



Državni izpitni center



M 0 4 1 7 8 1 1 3

SPOMLADANSKI ROK

RAČUNALNIŠTVO

NAVODILA ZA OCENJEVANJE

Sreda, 9. junij 2004

SPLOŠNA MATURA

IZPITNA POLA 1

Obkrožite pravilno trditev.

1. Po vodilu lahko potekata dva prenosa podatkov istočasno. Trditev je:

(2 točki)

- A popolnoma pravilna;
- B popolnoma napačna;
- C delno pravilna, prenos je mogoč, če je vodilo dvosmerno;
- D delno pravilna, če je vodilo optično;
- E delno pravilna, če je vodilo serijsko.

2. Predpomnilnik poveča zmogljivost računalnika, ker

(2 točki)

- A so v njem vedno na razpolago vsi potrebni podatki in programi;
- B ima krajši čas dostopa kot glavni pomnilnik;
- C je lahko širina vodila med procesorjem in predpomnilnikom večja;
- D omogoča uvedbo dveh ločenih pomnilnikov: za podatke in ukaze;
- E vmesnih rezultatov ni treba shranjevati v registre.

3. Za dodeljevanje prostora na sekundarnem pomnilniku skrbi:

(2 točki)

- A razvrščevalnik opravil;
- B prevajalnik;
- C uporabnik z uporabo ukazov za odpiranje datotek;
- D aparaturna oprema krmilnika sekundarnega pomnilnika;
- E operacijski sistem.

4. Prevajalnik je:

(2 točki)

- A del systemske programske opreme;
- B del operacijskega sistema, ki olajša uporabo računalnika;
- C programska oprema, s katero spremenimo operacijski sistem;
- D programska oprema, ki omogoča delovanje računalnika;
- E nujni del operacijskega sistema.

5. V čem je bistvena razlika med dinamičnim (DRAM) in statičnim (SRAM) bralno-pisalnim pomnilnikom?

(2 točki)

- A Dinamični pomnilnik (DRAM) zahteva osveževanje, ima krajši dostopni čas, večjo kapaciteto in manjšo porabo energije.
- B Dinamični pomnilnik (DRAM) zahteva osveževanje, ima daljši dostopni čas, večjo kapaciteto in manjšo porabo energije.
- C Statični pomnilnik (SRAM) ne zahteva osveževanja, ima krajši dostopni čas in večjo kapaciteto na enoto volumna.
- D Statični pomnilnik (SRAM) zahteva osveževanje, ima daljši dostopni čas in večjo kapaciteto na enoto volumna.
- E Ni bistvenih razlik.

6. Dano je zaporedje ukazov v operacijskem sistemu:

MS-DOS:

```
cd delo
cd naloge
del ?ma?.txt
copy *.doc \users\tmp
cd..
```

UNIX/LINUX:

```
cd delo
cd naloge
rm ?ma?.txt
cp *.doc /users/tmp
cd ..
```

Obkrožite odgovor, ki v popolnosti opisuje dogajanje.

(2 točki)

- A V poddirektoriju naloge se brišejo vse datoteke, ki v imenu vsebujejo ma, in kopirajo vse datoteke s podaljškom doc na direktorij \users\tmp (MS DOS) ali /users/tmp (Unix/Linux).
- B Iz poddirektorija naloge se kopirajo vse datoteke s podaljškom doc na direktorij \users\tmp (MS DOS) ali /users/tmp (Unix/Linux), iz trenutnega direktorija se nato odstranijo vse datoteke, ki v imenu vsebujejo niz ma in imajo podaljšek txt.
- C Izbran je poddirektorij naloge, na izbranem poddirektoriju se brišejo datoteke, ki v imenu vsebujejo ma in imajo podaljšek txt, iz izbranega poddirektorija se kopirajo vse datoteke s podaljškom doc na direktorij \users\tmp (MS DOS) ali /users/tmp (Unix/Linux), izbran je ponovno koren direktorijev na disku C (MS-DOS) oziroma / (Unix/Linux).
- D V zaporedju ukazov je napaka in ukazi ne bodo izvedeni v celoti.
- E Nobeden od odgovorov A, B, C in D ne opisuje pravilnega dogajanja.

7. Kolikšna šestnajstiška vrednost je v akumulatorju A po izvedbi naslednjih ukazov?

```
ldaa #$3D
adda #$08
```

(2 točki)

- A \$46
- B \$55
- C \$54
- D \$45
- E \$A8

8. Pri izvajanju programa procesor določi vrsto naslavljanja in dolžino operanda tako, da:*(2 točki)*

- A iz ukaza izloči operacijsko kodo;
- B programer zapiše vsak ukaz v novo vrsto, procesor pa tako ugotovi dolžino ukaza in operanda;
- C iz dostavljenega ukaza razbere format ukaza;
- D je vrsta naslavljanja in dolžina operanda značilna in v vseh primerih enaka za neki procesor;
- E se dolžina operanda sklada z velikostjo delovnega registra.

Obkrožite pravilno trditev.**9. Deklarirana je spremenljivka:****Pascal:**

```
var x : string[10];
```

C:

```
char x[10];
```

Kateri ukaz priredi spremenljivki x vrednost okno?*(2 točki)***Pascal:**

- A x="okno";
- B x:='okno';
- C copy(x, 'okno');
- D insert(x, 'okno');
- E x:="okno";

C:

- A x="okno";
- B strcpy(x, "okno");
- C strcat(x, 'okno');
- D strcpy("okno", x);
- E strcat(x, "okno");

10. Želimo napisati podprogram za zamenjavo vrednosti spremenljivk (parametrov) x in y , ki sta realni števili.

Pravilna oblika glave podprograma je:

(2 točki)

Pascal:

- A procedure zamenjaj(var x,y : real);
- B procedure zamenjaj(var x: real; y : real);
- C procedure zamenjaj(x : real; var y : real);
- D procedure zamenjaj(x,y : real);
- E procedure zamenjaj(var x : real, y : real);

C:

- A void zamenjaj(float *x, float *y);
- B void zamenjaj(float *x, float y);
- C void zamenjaj(float *x,*y);
- D void zamenjaj(float x,y);
- E void zamenjaj(float x, float y);

11. Kaj izpiše naslednji program?

Pascal:

```
var a,b : integer;
begin
  a:=1; b:=20;
  while a<b do begin
    a:= a + (a mod 2) + 4;
    b:= 4 + (b div 2) ;
  end;
  writeln(a, ' ',b);
end.
```

C:

```
#include <stdio.h>
void main()
{
  int a=1,b=20;
  while (a<b) {
    a= a + (a % 2) + 4;
    b= 4 + (b / 2);
  }
  printf("%d %d", a,b);
}
```

(2 točki)

- A 10 8
- B 4 4
- C 11 10
- D 14 9
- E 5 5

12. Dan imamo podprogram za jemanje elementa iz sklada, ki je realiziran kot tabela realnih števil.

Pascal:

```
procedure Pop(var x :real);
begin
  if kazalec >0 then
  begin
    x := sklad[kazalec];
    dec(kazalec)
  end
  else
    write('Sklad je prazen');
end;
```

C:

```
void pop(float *x)
{
  if (kazalec>-1){
    *x=sklad[kazalec];
    kazalec--;
  }
  else
    printf("Sklad je prazen");
}
```

Ugotovite, kam se sklad širi in kam kaže kazalec sklada.

(2 točki)

- A Sklad se širi od manjših naslovov (indeksov) proti večjim, kazalec kaže na zadnjo zasedeno lokacijo.
- B Sklad se širi od manjših naslovov (indeksov) proti večjim, kazalec kaže na prvo prosto lokacijo.
- C Sklad se širi od večjih naslovov (indeksov) proti manjšim, kazalec kaže na prvo prosto lokacijo.
- D Sklad se širi od večjih naslovov (indeksov) proti manjšim, kazalec kaže na zadnjo zasedeno lokacijo.
- E Nobena trditev ne opisuje v nalogi zapisane realizacije sklada.

13. Indeks elementa tabele je lahko tipa:

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

Pascal:

- A integer
- B real
- C char
- D record
- E Lahko uporabljamo vse navede podatkovne tipe.

C:

- A int
- B float
- C char
- D struct
- E Lahko uporabljamo vse navede podatkovne tipe

14. Programska jezika Pascal in C sodita v skupino:

(2 točki)

- A strojnih jezikov,
- B zbirnih jezikov,
- C nepostopkovnih višjih programskih jezikov,
- D naravnih jezikov,
- E postopkovnih višjih programskih jezikov.

15. Če želimo dodati zapis na konec že odprte binarne tipizirane datoteke, moramo pred ukazom `write` oziroma `fwrite` uporabiti:

(2 točki)

Pascal:

- A `reset (f);`
- B `seek (f, filesize(f));`
- C `seek (f, 1);`
- D `seek (f, filesize(f)-1);`
- E `rewrite (f);`

C:

- A `f = fopen("ime_datoteke", "r");`
- B `fseek(f, 0, SEEK_END);`
- C `fseek(f, 1, SEEK_SET);`
- D `fseek(f, -1, SEEK_END);`
- E `f= fopen("ime_datoteke", "w");`

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 je pravilno napisan?

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.

17. Za zaposlenega potrebujemo naslednje podatke:

priimek (30 znakov),

ime (20 znakov),

osebne dohodke za zadnjih 12 mesecev (12 realnih števil).

Napišite deklaracijo tipa za zapis, ki bo ustrezal zgoraj navedenim zahtevam.

(2 točki)

Rešitev:

Pascal:

```
type TDelavec = record
    Priimek : String[30];
    Ime : String[20];
    OD : array[1..12] of real;
end;
```

C:

```
struct Tdelavec {
    char priimek[30]; //ali char priimek[31]
    char ime[20];    //ali char ime[21]
    float OD[12];
};
```

18. Na osnovi sledi spreminjanja tabelarične spremenljivke `Tabela` ugotovite, kateri algoritem je bil uporabljen pri urejanju.

Prehod	Tabela	
0	[2, 8, 9, 3, 1, 6]	začetna vrednost
1	[1, 8, 9, 3, 2, 6]	
2	[1, 2, 9, 3, 8, 6]	
3	[1, 2, 3, 9, 8, 6]	
4	[1, 2, 3, 6, 8, 9]	
5	[1, 2, 3, 6, 8, 9]	končna vrednost

(2 točki)

Rešitev:

izbiranje (selection)

19. V katero skupino programskih jezikov sodi jezik SQL?

(2 točki)

Rešitev:

Deklarativni, povpraševalni jezik, jezik za delo s podatkovnimi bazami.

20. Sledite izvajanju naslednjega programa in ugotovite, kaj ta program izpiše.*(2 točki)***Pascal:**

```
var x,y : integer;
begin
  x:=22; y:=10;
  while (x<50) or (y mod 2 =1) do
  begin
    x:=x+ (y div 2);
    y:=x + (y div 3);
    if ((y+x) div 2)<20 then
      y:=y+10;
    writeln(x, ' ', y);
  end;
end.
```

C:

```
#include <stdio.h>
void main()
{
  int x=22,y=10;
  while ((x<50) || (y%2==1)) {
    x=x+(y/2);
    y=x+(y/3);
    if ((x+y)/2<20)
      y=y+10;
    printf("%d %d \n", x, y);
  }
}
```

Rešitev:

27 30

42 52

68 85

110 138

Opomba: za prvi 2 ali 3 vrstice izpisa, kandidat dobi 1 točko.

21. 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;
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;
procedure izpis(t:tabela);
var i : integer;

    p : real;
begin
  p:=povprecje(t);
  for i:=1 to 100 do
    if (abs(p-t[i]) > (0.1 * p)) then
      writeln(t[i]);

end;
```

C:

```
#include <stdio.h>

#include <math.h>
float povprecje (unsigned int t[100])
{
  unsigned int i;
  float vsota = 0;
  for (i=0;i<100;i++)
    vsota += t[i];
  return (vsota/100);
}
void izpis (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]);
  }
}
```

22. Na disku imamo tipizirano datoteko REZULTAT.DAT. Zapis na datoteki sestavljata podatka: vzdevek osebe (niz 10 znakov) in rezultat (celo število). Napišite vse potrebne deklaracije in program, ki bo izpisal vzdevek in rezultat vseh oseb, ki imajo najboljši rezultat.

(4 točke)

Rešitev:

Pascal:

```

type zapis = record
    vzdevek : string[10];
    rezultat: integer;
end;
datoteka = file of zapis;
var f : datoteka;
function naj(var f : datoteka):integer;
var n : integer;
    x : zapis;
begin
    reset(f);
    read(f,x);
    n:=x.rezultat;
    while not(eof(f)) do begin
        read(f,x);
        if x.rezultat>n then
            n:=x.rezultat;
        end;
    naj:=n;
end;
procedure izpis_naj(var f : datoteka);
var najrez:integer;
    x : zapis;
begin
    najrez:=naj(f);
    reset(f);
    while not(eof(f)) do begin
        read(f,x);
        if x.rezultat=najrez then
            writeln(x.vzdevek, ' ',x.rezultat);
        end;
    end;
begin
    assign(f,'rezultat.dat');
    reset(f);
    izpis_naj(f);
    close(f);
end.

```

C:

```

#include <stdio.h>
struct zapis {
    char vzdevek[11];
    int rezultat;
};
int naj(FILE *f)
{
    int n=0;

```

```

    zapis x;
    fseek(f,SEEK_SET,0);
    fread(&x,sizeof(zapis),1,f);
    n=x.rezultat;
    while (!feof(f)) {
        fread(&x,sizeof(zapis),1,f);
        if (n<x.rezultat) n=x.rezultat;
    }
    return n;
}
void izpis_naj(FILE *f)
{
    zapis x;
    int najrez=naj(f);
    fseek(f,SEEK_SET,0);
    fread(&x,sizeof(zapis),1,f);
    while (!feof(f)) {
        if (x.rezultat==najrez)
            printf("%s %d \n",x.vzdevek,x.rezultat);
        fread(&x,sizeof(zapis),1,f);
    }
}
void main()
{
    FILE *f;
    f = fopen("rezultat.dat","r");
    izpis_naj(f);
    fclose(f);
}

```

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

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

(4 točke)

Rešitev:

Pascal:

```

(* rekurzivna rešitev *)
function vrni(n:integer):integer;
begin
    case n of
        1 : vrni:=1;
        2 : vrni:=5;
        else vrni:=vrni(n-1)+vrni(n-2);
    end;
end;

(* iterativna rešitev *)
function vrni(n:integer):integer;

var n1,n2,i : integer;
begin
    if (n=1) then vrni:=1;
    if (n=2) then vrni:=5;
    if (n>2) then
        begin
            n1:=1;

```

```
    n2:=5;
    for i:=3 to n do begin
        n2:=n1+n2;
        n1:=n2-n1;
    end;
    vrni:=n2;
end;
```

C:

```
int vrni(int n)
{
    switch (n) {
        case 1 : return 1;
        case 2 : return 5;
        default : return(vrni(n-1)+vrni(n-2));
    }
}
// iterativna rešitev
int vrni (int n)
{
    if (n==1) return (1);
    if (n==2) return (5);
    int n1=1, n2=5, i;
    for (i=3; i<=n; i++) {
        n2=n1+n2;
        n1=n2-n1;
    }
    return (n2);
}
```

24. Napišite program, ki bo na tekstovno datoteko izpisal števila v naslednji obliki:

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

(4 točke)

Rešitev:**Pascal:**

```
var f : text;
    x,y : integer;
begin
  assign(f,'test.txt');
  rewrite(f);
  for x:=5 downto 1 do begin
    for y:=1 to x do
      write(f,y,' ');
    writeln(f);
  end;
  close(f);
end.
```

C:

```
#include <stdio.h>
void main()
{
  FILE *f;
  int x,y;
  if ((f=fopen("test.txt","w+"))!=NULL) {
    for (x=5;x>=1;x--) {
      for (y=1;y<=x;y++){
        fprintf(f,"%d ",y);
      }
      fprintf(f,"\n");
    }
    fclose(f);
  }
}
```

IZPITNA POLA 2**SKLOP RAČUNALNIŠKE KOMUNIKACIJE**
NALOGE IZBIRNEGA TIPA

Obkrožite pravilno trditev.

1. Obkrožite pravilno nastavitev maske naslova IP razreda B.

(2 točki)

- A 255.0.0.0
- B 255.255.255.255
- C 255.255.0.0
- D 255.0.255.0
- E 255.255.255.0

2. Telnet spada na plast ISO/OSI referenčnega modela:

(2 točki)

- A aplikacijsko,
- B mrežno,
- C transportno,
- D fizično,
- E sejno.

3. Prednosti fizične topologije zvezde so:

(vsi pravilni odgovori – 2 točki
en pravilen ali dva odgovora – 1 točka
že en nepravilen odgovor – 0 točk)

- A ena točka porušitve;
- B večja propustnost od drugih topologij;
- C manjša dolžina fizičnih vodov;
- D enostavnejši nadzor in upravljanje omrežja od drugih topologij;
- E lažje lociranje mesta napake.

4. Žeton v omrežju omogoča, da:*(2 točki)*

- A vsaka postaja »ve«, kdaj lahko odda paket, kdaj pa ga lahko le sprejme;
- B nekatere postaje pridejo na vrsto pogosteje, kakor druge, ker imajo žeton;
- C je delovanje omrežja lažje razumljivo uporabniku;
- D omrežje ni nikoli preobremenjeno zaradi velikega prometa;
- E paket poteka vedno v istem zaporedju med postajami.

5. Tipičen predstavnik sistema za podporo odločanju je:*(2 točki)*

- A transakcijski sistem, namenjen beleženju poslovnih dogodkov;
- B sistem, ki omogoča analizo podatkov v smislu »kaj-če«;
- C sistem, ki omogoča komunikacijo med zaposlenimi;
- D sistem, ki podpira skupinsko delo;
- E sistem za avtomatizacijo pisarniškega poslovanja.

6. Razlike med primarnim in sekundarnim indeksom (ključem) tabele relacijske podatkovne baze so:*(2 točki)*

- A Primarni indeks tabele je en sam, sekundarnih je lahko več, primarni indeks je razločevalen, sekundarni so lahko razločevalni ali nerazločevalni.
- B Primarni indeks tabele je en sam, sekundarnih je lahko več, primarni indeks je razločevalen, sekundarni so lahko samo nerazločevalni.
- C Primarni indeks tabele je en sam, sekundarnih je lahko več, primarni in sekundarni indeksi so razločevalni ali nerazločevalni.
- D Primarnih in sekundarnih indeksov je lahko več, lahko so razločevalni ali nerazločevalni.
- E Vse trditve so napačne.

7. Sistem številčenja procesov v diagramih toka podatkov omogoča, da*(2 točki)*

- A ugotovimo globino posameznega procesa oziroma identificiramo diagram, na katerem je proces;
- B ugotovimo zaporedje izvajanja posameznih procesov;
- C procesov ni treba poimenovati;
- D lažje opišemo podatkovne tokove v podatkovnem slovarju;
- E uporabniku predstavimo obseg zahtevane funkcionalnosti.

8. Na entitetno–relacijskem diagramu lahko zasledimo:*(vsi pravilni odgovori 2 točki
1 ali 2 pravilna odgovora 1 točka
en napačen – 0 točk)*

- A podatkovne tokove in njihov izvor/ponor;
- B kardinalnost oziroma števnost povezav (relacij);
- C kontrolne tokove;
- D relacije;
- E entitete s pripadajočimi atributi.

9. Obkrožite vse pravilne trditve. Katere faze načrtovanja podatkovne baze so neodvisne od sistema za upravljanje podatkovne baze (SUPB-ja)?*(vsi pravilni odgovori 2 točki
en napačen – 0 točk)*

- A Zbiranje in analiza zahtev.
- B Izdelava konceptualnega modela.
- C Logično načrtovanje.
- D Fizično načrtovanje.
- E Vse faze so odvisne od izbranega SUPB-ja.

10. Katero rezervirano besedo jezika SQL bi uporabili v ukazu, s katerim bi želeli spremeniti strukturo (ne podatkov!) tabele?

(2 točki)

- A ALTER
- B INSERT
- C SELECT
- D UPDATE
- E DROP

11. Naštejte vsaj štiri osnovne funkcije informacijskih sistemov:

(3 ali več pravih odgovorov – 2 točki
2 pravilna odgovora – 1 točka)

Odgovor:

Zajem podatkov

bodisi prek tipkovnic, čitalnikov črtne kode, zunanjih datotek.

Hranjenje

zajetih in preoblikovanih podatkov za potrebe nadaljnje uporabe.

Iskanje

po različnih merilih, iskanje je temelj funkciji prikazovanje.

Preoblikovanje

pridobivanje novih podatkov na osnovi že obstoječih + spreminjanje + brisanje.

Prikazovanje

različni načini in oblike prikaza.

12. Navedite vsaj 4 težave pri razvoju programske opreme:*(2 točki)*

predolgi razvojni cikli, nizka produktivnost osebja in skupin, nepredvidljiva kakovost

končnega proizvoda, pomanjkanje strokovnjakov za razvoj programske opreme, visoki stroški

razvoja, visoki stroški vzdrževanja

13. Razen funkcionalnih zahtev v specifikacijah zahtev za programsko opremo običajno navedemo tudi nefunkcionalne zahteve. Zapišite vsaj tri primere nefunkcionalnih zahtev.*(3 ali več pravih odgovorov – 2 točki
2 pravilna odgovora – 1 točka)*

zanesljivost

varnost

prenosljivost

dostopnost

možnost vzdrževanja

možnost testiranja

upoštevanje standardov

14. Za atomarne (elementarne) procese diagramov toka podatkov izdelamo minispecifikacije, v katerih opišemo preslikavo vhodnih podatkov v izhodne. V ta namen lahko razen strukturiranega besedila uporabimo tudi:*(2 pravilna odgovora – 2 točki
1 pravih odgovor – 1 točka)*

odločitvene tabele

odločitvena drevesa

SKLOP INFORMATIKA
KOMPLEKSNE NALOGE

15. Z notacijo podatkovnega slovarja opišite podatkovni tok Račun. Uporabite vse konstrukte podatkovnega slovarja. Naslov prejemnika računa in dostave nista nujno enaka. Različnih izdelkov na računu je lahko največ pet. Za vsako postavko računa je na računu navedena enota mere (kos, liter ali meter), količina, cena brez DDV, stopnja DDV in skupna cena. Stopnja davka na dodano vrednost je lahko 20 %, 8,5 % ali 0 %.

(4 točke)

2 točki – pravilno uporabljena vsaj 2 konstrukta)

racun = naslovprejemnika + (naslovdostave) + 1 {postavka} 5

naslovprejemnika = naslov

naslovdostave = naslov

naslov = poštnaštevilka + kraj + ulica + hišnaštevilka

postavka = enotamere + kolicina + cenabrezDDV + stopnjaDDV + skupnacena

enotamere = [kos | liter | meter]

stopnjaDDV = [20 | 8,5 | 0]

16. Podatkovno bazo sestavljajo tabele:

Dijak(Sifra_dijaka, Priimek_dijaka, Ime_dijaka, Razred)

Predmet(Sifra_predmeta, Ime_predmeta)

Uspeh(Sifra_predmeta, Sifra_dijaka, Ocena)

Napišite stavek SQL, ki izpiše abecedno urejen seznam dijakov razreda G1A.

(1 točka)

SELECT Sifra_dijaka, Priimek_dijaka, Ime_dijaka, Razred // lahko tudi *

FROM Dijak

WHERE Razred = 'G1A'

ORDER BY Priimek_dijaka, Ime_dijaka // lahko tudi le Priimek_dijaka

Napišite stavek SQL, ki izpiše šifre, priimke, imena in povprečne ocene dijakov (upoštevajte ocene pri vseh predmetih in ne za vsak predmet posebej).

(3 točke)

```
SELECT Sifra_dijaka, Priimek_dijaka, Ime_dijaka, Avg(Ocena)
```

```
FROM Dijak, Uspeh
```

```
WHERE Dijak.Sifra_dijaka = Uspeh.Sifra_dijaka
```

```
GROUP BY Dijak.Sifra_dijaka, Dijak.Priimek_dijaka, Dijak.Ime_dijaka
```
