



Državni izpitni center



M 0 6 1 7 8 1 1 3

SPOMLADANSKI ROK

RAČUNALNIŠTVO

NAVODILA ZA OCENJEVANJE

Sobota, 10. junij 2006

SPLOŠNA MATURA

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

Obkrožite pravilne trditve.

1. Stalno prisotni program BIOS je shranjen v pomnilniku, katerega izvedbo imenujemo:

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

- A RAM
- B DISK
- C EEPROM
- D ROM
- E USB key

2. Pri izvedbi maskirane vektorske prekinitve najprej:

(2 točki)

- A procesor prevzame s sklada povratni naslov;
- B periferna enota pošlje procesorju prekinitveni vektor;
- C procesor ugotovi začetni naslov prekinitvenega strežnega programa;
- D procesor začne izvajati prekinitveni strežni program;
- E periferna enota sproži signal za prekinitveno zahtevo.

3. V ukazu procesorja sta lahko prisotna naslova dveh operandov. To je:

(2 točki)

- A nemogoče;
- B res, če lahko na vhoda aritmetično-logične enote vodimo vsebini dveh različnih pomnilniških lokacij;
- C res, če ima ukaz dovolj binarnih mest;
- D res, kadar je v ukazu prisoten tudi naslov za rezultat operacije;
- E vedno mogoče.

4. Procesor lahko naenkrat računa rezultat samo ene operacije. Trditev je:

(2 točki)

- A pravilna;
- B pravilna, če procesor nima več aritmetično-logičnih enot;
- C pravilna, če procesor deluje kot cevovod;
- D vedno napačna;
- E pravilna, če ima procesor več aritmetično-logičnih enot.

5. Lupina predstavlja v operacijskem sistemu vmesnik med:

(2 točki)

- A operacijskim sistemom in strojno opremo;
- B uporabnikom in strojno opremo;
- C uporabnikom in operacijskim sistemom;
- D več uporabniki;
- E ne predstavlja vmesnika.

6. Ali je lahko ista datoteka prisotna v več seznamih pod različnimi imeni?

(2 točki)

- A To ni mogoče.
- B Da, če sezname niso koren datotečnega sistema.
- C To je vedno mogoče.
- D Da, če datotečni sistem omogoča simbolično povezovanje (link) datotek.
- E To je mogoče, če so ti sezname na fizično različnih pomnilniških enotah.

7. Vzrok za fragmentacijo (razdrobljenost) diska je predvsem odstranjevanje datotek. Posledica razdrobljenosti pa je:

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

- A dobra izkoriščenost pomnilnega prostora;
- B podaljšanje dostopnih časov;
- C skrajšanje dostopnih časov;
- D slaba izkoriščenost pomnilnega prostora;
- E skrajšanje odzivnih časov.

8. Ali lahko z dvema ukazoma operacijskega sistema Linux izpišemo vse procese, ki jih izvaja neki uporabnik?

(2 točki)

- A Lahko.
- B Ne moremo.
- C Lahko, če je operacijski sistem večprocesorski.
- D Lahko, če procesi niso aktivni.
- E Lahko, če je vsaj en proces neaktiven.

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
NALOGE IZBIRNEGA TIPA

Obkrožite pravilno trditev.

9. Kaj se izpiše med izvajanjem naslednjega zaporedja stavkov:

Pascal:

```
for m:=0 to 3 do
  for n:= 0 to 2 do
    write(m:2,n:2);
```

C:

```
for (int m=0;m<=3;m++){
  for (int n=0;n<=2;n++){
    printf("%2d%2d",m,n);}}
```

(2 točki)

Pascal:

- A 3 2
- B 0 1 2 3 0 1 2
- C 0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2 3 0 3 1 3 2
- D 0 0 0 1 0 2 0 3 1 0 1 1 1 2 1 3 2 0 2 1 2 2 2 3
- E 2 2

10. Za zanko `while` je značilno, da:

(2 točki)

- A se pogoj preverja na koncu, stavki v telesu zanke se izvajajo, dokler pogoj ni izpolnjen;
- B se stavki v telesu zanke izvedejo vsaj enkrat;
- C se stavki v telesu zanke nikoli ne izvedejo, če že na začetku zanke ni izpolnjen pogoj;
- D jo uporabimo, kadar je točno znano število ponavljanj;
- E se pogoj preverja na začetku, stavki v telesu zanke pa se izvajajo, dokler je pogoj napačen.

11. Deklarirane so spremenljivke**Pascal:**

```
var x,y : integer; z: real;
```

C:

```
int x,y; float z;
```

Kateri stavek priredi spremenljivki z rezultat realnega deljenja spremenljivke x s spremenljivko y?

*(2 točki)***Pascal:**

- A `z:= x div y;`
- B `z:= x % y;`
- C `z:= x mod y;`
- D `z:= x divide y;`
- E `z:= x / y;`

C:

- A `z= x/y;`
- B `z= x % y;`
- C `z=float(x/y);`
- D `z= x divide y;`
- E `z= float(x)/y;`

12. Imamo tabelo desetih celih števil. Podatki v tabeli so urejeni v naraščajočem vrstnem redu. Kateri od navedenih iskalnih algoritmov je najhitrejši:*(2 točki)*

- A algoritem zaporednega iskanja;
- B algoritem linearnega iskanja;
- C algoritem naključnega iskanja;
- D algoritem binarnega (dvojiškega) iskanja;
- E hitrost vseh navedenih algoritmov je enaka.

13. Kaj od navedenega sodi v skupino programskih orodij za hiter razvoj aplikacij (orodja RAD):*(vsaj en pravilen odgovor 2 točki, že en nepravilen odgovor 0 točk)*

- A Microsoft Visual Studio;
- B Turbo Pascal;
- C mySQL;
- D Delphi;
- E Windows Movie Maker.

14. Dana je deklaracija in stavki za odpiranje datoteke:**Pascal:**

```
var f : text;  
assign(f, 'vaja'); reset(f);
```

C:

```
FILE *f;  
f=fopen("vaja", "r");
```

Katera trditev velja po izvedbi stavka `reset` oziroma `fopen`?*(2 točki)*

- A Vsebina datoteke se ohrani, datotečni kazalec kaže na začetek datoteke, lahko izvajamo operacije branja in zapisovanja v datoteko.
- B Vsebina datoteke se izbriše, datotečni kazalec kaže na začetek datoteke, lahko izvajamo operacije branja in zapisovanja v datoteko.
- C Vsebina datoteke se ohrani, datotečni kazalec kaže na začetek datoteke, lahko izvajamo le operacije branja iz datoteke.
- D Vsebina datoteke se ohrani, datotečni kazalec kaže na konec datoteke, lahko izvajamo le operacije branja iz datoteke.
- E Vsebina datoteke se izbriše, datotečni kazalec kaže na začetek datoteke, lahko izvajamo le operacije zapisovanja v datoteko.

15. Programski jeziki Pascal, C in Basic sodijo v skupino:*(2 točki)*

- A strojnih jezikov;
- B višjih programskih jezikov;
- C zbirnikov;
- D jezikov HTML;
- E povpraševalnih (query) jezikov.

16. Kateri sortirni algoritem v povprečju primerja $n \cdot \log(n)$ elementov, če je le-teh n ?*(2 točki)*

- A Sortiranje z mehurčki.
- B Quicksort.
- C Sortiranje z vstavljanjem.
- D Sortiranje z izbiranjem.
- E Popravljeno sortiranje z mehurčki.

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
NALOGE S KRATKIMI ODGOVORI

17. Napišite stavek `case` (Pascal) oziroma `switch` (C), ki bo enakovreden naslednjim stavkom `if`:

(2 točki)

Pascal:

```
if ocena = 6 then
  writeln('zadostno')
else
  if ocena = 7 then
    writeln('dobro')
  else
    if (ocena=8) or (ocena=9) then
      writeln('prav dobro')
    else
      if ocena=10 then
        writeln('odlično')
      else
        writeln('nezadostno');
```

C:

```
if (ocena == 6){
  printf("zadostno \n");}
else {
  if (ocena ==7) {
    printf("dobro");}
  else {
    if ((ocena==8) || (ocena==9)) {
      printf("prav dobro \n");}
    else {
      if (ocena==10) {
        printf("odlično \n");}
      else {
        printf("nezadostno \n");};};};}
```

Rešitev:**Pascal:**

```
case ocena of
  6:writeln('zadostno');
  7:writeln('dobro');
  8,9:writeln('prav dobro');
  10: writeln('odlično')
  else writeln('nezadostno');
end;
```

C:

```
switch (ocena) {
  case 6: printf("zadostno \n");break;
  case 7: printf("dobro \n");break;
  case 8: printf("prav dobro \n");break;
  case 9: printf("prav dobro \n");break;
  case 10: printf("odlično \n");break;
  default: printf("nezadostno \n");
};
```

18. Naslednjo iteracijo zapišite z zanko while:**Pascal:**

```
for x:='A' to 'F' do begin
  writeln (pred(x), ' ', succ(x));
  writeln(x:2);
end;
```

C:

```
for (x='A';x<='F';x++) {
  printf("%c %c\n",x-1,x+1);
  printf("%2c\n",x);
}
```

*(2 točki)***Rešitev:****Pascal:**

```
x:='A';
while (x<='F') do
begin
  writeln (pred(x), ' ', succ(x));
  writeln(x:2);
  inc(x);
end;
```

C:

```
x='A';
while (x<='F') {
  printf("%c %c\n",x-1,x+1);
  printf("%2c\n",x);
  x++;
}
```

19. Kaj izpiše naslednji program?**Pascal:**

```
var a,b,c : integer;
begin
  a:=17; b:=3; c:=0;
  while (a>b) do begin
    inc(c);
    a:=a-b;
  end;
  writeln(c, ' ', a);
end.
```

C:

```
#include <stdio.h>
int main()
{
  int a=17,b=3,c=0;
  while (a>b) {
    c++;
    a-=b;
  }
  printf("%d %d\n",c,a);
  return 0;
}
```

Rešitev:

20. Kaj naredi naslednji program?*(2 točki)***Pascal:**

```
var stevilo:integer;
begin
  for stevilo := 1 to 10 do
    begin
      writeln('stevilo: ', stevilo);
      writeln('kvadrat: ', sqr(stevilo));
      writeln('kub: ', stevilo*(sqr(stevilo)));
    end;
end.
```

C:

```
int c=1;
int main(){
  for(int stevilo=1;stevilo<=10;stevilo++){
    printf("stevilo: %d \n", stevilo);
    printf("kvadrat: %d \n", stevilo*stevilo);
    printf("kub: %d \n", stevilo*stevilo*stevilo);
  };
return 0;}
```

Rešitev:

Za vsako število od 1 do 10 izpiše število, kvadrat števila in kub števila.

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
STRUKTURIRANE NALOGE

21. V tekstovni datoteki 'test.txt' so zapisana cela števila.

Napišite podprogram, ki vrne povprečno vrednost zapisanih števil. Če je datoteka prazna (nima zapisanih števil), naj podprogram vrne 0.

(2 točki)

Napišite podprogram, ki vrne število vrstic v datoteki 'test.txt'.

(1 točka)

Napišite glavni program s klici podprogramov.

(1 točka)

Rešitev:

```
var f : text;

function povprecje(var f:text):real;
var x,vs,stevec:integer;
begin
  vs:=0; stevec:=0;
  reset(f);
  while (not(eof(f))) do begin
    read(f,x);
    vs:=vs+x;
    inc(stevec);
  end;
  if stevec=0 then
    povprecje:=0
  else
    povprecje:=vs/stevec;
end;

function stevilo_vrstic(var f:text):integer;
var stevec : integer;
begin
  stevec:=0;
  reset(f);
  while (not(eof(f))) do begin
    readln(f);
    inc(stevec);
  end;
  stevilo_vrstic:=stevec;
end;
begin
  assign(f,'test.txt');
  {$I-}
  reset(f);
  if (IOResult=0) then begin
    Writeln('Povprečna vrednost=',povprecje(f):7:1);
    Writeln('Število vrstic=',stevilo_vrstic(f));
    close(f);
  end;
  {$I+}
end.
```

C:

```
#include <stdio.h>
float povprecje(FILE *f) {
    int x, vs=0, stevec=0;
    rewind(f);
    do {
        fscanf(f, "%d", &x);
        if (!feof(f)) {
            vs+=x;
            stevec++;
        }
    } while (!feof(f));
    if (stevec==0) (stevec++);
    return ((float(vs)/stevec));
}
int stevilo_vrstic(FILE *f)
{
    char ch;
    rewind(f);
    int stevec=0;
    do {
        ch=fgetc(f);
        if (ch=='\n') stevec++;
    }
    while (ch!=EOF);
    return(stevec);
}
int main()
{
    FILE *f;
    if ((f=fopen("test.txt", "r"))!=NULL) {
        printf("Stevilo vrstic= %d \n", stevilo_vrstic(f));
        printf("Povprečna vrednost= %7.1f", povprecje(f));
        fclose(f);
    }
    return 0;
}
```

22. Napišite podprogram `crkaV`, ki glede na vrednost parametra `stVrstic` z zvezdicami izpiše sliko v obliki črke V. Prikazani so primeri izpisa za vrednost parametra 3, 4 in 5.

(4 točke)

```

**  **      **      **      **      **
****      **  **      **      **      **
**              ****      **
                **

```

Rešitev:

```

var i,j,n,stVrstic:integer;
procedure crkaV(stVrstic:integer);
var z:char;
begin
  z:=' ';
  for i:=stVrstic downto 2 do
    begin
      for n:=0 to stVrstic-i do
        write(z);
      write('**');
      for j:=1 to i-2 do
        write(z,z);
      writeln('**');
    end;
  for n:=1 to stVrstic do
    write (z);
  writeln('**');
end;

```

C:

```

#include <stdio.h>

int stVrstic;
char z=' ';
void crkaV(stVrstic){
  for(int i=stVrstic;i>1;i--){
    for (int n=0; n<=stVrstic-i; n++){
      printf("%c",z); }
    printf("**");
    for (int j=1;j<=i-2;j++){
      printf("%c%c",z,z); }
    printf("**\n");
  };
  for (int n=1;n<=stVrstic;n++){
    printf("%c",z);}
  printf("**\n");
}

```

23. **Dani sta dve enako dolgi tabeli celih števil (tabeli a in b). Napišite podprogram, ki ugotovi, ali sta v obeh tabelah shranjeni različni permutaciji istih števil. Upoštevajte naslednje deklaracije:**

(4 točke)

Pascal:

```
const n = 5;
type tabela=array[1..n] of integer;
function staPermutaciji (a,b:tabela):boolean;
```

C:

```
const n = 5;
int a[n],b[n];
int staPermutaciji(int a[n],int b[n]);
```

Rešitev:

```
function staPermutaciji (a,b:tabela):boolean;
var ax,bx,i,j:integer;
    je:boolean;
begin
    je:=true;
    for i:=2 to n do
        for j:=n downto i do begin
            if a[j-1]>a[j] then
                begin
                    ax:=a[j-1]; a[j-1]:=a[j]; a[j]:=ax;
                end;
            if b[j-1]>b[j] then
                begin
                    bx:=b[j-1]; b[j-1]:=b[j]; b[j]:=bx;
                end;
            end;
        for i:=1 to n do
            if a[i]<>b[i] then
                je:=false;
        staPermutaciji:=je;
end;
```

C:

```
const n = 5;
int a[n],b[n];
int staPermutaciji(int a[n],int b[n]);
int staPermutaciji(int a[n], int b[n]){
int ax,bx,je=1;
for (int i=1;i<n;i++){
    for (int j=n-1;j>=i;j--){
        if (a[j-1]>a[j]){
            ax=a[j-1]; a[j-1]=a[j]; a[j]=ax;}
        if (b[j-1]>b[j]){
            bx=b[j-1]; b[j-1]=b[j]; b[j]=bx;} ;
    };};
for (int i=0;i<n;i++){
```

```

    if (a[i]!=b[i]) {je=0;}
    }
    return je;
} ;

```

- 24. Deklarirajte podatkovno strukturo avto z dvema komponentama: znamka (niz 80 znakov) in hitrost (celo število). Deklarirajte tabelo desetih takšnih struktur. Napišite podprogram za vnos podatkov v tabelo in podprogram za iskanje ter izpis hitrosti, ki jo doseže najhitrejši avto.**

(4 točke)

Rešitev:

```

type avto=record
    znamka:string[80];
    hitrost:integer;
end;
var tabela:array[1..10] of avto;
procedure vnos;
    var i:integer;
    begin
        for i:=1 to 10 do begin
            write('znamka: ');
            readln(tabela[i].znamka);
            write('hitrost: ');
            readln(tabela[i].hitrost);
        end;
    end;

procedure najhit;
    var i,hit:integer;
    begin
        hit:=0;
        for i:=1 to 10 do begin
            if tabela[i].hitrost>hit then
                hit:=tabela[i].hitrost;
        end;
        writeln('najhitrejši avto pelje s hitrostjo ',hit,' km/h');
    end;

```

C:

```

#include <stdio.h>
typedef struct{
    char znamka[80];
    int hitrost;
} avto;
avto tabela[10];

void vnos(){
    for (int i=0;i<10;i++)
    { printf("znamka: ");
      scanf("%s",tabela[i].znamka);
      printf("hitrost: ");
      scanf("%d",&tabela[i].hitrost);
    }
};

```

```
void najhit()
{
    int hit=0;
    for (int i=0;i<10;i++)
        { if (tabela[i].hitrost>hit){
            hit=tabela[i].hitrost;
        };
        };
    printf("najhitrejši avto pelje s hitrostjo %d km/h",hit);
};
```

IZPITNA POLA 2

Obkrožite pravilno trditev.

1. Aplikacijska storitev FTP (File Transfer Protocol) se na transportni plasti povezuje s/z:

(2 točki)

- A protokolom FTP komunikacijskega modela TCP/IP;
- B protokolom TCP ali UDP, odvisno od velikosti datoteke, ki se prenaša;
- C protokolom UDP, če ne potrebujemo prenosa brez napak;
- D protokolom TCP, ki vključuje tudi potrjevanje prenosov;
- E nobenim od naštetih protokolov.

2. Katera je glavna prednost protokola CSMA/CD?

(2 točki)

- A Je zelo učinkovit pri veliki obremenjenosti omrežja, saj onemogoča sinhronizacijo trkov po sprostitvi prenosnega medija. Ko ugotovi zasedenost prenosnega medija, sproži časovno kontrolo in šele po izteku ponovno preverja zasedenost prenosnega medija.
- B Oddajniku omogoča ugotavljanje zasedenosti prenosnega kanala in ugotavljanje trkov. Ker ima možnost ugotavljanja trkov, lahko takoj po trku ustavi oddajo in tako sprosti prenosni medij.
- C Prednost je v enostavnosti, saj ne preverja zasedenosti prenosnega medija in ne glede na zasedenost odda paket.
- D Izključuje možnost trkov z uporabo faze rezervacije oziroma žetonov. Le postaja, ki ima žeton, lahko odda paket in tako ne pride do trkov.
- E Ponovni poskusi oddaje niso potrebni.

3. Glavni namen preklapljanja je usmerjanje sporočil. To se dogaja na:

(2 točki)

- A aplikacijski plasti referenčnega modela ISO/OSI;
- B transportni plasti referenčnega modela ISO/OSI;
- C omrežni plasti referenčnega modela ISO/OSI;
- D fizični plasti referenčnega modela ISO/OSI;
- E sejni plasti referenčnega modela ISO/OSI.

4. Most (bridge):

(2 točki)

- A povezuje LAN-e;
- B povezuje WAN-e;
- C ojača signale na fizični plasti;
- D povezuje omrežja z različnimi protokoli;
- E loči logični segment omrežja na fizične segmente.

5. Katere aktivnosti se izvajajo v prvi fazi razvoja informacijskega sistema:

(2 točki)

- A programiranje;
- B zbiranje in analiza zahtev uporabnikov;
- C izdelava načrta podatkovne baze in aplikacij;
- D izdelava testnih primerov;
- E izbira programske in strojne opreme.

6. Katero tehniko lahko uporabimo pri pisanju minispecifikacij procesov:

(vsi odgovori 2 točki, en odgovor 1 točka, že en nepravilen odgovor 0 točk)

- A psevdokodo;
- B entitetno-relacijske diagrame;
- C odločitvene tabele;
- D odločitvena drevesa;
- E diagrame prehajanja stanj.

7. Kateri gradniki se NE pojavljajo na kontekstnem nivoju DTP-jev:

(vsi odgovori 2 točki, en odgovor 1 točka, že en nepravilen odgovor 0 točk)

- A zbirke;
- B zunanje entitete;
- C procesi;
- D tokovi podatkov;
- E atributi.

8. Kateri od navedenih uporabnikov podatkovne baze je odgovoren za spreminjanje njene strukture:

(2 točki)

- A sistemski analitik;
- B sistemski programer;
- C aplikacijski programer;
- D skrbnik (administrator) podatkovne baze;
- E vsi navedeni uporabniki lahko spreminjajo strukturo podatkovne baze.

9. Kaj NI namen vzdrževanja programske opreme:

(2 točki)

- A pocenitev programa;
- B izboljšanje (dopolnjevanje) funkcij programa;
- C odpravljanje napak v programu;
- D podaljšanje življenjske dobe programa;
- E večje zadovoljstvo uporabnikov s programom.

10. Katera od navedenih značilnosti velja za entitetno-relacijske diagrame:

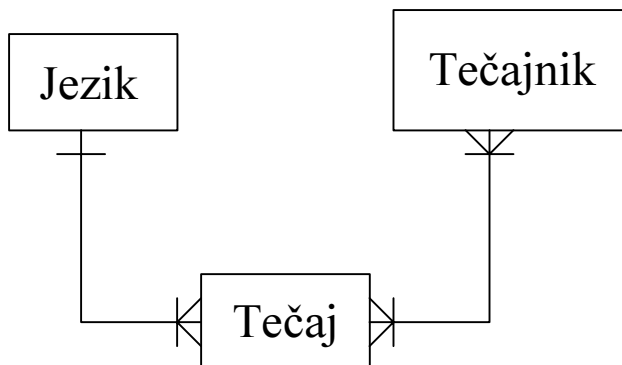
(vsi odgovori 2 točki, en odgovor 1 točka, že en nepravilen odgovor 0 točk)

- A prikazujejo vse podatke in procese v informacijskem sistemu;
- B sodijo med grafične načine predstavitev podatkovnega modela;
- C ponazarjajo entitetne tipe in povezave med entitetnimi tipi;
- D prikazujejo logično in fizično predstavitev podatkov v podatkovni bazi;
- E nazorno prikazujejo strukturo sistema za upravljanje podatkovne baze.

11. Podjetje Poliglot izvaja tečaje tujih jezikov. Za vsak jezik je razpisanih več tečajev, na vsak tečaj se lahko prijavi več tečajnikov. Isti tečajnik lahko obiskuje več tečajev (npr. tečaj nemščine in italijanščine). Narišite entitetno-relacijski diagram, iz katerega bodo razvidna razmerja med entitetnimi tipi Jezik, Tečaj in Tečajnik.

(2 točki)

Odgovor:



12. Dana je tabela `Dijak` s podatki:

DijakID	Priimek	Ime	Razred
003	Dolinar	Matej	G1A
004	Mlinar	Marko	G1A
008	Kralj	Matej	G1A
010	Koželj	Peter	G1C
012	Dolenc	Miha	G1B
013	Mihevc	Matjaž	G1A

Kaj vrne naslednji stavek SQL?

(2 točki)

```
SELECT Ime FROM Dijak WHERE Razred='G1A';
```

Odgovor:

Matej, Marko, Matej, Matjaž

13. Zakaj uporabljamo indekse?

(2 točki)

Odgovor:

Za hitrejši dostop do podatkov.

14. Navedite vsaj tri prednosti uporabe računalniško podprtih informacijskih sistemov:

(2 točki)

Odgovor:

Hitrejša obdelava podatkov, hitrejši prenos podatkov, enostavno oblikovanje izhodnih rezultatov,
avtomatizacija opravil, zmožnost hranjenja velike količine podatkov, hiter dostop do podatkov.

15. Podatkovno bazo sestavljajo naslednje tabele:

Krozek (KrozekID:N, ImeKrozka:A20)

Dijak (DijakID:N, Priimek:A20, Ime:A10, Razred:A3)

Obiskuje (DijakID:N, KrozekID:N, Solsko_let0:A9)

Narišite ustrezni entitetno-relacijski diagram.

(1 točka)

Rešitev:

[1.1]



Napišite stavek SQL, ki izpiše abecedni seznam priimkov in imen dijakov, obiskovalcev krožka PHP.

(3 točke)

Rešitev:

SELECT D.Priimek, D.Ime

FROM Krozek K, Obiskuje O, Dijak D

WHERE (K.KrozekID=O.KrozekID) and (O.DijakID=D.DijakID) and (K.ImeKrozka='PHP')

ORDER BY D.Priimek, D.Ime;

16. Za elementarni proces vzgojni_ukrepi napišite minispecifikacijo v obliki odločitvene tabele. Vhoda v proces sta število neopravičenih ur in predhodni ukrep. Izhod iz procesa pa je izrečeni ukrep.

- Če je neopravičenih ur manj ali enako 10, ni vzgojnega ukrepa.
- Če je število neopravičenih ur med 10 in 30, dobi dijak ukor razrednika, če zanj še ni bil izrečen predhodni ukrep.
- Če je število neopravičenih ur med 10 in 30, dobi dijak ukor oddelčnega učiteljskega zbora, če je bil zanj že izrečen predhodni ukrep.
- Če je število neopravičenih ur večje ali enako 30, dobi dijak ukor ravnatelja, če zanj še ni bil še izrečen predhodni ukrep.
- Če je število neopravičenih ur večje ali enako 30, dobi dijak ukor celotnega učiteljskega zbora, če je bil zanj že izrečen predhodni ukrep.

(4 točke)

Rešitev:

število_neopravicjenih_ur predhodni ukrep	ure<10	10<ure<30	10<ure<30	30<ure	30<ure
	-	ne	da	ne	da
podeljeni ukrep	ne	UR	UOZ	URA	UCUZ