uporabnika,
- E skromnost glede porabe računalniških resursov (pomnilnika in procesorja).

13. Programski jezik Pascal oziroma C uvrščamo med:

(2 točki)

- A programske jezike prve generacije,
- B programske jezike druge generacije,
- C programske jezike tretje generacije,
- D programske jezike četrte generacije,
- E poizvedovalne jezike.

14. Dana je naslednja deklaracija podatkovnega tipa:**Pascal:**

```
type zapis1 = record
  x: char;
  y: integer;
end;
type zapis2 = record
  tab: array [1..10] of zapis1;
  x : char;
end;
var z : zapis2;
```

C:

```
struct zapis1 {
  char x;
  int y;
};
struct zapis2 {
  zapis1 tab [10];
  char x;
};
zapis2 z;
```

Katera izmed naslednjih prireditev se izvede uspešno?

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

Pascal:

- A z.tab [1].x:='A';
- B z.x:='A';
- C z.zapis1 [1].x:='A';
- D z.zapis2 [1].x:='A';
- E z.x.tab [1]:='A';

C:

- A z.tab [1].x='A';
- B z.x='A';
- C z.zapis1 [1].x='A';
- D z.zapis2 [1].x='A';
- E z.x.tab [1]='A';

15. Kaj izpiše naslednji program?

(2 točki)

Pascal:

```
var n:integer;
function ugani(var x:integer) : integer;
var i:integer;
begin
  for i:=1 to 3 do inc(x);
  ugani:=x;
end;
begin
  n:=50;
  while (n<=60) do begin
    write (ugani(n):3);
  end;
end.
```

C:

```
#include <stdio.h>
int n;
int ugani (int *x)
{
  for (int i=1; i<4; i++)
    (*x)++;
  return *x;
}
void main()
{
  n=50;
  while (n<=60) {
    printf("%3d",ugani(&n));
  }
}
```

- A 53 56 59 62
- B 50 53 56 59
- C 50 51 52 53 54 55 56 57 58 59 60
- D 50 50 50 50 50 50 50 50 50 50 50 (neskončnokrat)
- E 53 53 53 53 53 53 53 53 53 53 53 (neskončnokrat)

16. Za algoritem Quicksort velja:

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

- A Pričakovani čas za urejanje tabel je velikostnega reda $n \cdot \log_2 n$.
- B V nekaterih primerih porabi za urejanje podatkov toliko časa kakor enostavne metode.
- C Temelji na porazdelitvi tabele na dva dela, v enem so le elementi, ki so večji od izbranega elementa, v drugem pa elementi, ki so manjši od izbranega elementa.
- D Primeren je samo za urejanje podatkov na sekundarnem pomnilniku.
- E Pričakovani čas za urejanje tabel je velikostnega reda n^2 .

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
NALOGE S KRATKIMI ODGOVORI

17. Dan je podprogram izracun():

Pascal:

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

C:

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

Kakšno vrednost ima celoštevilka spremenljivka x, ki je definirana v glavnem programu oziroma v glavni funkciji, po izvršitvi gornjih stavkov?

(2 točki)

Rešitev: *Priznata se odgovora 32 in 2.*

18. Napišite podprogram, ki sešteje vsa negativna števila v tabeli t in vrne njihovo vsoto. Pri pisanju podprograma uporabite naslednje deklaracije:

(2 točki)

Pascal:

```
Type tab = array [1..50] of integer;
Function vsota_negativnih(var t:tab):integer;
```

C:

```
int vsota_negativnih(int t [50]);
```

Rešitev:

Pascal:

```
Function vsota_negativnih(var t:tab):integer;
var i,v : integer;
begin
  v:=0;
  for i:=1 to 50 do
    if t [i]<0 then inc(v,t [i]);
  vsota_negativnih:=v;
end;
```

C:

```
int vsota_negativnih(int t [50])
{
  int i,v=0;
  for (i=0;i<50;i++)
    if (t [i]<0) v+=t [i];
  return v;
}
```

Delitev točk: Če je edina napaka bodisi, da je kandidat pozabil vrniti dobljeno vrednost ali pa narobe postavil pogoj v stavku if, dobi 1 točko.

19. Kaj izpiše dani program?

(2 točki)

Pascal:

```
Var s : string [10];
    i : integer;
    c : char;
Begin
  S:='matura';
  For i:=1 to length(s) div 2 do begin
    c:=s [i];
    s [i]:=s [length(s)+1-i];
    s [length(s)+1-i]:=c;
    writeln(s);
  End;
End.
```

C:

```
#include <string.h>
#include <stdio.h>
void main()
{
  char s [10]="matura";
  int i;
  char c;
  for (i=0;i<strlen(s)/2;i++) {
    c=s [i];
    s [i]=s [strlen(s)-i-1];
    s [strlen(s)-i-1]=c;
    printf("%s\n",s);
  }
}
```

Rešitev: aaturm

artuam

arutam

Delitev točk: Dve pravilno izpisani vrstici 1 točka, vse 3 pravilno izpisane vrstice 2 točki.

20. Deklarirana je enodimenzijska tabela t desetih znakov. Podatki so že vpisani. Napišite zaporedje stavkov, ki vsebino tabele premakne za k mest v desno. V izpraznjene elemente vpišite ' ' (presledek).

(2 točki)

Rešitev:

Pascal:

```
for i:=10 downto 1 do begin
  if (i>k) then
    t [i]:=t [i-k]
  else
    t [i]:=' ';
end;
```

C:

```
for (int i=9; i>=0; i--){
  if (i>=k)
    t [i]=t [i-k];
  else
    t [i]=' ';
}
```

Delitev točk: Pravilno premikanje znakov 1 točka, pravilno vpisovanje presledkov 1 točka.

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
STRUKTURIRANE NALOGE

21.

- a) Napišite podprogram, s katerim v novo tekstovno datoteko 'ZNAKI.TXT' vpišete 20 naključnih velikih črk. Med njimi naj bo en presledek. V eni vrstici naj bo pet črk. V podprogramu tudi zaprite datoteko.

Glava pascalskega podprograma: procedure ustvari;

Prototip C funkcije: void ustvari();

(2 točki)

- b) Napišite podprogram, ki izpiše, kolikokrat se v datoteki 'ZNAKI.TXT' pojavi črka, ki ima ASCII kodo n. Koda iskanega znaka naj bo argument podprograma.

Glava pascalskega podprograma: procedure isci(n:integer);

Prototip C funkcije: void isci(int n);

(2 točki)

Rešitev:

Pascal:

a)

```
procedure ustvari;
var f : text;
    i, j : integer;
    c : char;
begin
  assign(f, 'znaki.txt');
  rewrite(f);
  for i:=1 to 4 do begin
    for j:=1 to 5 do begin
      c:=chr(random(26)+65);
      write(f,c,' ');
    end;
    writeln(f);
  end;
  close(f);
end;
```

b)

```
procedure isci(n:integer);
var f : text;
    c : char;
    st : integer;
begin
  assign(f, 'znaki.txt');
  reset(f);
  st:=0;
  while not(eof(f)) do begin
    read(f,c);
    if (c=chr(n)) then
      inc(st);
  end;
  writeln(st);
  close(f);
end;
```

Rešitev:**C:****a)**

```
#include <stdio.h>
#include <stdlib.h>
void ustvari ()
{ FILE *f;
  f=fopen("znaki.txt","wt");
  for (int i=1;i<5;i++){
    for (int j=1;j<6;j++){
      char c=random(26)+65;
      fprintf(f,"%c ",c);
    }
    fprintf(f,"\n");
  }
  fclose(f);
}
```

b)

```
void isci(int n)
{
  FILE *f;
  char c;
  f=fopen("znaki.txt","rt");
  int st=0;
  while (fscanf(f,"%c",&c) && (!feof(f))) {
    if (c==n) st++;
  }
  printf("*****%d",st);
  fclose(f);
}
```

22. Deklarirana je dvodimenzijska tabela, ki ima pet vrstic in pet stolpcev realnih števil. V tabelo so vpisana le pozitivna realna števila.

a) Napišite podprogram, ki izpiše indeksa tistega diagonalnega elementa, ki ima največjo vrednost dela števila, ki je za decimalno piko. Pri reševanju naloge upoštevajte, da ima tabela dve diagonali.

(2 točki)

b) Napišite podprogram, ki preslika oziroma transponira vsebino tabele (zamenja vrednosti stolpcev in vrstic).

(2 točki)

Rešitev:

Pascal:

a)

```

type tabela = array [1..5,1..5] of real;
var t : tabela;
procedure naj_ind(var t : tabela);
var naj,tmp : real;
    i,j,naj_i,naj_j : integer;
begin naj:=0;
    for i:=1 to 5 do begin
        for j:=1 to 5 do begin
            if ((i=j) or (i+j=6)) then
                begin naj :=0;
                    tmp:=t [i,j]-trunc(t [i,j]);
                    if (tmp>naj) then
                        begin
                            naj:=tmp;
                            naj_i:=i;
                            naj_j:=j;
                        end;
                    end;
                end;
            end;
        writeln(naj_i, naj_j);
    end;
end;

```

b)

```

procedure transponiraj (var t: tabela);
var tmp:real;
    i,j : integer;
begin
    for i:=1 to 5 do
        for j:=i+1 to 5 do begin
            tmp:=t [i,j];
            t [i,j]:=t [j,i];
            t [j,i]:=tmp;
        end;
    end;
end;

```

Rešitev C:**a)**

```
#include <math.h>
#include <stdio.h>
void naj_ind(float t [5][5])
{
    float naj=0,tmp;
    int i,j,naj_i,naj_j;
    for (i=0;i<5;i++){
        for (j=0;j<5;j++){
            if ((i==j) || (i+j==4)) {
                tmp=t [i][j]-floor(t [i][j]);
                if (tmp>naj) {
                    naj=tmp;
                    naj_i=i;
                    naj_j=j;
                }
            }
        }
    }
    printf ("%d %d %6.2f",naj_i,naj_j,tmp);
}
```

b)

```
void transponiraj(float t [5][5])
{
    float tmp;
    int i,j;
    for (i=0;i<5;i++){
        for (j=i+1;j<5;j++){
            tmp=t [i][j];
            t [i][j]=t [j][i];
            t [j][i]=tmp;
        }
    }
}
```


23. Napišite podprogram (Pascal: Procedure lovec(x:integer; y:integer);) oziroma funkcijo (C: void lovec(int x, int y)), ki izpiše položaje, na katere lahko na prazni šahovnici prestavimo lovca, v naslednji obliki:

Pozicija poz_x,poz_y

Parametra x in y predstavljata zaporedno številko vrstice in stolpca, v katerem je lovec. Za tiste, ki ne poznate šaha, podajamo sliko:

- B - belo polje,
- C - črno polje,
- : - pozicija lovca,
- * - pozicija, kamor lahko prestavimo lovca.

```
BC*C*CBC
CBC:CBCB
BC*C*CBC
C*CBC*CB
*CBCBC*C
CBCBCBC*
BCBCBCBC
CBCBCBCB
```

(4 točke)

Rešitev:

Pascal:

```
procedure lovec(x,y: integer);
var i,j:integer;
begin
  for i:=1 to 8 do
    for j:=1 to 8 do
      if (x<>i) and (y<>j) then begin
        if ((i+j)=(x+y)) or
           ((x-i)=(y-j)) then
          writeln('pozicija ',i,',',j);
        end;
      end;
    end;
end;
```

Rešitev:

C:

```
#include <stdio.h>
void lovec(int x, int y)
{
  for (int i=1;i<=8;i++)
    for (int j=1;j<=8;j++) {
      if ((x!=i) && (y!=j)) {
        if (((i+j)==(x+y)) || ((x-i)==(y-j)))
          printf("pozicija %d,%d\n",i,j);
      }
    }
}
```

24. Napišite podprogram, ki vrne vsoto $1! + 2! + 3! + 4! + \dots + n!$, n je argument podprograma.

(4 točke)

Rešitev:

Pascal:

```
function vsota_n(n:integer):longint;
var vsota,f : longint;
    i : integer;
begin
    vsota:=0;
    f:=1;
    for i:=1 to n do begin
        f:=f*i;
        vsota:=vsota + f;
    end;
    vsota_n:=vsota;
end;
```

Rešitev:

C:

```
#include <stdio.h>
long int vsota_n(int n)
{
    long int vsota=0,f=1;
    for (int i=1;i<=n;i++) {
        f*=i;
        vsota+=f;
    }
    return vsota;
}
void main()
{
    printf("\n\n");
    for (int i=1;i<=10;i++)
        printf("%ld\n",vsota_n(i));
}
```

IZPITNA POLA 2

Obkrožite pravilno trditev.

1. Pri prenosu elektronske pošte se standardni aplikacijski protokol SMTP uporabi za:

(2 točki)

- A prenos od pošiljatelja do prejemnika pošte;
- B prenos od pošiljatelja do strežnika, ki prevzame pošiljateljevo pošto;
- C prenos od pošiljatelja do strežnika prejemnika pošte;
- D prenos od strežnika prejemnika pošte do prejemnika pošte;
- E prenos paketov na transportni plasti.

2. Omrežni protokol določa:

(2 točki)

- A »jezik« pravil in konvencij, po katerih komunicirajo omrežni elementi;
- B samo format sporočila;
- C kateri omrežni element začne komunikacijo;
- D samo dolžino sporočila;
- E aplikacije, ki lahko komunicirajo.

3. Maska omrežja (network mask) določa:

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

- A največje možno število računalnikov v omrežju,
- B ime podomrežja,
- C kateri del naslova IP se uporabi za omrežni naslov,
- D del omrežja, do katerega nimamo dostopa,
- E koliko računalnikov lahko uporablja isti naslov IP.

4. Protokol IP temelji na pošiljanju:

(2 točki)

- A sporočil,
- B točno določenega števila bajtov,
- C datagramov,
- D niza binarnih vrednosti, v katerega vrivamo binarne vrednosti za sinhronizacijo,
- E niza bajtov, ki ga začinja in zaključuje bajt s samimi 1.

5. Ključ, ki funkcionalno opredeljuje vse druge attribute relacije, se imenuje:

(2 točki)

- A sestavljeni ključ,
- B primarni ključ,
- C kandidat za ključ,
- D zunanji ključ,
- E sekundarni ključ.

6. Kateri od navedenih vrst informacijskih sistemov zahtevajo najvišjo stopnjo zanesljivosti delovanja?

(2 točki)

- A Vodstveni IS.
- B Ekspertni sistemi.
- C Odločitveni sistemi.
- D Managerski IS.
- E Transakcijski IS.

7. Logična datoteka je

(2 točki)

- A množica logičnih zapisov enakega tipa,
- B urejena množica povezanih zapisov,
- C množica podatkovnih elementov enakega tipa,
- D množica opisov zapisov,
- E množica opisov podatkovnih elementov.

8. Odločitvena tabela ima tri pogoje. Pogoj A ima štiri vrednosti, pogoj B ima tri vrednosti in pogoj C ima pet vrednosti. Število pravil odločitvene tabele je:

(2 točki)

- A 16
- B 12
- C 36
- D 60
- E 180

9. Kateri gradnik NI del modela entitete-razmerja (ER)?

(2 točki)

- A Entiteta.
- B Povezava.
- C Objekt.
- D Atribut.
- E Kardinalnost.

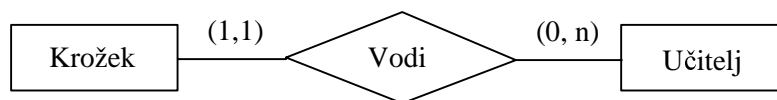
10. Kateri element diagrama toka podatkov predstavimo s simbolom ?

(2 točki)

- A Tok podatkov.
- B Terminator.
- C Proces.
- D Minispecifikacijo.
- E Zbirko podatkov.

11. Katere tabele bodo nastale iz naslednjega diagrama entiteta razmerja (ERD)?

(2 točki)



Rešitev: Krožek, Učitelj

12. Diagram toka podatkov, ki smo ga naredili v fazi analize, v fazi načrtovanja pretvorimo v

(2 točki)

Rešitev: strukturni diagram

13. Podatkovno bazo sestavljajo naslednje tabele:

Dobavitelj (DobaviteljID:N, ImeDob:A20, Naslov°:A50)

Izdelek (IzdelekID:N, ImeIzdelka:A10, Barva°:A10)

Katalog (DobaviteljID→Dobavitelj:N, IzdelekID→Izdelek:N, Cena:N)

Kaj vrne naslednji izraz relacijske algebre?

$$\pi[\text{Izdelek.ImeIzdelka}] \text{Izdelek} - \pi[\text{Izdelek.ImeIzdelka}] (\text{Izdelek} \mid x \mid \text{Katalog})$$

(2 točki)

Rešitev: Vrne imena izdelkov, ki niso v katalogu.

14. Naštete vsaj dve nalogi, ki jih opravlja sistem za upravljanje podatkovnih baz (SUPB).

(2 točki)

Zagotavlja integriteto podatkov, preverja dostopne pravice uporabnikov, izvaja poizvedbe.

Omogoča kreiranje in vzdrževanje podatkovne baze, omogoča izdelavo arhivskih kopij PB ...

15. Relacijsko podatkovno bazo (PB) sestavljata relaciji Delavec in Oddelek. Shemi relacij sta:

Delavec (DelavecID, Priimek, Ime, OddelekID→Oddelek, Poklic, Naslov, Kraj, Izobrazba, Datum_rojstva, Placa)

Oddelek (OddelekID, ImeOddelka, Naslov, Kraj)

a) Zapišite SQL stavek, ki vrne abecedno urejen seznam vseh delavcev s poklicem 'komercialist'.

(2 točki)

Rešitev:

```
SELECT priimek, ime FROM Delavec WHERE poklic='komercialist' ORDER BY priimek, ime;
```

b) Zapišite SQL stavek, ki vrne imena oddelkov iz Ljubljane, ki imajo več kakor deset zaposlenih računalniških tehnikov (računalniški tehnik je poklic).

(2 točki)

Rešitev:

```
SELECT imeoddelka
FROM Oddelek, Delavec
WHERE Oddelek.OddelekID=Delavec.OddelekID AND
Delavec.Poklic='računalniški tehnik'
GROUP BY imeoddelka
HAVING COUNT(Delavec.DelavecID) >10;
```

16. Študent mora med študijem opraviti več izpitov iz več predmetov. Vsak izpit dela pri točno določenem učitelju, lahko tudi večkrat. Za vsak izpit se beležijo StudentID, UciteljID, PredmetID, datum_izpita in ocena.

a) Narišite entitetnorelacijski (ER) diagram.

Če so v diagramu ER pravilno navedeni in povezani entitetni tipi in je napaka le v števnosti povezav, kandidat dobi 1 točko.

(2 točki)

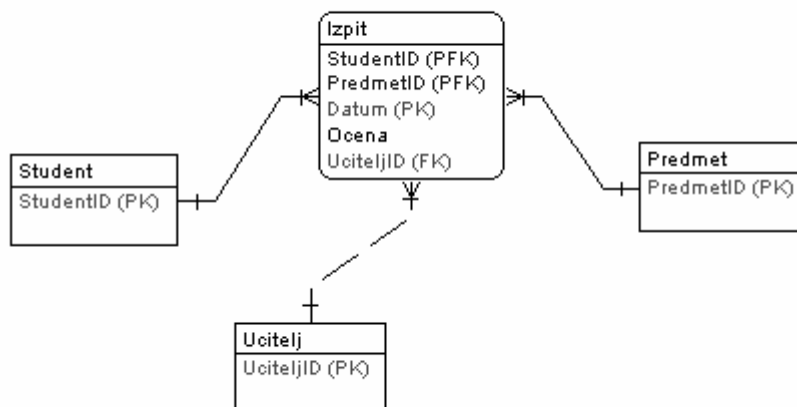
b) Na osnovi entitetnorelacijskega (ER) diagrama zapišite relacijske sheme relacij.

(2 točki)

Rešitev:

a)

[1.1]



b)

Student (StudentID)

Ucitelj (UciteljID)

Predmet (PredmetID)

Izpit (StudentID→Student, PredmetID→Predmet, Datum, Ocena, UciteljID→Ucitelj)