



Državni izpitni center



M 0 5 1 7 8 1 1 3

SPOMLADANSKI ROK

RAČUNALNIŠTVO

NAVODILA ZA OCENJEVANJE

Četrtek, 16. junij 2005

SPLOŠNA MATURA

Popravljená različica

IZPITNA POLA 1

Obkrožite pravilno trditev.

1. Med osnovne naloge jedra operacijskega sistema NE sodi:

(2 točki)

- A razvrščanje procesov,
- B skrb za datotečni sistem,
- C upravljanje glavnega pomnilnika,
- D postavitve uporabniškega vmesnika,
- E zagotavljanje dostopa do perifernih enot.

2. Koliko prostora na disku zavzame datoteka, ki je dolga 3 zloge? Disk je organiziran na skupke (clusters), dolge 2 bloka (2*512 zlogov = 1024 zlogov).

(2 točki)

- A 1024 zlogov,
- B 512 zlogov,
- C 3 zloge,
- D 10 zlogov,
- E 4 zloge.

3. V operacijskem sistemu Linux se prijavite kot *Uporabnik1*. Ali lahko pregledujete domači imenik uporabnika *Uporabnik2*?

(2 točki)

- A Lahko, če so dovolilnice za imenik naslednje:
`drwx --x --x 4 Uporabnik2 group2 512 jul31 11:05 up2/`
- B Uporabnik *Uporabnik1* ne more pregledovati domačega imenika uporabnika *Uporabnik2*.
- C Lahko, če so dovolilnice za imenik naslednje:
`drwx r-x r-x 4 Uporabnik2 group2 512 jul31 11:05 up2/`
- D Uporabnik, ki je prijavljen kot *Uporabnik1*, lahko vedno pregleduje imenik uporabnika *Uporabnik2*, nikakor pa ne more izvajati datotek na tem imeniku.
- E Uporabnik *Uporabnik1* lahko vedno pregleduje imenik uporabnika *Uporabnik2*, nikakor pa ne more brisati datotek na tem imeniku.

4. Posamezni deli mikrokrmilnika so med seboj povezani z vodili. Glede na funkcijo signalov, ki se prenašajo po vodilih, ločimo naslednja vodila:

(2 točki)

- A naslovno, podatkovno in notranje vodilo;
- B naslovno, notranje in zunanje vodilo;
- C podatkovno, krmilno in zunanje vodilo;
- D podatkovno, naslovno in krmilno vodilo;
- E vhodno, izhodno in podatkovno vodilo.

5. Pomnilniška beseda je definirana kot

(2 točki)

- A najmanjše število bitov, ki sestavlja eno besedo;
- B največje dovoljeno število bitov pri enem prenosu;
- C naslovljiva lokacija v pomnilniku;
- D število bitov v pomnilniškem registru;
- E število bitov v podatkovnem registru.

6. Sklad obravnavamo kot podatkovno strukturo s pristopno metodo LIFO (last in first out). V katerih primerih za uporabo sklada v programu ne potrebujemo posebnih ukazov?

(2 točki)

- A Ob servisiranju prekinitvenih zahtev in klicih podprogramov.
- B Ob nastopu prekinitve.
- C Ob obdelavi podatkov v inverzni smeri.
- D Ob ponovni postavitvi sistema ob mehkem resetiranju.
- E Za uporabo sklada v programu vedno potrebujemo dodatne ukaze.

7. **Vhodno/izhodne naprave lahko na računalniški sistem priključimo na več načinov. Katere so značilnosti priklopa vhodno/izhodnih naprav prek pomnilniško preslikanega vhoda in izhoda?**

(2 točki)

- A Registri krmilnikov V/I naprav so v posebnem naslovnem prostoru, CPE teh lokacij ne vidi, posebni vhodno/izhodni ukazi niso potrebni.
- B Registri krmilnikov V/I naprav so v posebnem naslovnem prostoru, CPE jih vidi kot druge pomnilniške lokacije, potrebni so posebni vhodno/izhodni ukazi.
- C Registri krmilnikov V/I naprav so v pomnilniškem naslovnem prostoru, CPE teh lokacij ne vidi, potrebni so posebni vhodno/izhodni ukazi.
- D Registri krmilnikov V/I naprav so v pomnilniškem naslovnem prostoru, CPE jih vidi kot druge pomnilniške lokacije, posebni vhodno/izhodni ukazi niso potrebni.
- E Nobeden od odgovorov A, B, C in D ne opisuje značilnosti priklopa vhodno/izhodnih naprav prek pomnilniško preslikanega vhoda in izhoda.

8. **Dvojiški komplement števila 4 je v dvojiškem sestavu:**

(2 točki)

- A 11111011
- B -00000100
- C 11111100
- D -00000011
- E 00000110

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

```
var n,x,i,rezultat : word;
begin
  readln(n);
  readln(x);
  rezultat:=0;
  for i:=1 to n do
    rezultat:=rezultat + x;
  writeln(rezultat);
end.
```

C:

```
#include <stdio.h>
void main()
{
  unsigned int n,x,i,rezultat=0;
  scanf("%d",&n);
  scanf("%d",&x);
  for (i=1; i<=n; i++)
    rezultat+=x;
  printf("%d",rezultat);
}
```

(2 točki)

- A Vsoto števil rezultat in x.
- B Produkt števil n in x.
- C Vsoto števil n in x.
- D Produkt števil rezultat in x.
- E Večkratnike števila x.

10. Deklarirano imamo tabelo treh znakov:**Pascal:**

```
var t:array[0..2] of char;
```

C:

```
char t[3];
```

V to tabelo želimo vpisati črke 'A', 'B' in 'C'. Obkrožite, katero zaporedje stavkov nam to zagotovi.

(2 točki)

Pascal:

A `t[1]:='A'; t[2]:='B'; t[3]:='C';`

B `t['A']:=1;t['B']:=2; t['C']:=3;`

C `t1:='A'; t2:='B'; t3:='C';`

D `t[0]:='A'; t[1]:='B'; t[2]:='C';`

E `t:='ABC';`

C:

A `t[1]='A'; t[2]='B'; t[3]='C';`

B `t0='A'; t1='B'; t2='C';`

C `t1='A'; t2='B'; t3='C';`

D `t[0]='A'; t[1]='B'; t[2]='C';`

E `t="ABC";`

11. Želimo napisati podprogram, ki vrne peto potenco števila n. Obkrožite VSE ustrezne glave podprogramov:

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

Pascal:

- A procedure peta_potenca(var n, rezultat : integer);
- B procedure peta_potenca(n, rezultat : integer);
- C function peta_potenca(n : integer) : integer;
- D function peta_potenca(var n : integer; rezultat : integer);
- E procedure peta_potenca(n : integer; rezultat : integer);

C:

- A void peta_potenca(int *n, int *rezultat);
- B void peta_potenca(int n, int rezultat);
- C int peta_potenca(int n);
- D void peta_potenca(int *n, *rezultat);
- E void peta_potenca(int n, rezultat);

12. Programer je napisal stavek:

Pascal:

```
while (true) do begin end;
```

C:

```
while (3) { }
```

Prevajalnik za programski jezik Pascal oziroma C bo:

(2 točki)

- A sporočil, da gre za sintaktično napako;
- B sporočil, da gre za semantično napako;
- C prevedel program brez težav, saj gre za sintaktično napako, ki je prevajalnik ne pozna;
- D sporočil, da je uporaba pomnilnika prekoračena;
- E prevedel program brez javljanja napak.

13. Katera od naslednjih zank se ponovi natanko 3-krat?*(2 točki)***Pascal:**

- A `i:=1;`
`while (i<3) do`
`begin write(i);inc(i);`
`end;`
- B `i:=1;`
`repeat`
`write(i); inc(i);`
`until i=3;`
- C `for i:=7 to 9 do write(i);`
- D `for i:=7 to 10 do write(i);`
- E `for i:=50 downto 47 do write(i);`

C

- A `int i=1;`
`while (i<3) {`
`printf("%d",i);i++;}`
- B `int i=1;`
`do {printf("%d",i);`
`i++;}`
`while (i<3);`
- C `for(i=7;i<=9;i++){printf("%d",i);}`
- D `for(i=7;i<=10;i++){printf("%d",i);}`
- E `for (i=50;i>=47;i--) {printf("%d",i);}`

14. Obkrožite pravilno trditev.*(2 točki)*

- A Tabela je enostaven, osnoven pascalski oziroma C – podatkovni element. Elemente tabele enostavno vpisujemo in izpisujemo.
- B Tabela je sestavljena, homogena podatkovna struktura, ki jo inicializiramo s stavkom
Pascal: `tabela := [3, 5, 12, 8];` **C:** `tabela = [3, 5, 12, 8];`
- C Tabela je sestavljena nehomogena podatkovna struktura. Do elementov tabele dostopamo prek indeksa posameznega elementa.
- D Tabela je homogena podatkovna struktura. Element tabele je lahko poljubnega enostavnega ali predhodno definirane podatkovnega tipa.
- E Tabela je nehomogena sestavljena podatkovna struktura. Zanj je značilno, da njeni podatkovni elementi ohranjajo vse svoje osnovne lastnosti.

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

```
procedure SpremeniX(x:integer);
var i:integer;
begin
  for i:=1 to 10 do x:=x+2*i
end;
begin
  ...
  x:=5; SpremeniX(x);
  ...
end.
```

C:

```
void SpremeniX(int x){
  int i;
  for (i=1; i<=10; i++)
    x+=2*i;
}
void main(){
  ...
  x=5; SpremeniX(x);
  ...
}
```

Kakšno vrednost ima celoštevilski spremenljivka x, ki je definirana v glavnem programu, po izvršitvi gornjih stavkov:

(2 točki)

- A 60
- B 115
- C 5
- D nedefinirano
- E 7

16. Katere vrednosti se izpišejo za X in Y, ko se izvede naslednje zaporedje stavkov:

Pascal:

```
x:=0; y:=10;
repeat
  x:=x+3;
  y:=y-4;
  if (((y-x) mod 2) <> 0) then
    writeln(x, ', ', y)
until y <= 3;
```

C:

```
x=0; y=10;
do {
  x+=3;
  y-=4;
  if (((y-x)%2) != 0) {
    printf("%d, %d\n", x, y); }
while (y>3);
```

(2 točki)

- A 3, 6
- B 6, 2
- C 2, 6
- D 6, 3
- E 5, 2

17. Razvrstite predlagane podatkovne tipe po skupinah. V tabelo vpišite ustrezne oznake za navedene podatkovne tipe.

Oznake skupin:

1. enostavni tipi podatkov
2. homogeni sestavljeni tipi podatkov
3. nehomogeni sestavljeni tipi podatkov

(2 točki)

Pascal:

	Oznaka skupine
integer	
record	
array	
char	
real	
string	

C

	Oznaka skupine
int	
struct	
int []	
char	
float	
char []	

Rešitev:

Pascal	C	Oznaka skupine
integer	int	1.
record	struct	3.
array	int []	2.
char	char	1.
real	float	1.
string	char []	2.

18. Kaj izpiše naslednji program?*(2 točki)***Pascal:**

```
var x,y : char;
  procedure abc(var x1,y1:char);
  var x : char;
    procedure fun;
    var x : char;
    begin
      x:='T';
      inc(x);
      inc(x1);
      writeln(x,x1,y1);
    end;
  begin
    x:=y1;
    writeln(x,x1,y1);
    fun;
    writeln(x,x1,y1);
  end;
begin
  x:='A';
  y:='B';
  abc(x,y);
  writeln(x,y);
end.
```

C:

```
#include <stdio.h>
char x='A',y='B';
void abc (char *x1, char *y1)
{
  char x;
  x=*y1;
  printf("%c%c%c\n",x,*x1,*y1);
  {
    char x='T';
    x++;
    (*x1)++;
    printf("%c%c%c\n",x,*x1,*y1);
  }
  printf("%c%c%c\n",x,*x1,*y1);
}
void main()
{
  abc(&x,&y);
  printf("%c%c\n",x,y);
}
```

Rešitev: vsaj dva odgovora 1 točka

BAB
UBB
BBB
BB

19. Naslednji del programa prepisi z zanko for.*(2 točki)***Pascal:**

```
i:=550;
while (i>=100) do
  begin
    writeln(i);
    i:=i-1;
  end;
```

C:

```
int i=550;
while (i>=100) {
  printf("%d\n",i);
  i--;
}
```

Rešitev:**Pascal:**

```
for i:=550 downto 100 do
  Writeln(i);
```

C:

```
for(i=550;i>=100;i--) {
printf("%d\n",i);}
}
```

20. Deklarirana je spremenljivka Pascal: `var x : string[8], oziroma C: char x[9];`. V spremenljivko `x` je vpisana beseda, ki ima najmanj dve črki. Napišite ukaz oziroma zaporedje ukazov, s katerim izpišete zadnji dve črki besede.

(2 točki)

Rešitev:

Pascal:

```
writeln(copy(x, length(x)-1, 2));
```

C:

```
p=strchr(x, '\\0');  
printf("%s", p-2);
```

21. Napišite program, s katerim preberete celo število. Števke prebranega števila zapišite v tabelo osmih znakov tako, da so desno poravnane. V preostale celice tabele zapišite presledke.

(4 točke)

Primer:

Če je prebrano število 1524, naj bo vsebina tabele:

' '	' '	' '	' '	'1'	'5'	'2'	'4'
-----	-----	-----	-----	-----	-----	-----	-----

Če je prebrano število -10599, naj bo vsebina tabele:

' '	' '	'-'	'1'	'0'	'5'	'9'	'9'
-----	-----	-----	-----	-----	-----	-----	-----

Rešitev:**Pascal:**

```

const t : array [1..8] of char = (' ',' ',' ',' ',' ',' ',' ',' ');
const i : integer = 8;
      znak : char = ' ';
var stevilo : integer;
begin
  readln(stevilo);
  if (stevilo = 0) then
    t[i]:='0'
  else
    begin
      if (stevilo<0) then
        begin
          znak:='-';
          stevilo:=abs(stevilo);
        end;
      while (stevilo<>0) do
        begin
          t[i]:=chr (stevilo mod 10 + ord('0'));
          stevilo:=stevilo div 10;
          dec(i);
        end;
      t[i]:=znak;
    end;
end.

```

C:

```

#include <stdio.h>
void main()
{
  char t[8]={' ',' ',' ',' ',' ',' ',' ',' '};
  int i=7, stevilo;
  char znak=' ';
  scanf("%d",&stevilo);
  if (stevilo==0)
    t[i]='0';
  else {
    if (stevilo<0) {
      znak='-';
      stevilo=-stevilo;
    }
    while (stevilo!=0) {
      t[i]= stevilo % 10 + '0';
      stevilo=stevilo /10;
      i--;
    }
    t[i]=znak;
  }
}

```

22. Deklarirajte zapis, ki vsebuje dve znaka in celo število. Vrednost celega števila izračunamo tako, da seštejemo kodi ASCII znakov. Napišite program, ki napolni dvodimenzijsko tabelo (osem stolpcev in osem vrstic) z vrednostmi, prikazanimi v spodnji tabeli.

(4 točke)

AA130	AB131	AC132	AD133	AE134	AF135	AG136	AH137
BA131	BB132	BC133	BD134	BE135	BF136	BG137	BH138
CA132	CB133	CC134	CD135	CE136	CF137	CG138	CH139
DA133	DB134	DC135	DD136	DE137	DF138	DG139	DH140
EA134	EB135	EC136	ED137	EE138	EF139	EG140	EH141
FA135	FB136	FC137	FD138	FE139	FF140	FG141	FH142
GA136	GB137	GC138	GD139	GE140	GF141	GG142	GH143
HA137	HB138	HC139	HD140	HE141	HF142	HG143	HH144

Rešitev: (2 točki za deklaracije, 2 točki za program)

Pascal:

```
type zapis = record
    z1, z2 : char;
    stev   : integer;
end;
var t : array ['A'..'H', 'A'..'H'] of zapis;
var i, j : char;
begin
    for i:='A' to 'H' do
        for j:='A' to 'H' do begin
            t[i, j].z1:=i;
            t[i, j].z2:=j;
            t[i, j].stev:=ord(i)+ord(j);
        end;
    end;
end.
```

C:

```
struct zapis {
    char zn1, zn2;
    int  stev;
};
zapis t[8][8];
void main()
{
    char i, j;
    for (i=0; i<8; i++)
        for (j=0; j<8; j++){
            t[i][j].zn1='A'+i;
            t[i][j].zn2='A'+j;
            t[i][j].stev='A'+i+'A'+j;
        }
}
```

23. Imamo besedilno datoteko `dohodki.txt`. V vsaki vrstici datoteke so zabeleženi podatki za osebo: priimek, ime, starost in dohodek. Podatek starost je celo število, podatek dohodek je realno število. Podatki so ločeni z vejico. Vzemimo, da je najdaljši priimek dolg 20 znakov in najdaljše ime 10 znakov. Primer vsebine datoteke:

Novak,Andrej,25,120580.00

Jurman,Miha,38,425000.00

Jerman,Grega,22,121980.50

...

Napišite program, ki bo prepisal vsebino datoteke `dohodki.txt` v binarno datoteko `dohodki.dat`. Vsebina ene vrstice datoteke `dohodki.txt` naj ustreza vsebini enega zapisa binarne datoteke `dohodki.dat`.

(4 točke)

Rešitev:

Pascal:

```

type zapis = record
    priimek : string[20];
    ime      : string[10];
    starost  : integer;
    dohodek : real;
end;
bin_datoteka = file of zapis;
var f1 : text;
    f2 : bin_datoteka;
    vrstica : string;
procedure zapisi(var f2 : bin_datoteka; v:string);
var z : zapis;
    err : integer;
begin
    z.priimek:=copy(v,1,pos(',',v)-1);
    delete(v,1,pos(',',v));
    z.ime:=copy(v,1,pos(',',v)-1);
    delete(v,1,pos(',',v));
    val(copy(v,1,pos(',',v)-1),z.starost,err);
    delete(v,1,pos(',',v));
    val(v,z.dohodek,err);
    write(f2,z);
end;
begin
    assign(f1,'meritve.txt');
    reset(f1);
    assign(f2,'meritve.dat');
    rewrite(f2);
    while not eof(f1) do begin
        readln(f1,vrstica);
        zapisi(f2,vrstica);
    end;
    close(f1);
    close(f2);
end.

```

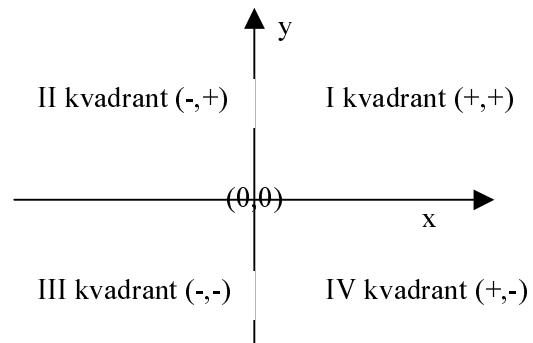

C:

```
#include <stdio.h>
#include <stdlib.h>
struct zapis {
    char priimek[21];
    char ime[11];
    int starost;
    float dohodek;
};
FILE *f1,*f2;
void parsing(char *v, zapis *z)
{
    int i,j;
    char *ss;
    i=0;j=0;
    while (v[i]!='(',')') {
        z->priimek[j]=v[i];
        i++;j++;
    }
    i++;
    z->priimek[j]='\0';
    j=0;
    while (v[i]!='(',')') {
        z->ime[j]=v[i];
        i++;j++;
    }
    i++;
    z->ime[j]='\0';
    j=0;
    while (v[i]!='(',')') {
        ss[j]=v[i];
        i++;j++;
    }
    i++;
    ss[j]='\0';
    z->starost=atoi(ss);
    j=0;
    while (v[i]!='\0') {
        ss[j]=v[i];
        i++;j++;
    }
    ss[j]='\0';
    z->dohodek=atof(ss);
}
void main()
{ zapis z;
  char *vrstica;
  f1 = fopen("meritve.txt","rt");
  f2 = fopen("meritve.dat","wb");
  while (!feof(f1)) {
      fscanf(f1,"%s\n",vrstica);
      parsing(vrstica,&z);
      fwrite(&z,sizeof(z),1,f2);
  };
  fclose(f1);
  fclose(f2);
}
```

24. Podana je naslednja deklaracija točke v ravnini:

Pascal:

```
type tocka = record
    x : integer;
    y : integer;
    barva : integer;
end;
var t : tocka;
```



C:

```
struct tocka {
    int x;
    int y;
    int barva;
} t;
```

Napišite zaporedje ukazov, ki izpišejo številko kvadranta, v katerem je točka t.

(4 točke)

Rešitev:

Pascal:

```
if (t.x >= 0) then
    begin
        if t.y>=0 then
            writeln('1. kvadrant')
        else
            writeln('2. kvadrant');
        end
    else
        begin
            if t.y>=0 then
                writeln('4. kvadrant')
            else
                writeln('3. kvadrant');
            end;
        end;
```

C:

```
if (t.x >= 0) {
    if (t.y>=0)
        printf("1. kvadrant");
    else
        printf("2. kvadrant");
}
else {
    if (t.y>=0)
        printf("4. kvadrant");
    else
        printf("3. kvadrant");
};
```

IZPITNA POLA 2

Obkrožite pravilno trditev.

1. Katera od plasti ISO/OSI referenčnega modela skrbi za vzpostavitev, vzdrževanje in prekinitev komuniciranja na ravni končnih računalnikov?

(2 točki)

- A Aplikacijska plast.
- B Predstavitvena plast.
- C Plast seje.
- D Transportna plast.
- E Omrežna plast.
- F Povezovalna plast.
- G Fizična plast.

2. 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 najmanj zanesljive povezave.

3. Kaj je značilno za omrežja, ki za prenos podatkov uporabljajo preklapljanje povezav?

(2 točki)

- A Paketi, ki sestavljajo neko sporočilo, potujejo od izvora do ponora po različnih poteh in različno hitro.
- B Sporočilo potuje tako, da se v celoti sestavi v vsakem vozlišču na poti od izvora do ponora.
- C Za prenos podatkov se med izvorom in ponorom vzpostavi logični kanal za čas prenosa celotnega sporočila.
- D Paketi, iz katerih je sestavljeno sporočilo, potujejo po najkrajši možni poti, ne glede na zasedenost omrežja.
- E Nobeden od odgovorov A, B, C in D ne opisuje značilnosti omrežja, ki za prenos podatkov uporablja preklapljanje povezav.

4. Kaj omogoča storitev telnet?*(2 točki)*

- A Enostaven prenos datotek med računalnikoma.
- B Prijavo na oddaljeni računalnik.
- C Prenos datotek na lokani računalnik.
- D Pošiljanje sporočil vsem uporabnikom omrežja.
- E Sprejem elektronske pošte.

5. Organizacija datoteke opredeljuje:*(2 točki)*

- A način shranjevanja datotek na zunanjem pomnilniku;
- B način urejanja in shranjevanja zapisov v datoteki;
- C način zaščite podatkov v datoteki;
- D organizacijo datotek na zunanjem pomnilniku;
- E postopke za izdelavo arhivskih kopij datotek (back-up).

6. Entiteta Oseba je opisana z atributi: davcna_stevilka, priimek, ime, spol, stopnja_izobrazbe.**Kateri atribut je primeren za primarni indeks tabele:***(2 točki)*

- A davcna_stevilka.
- B priimek,
- C ime,
- D spol,
- E stopnja_izobrazbe.

7. Sekundarne indekse ustvarimo predvsem za podatke, za katere je značilna pogosta izvedba operacije:*(2 točki)*

- A dodajanja,
- B brisanja,
- C spreminjanja,
- D iskanja.
- E vseh navedenih operacij.

8. Na kontekstni ravni diagrama toka podatkov se pojavljajo:*(2 točki)*

- A samo glavni proces in zunanje entitete;
- B samo glavni proces in glavna zbirka podatkov;
- C glavni proces, podatkovni tokovi in zunanje entitete;
- D glavni proces, podatkovni tokovi in zbirke podatkov;
- E glavni proces, podatkovni tokovi, zbirke podatkov in zunanje entitete.

9. Kateri so osnovni gradniki modela E-R?*(2 točki)*

- A Entitete, objekti, zapisi.
- B Atributi, relacije, entitete.
- C Zapisi, atributi, objekti.
- D Relacije, tabele, grafi.
- E Tabele, objekti, entitete.

10. Kaj je podatkovna baza?*(2 točki)*

- A Model okolja, ki je osnova za sprejemanje odločitev.
- B Sistem za sprejemanje odločitev.
- C Zbirka odločitev.
- D Zbirka informacij in funkcij.
- E Odločitveni sistem.

11. Naštejte vsaj 3 funkcije orodij za računalniško podprto inženirstvo (CASE):*(2 točki)***Odgovor:**

risanje diagramov (DTP, ERD)

preverjanje specifikacij

prevajanje modela ER v relacijski model

generiranje dokumentacije

vizualno programiranje – generiranje kode

12. Kaj sestavlja opis datoteke v datotečnem seznamu? Naštejte vsaj tri.

(2 točki)

Odgovor:

(2 odgovora 1 točka, 3 odgovori ali več 2 točki)

Ime datoteke

Datotečni atributi

Lastnik datoteke

Datum in čas oblikovanja oziroma zadnje spremembe datoteke

Številka logično prve dodelitvene enote

13. Katere gradnike diagramov toka podatkov (DTP) opišemo v podatkovnem slovarju?

(2 točki)

Odgovor (1 odgovor 1 točka):

Tokove podatkov in zbirke.

14. Poimenujte naslednje operacije relacijske algebre:

(2 pravilna odgovora – 1 točka
3, 4 pravilni odgovori – 2 točki)

$\sigma[F]r$ –

$r \cap s$ –

$r \cup s$ –

$\pi[F]r$ –

Odgovor:

$\sigma[F]r$ – selekcija

$r \cap s$ – presek

$r \cup s$ – unija

$\pi[F]r$ – projekcija

15. Dani sta tabeli:

Student			
Sifra	ime	priimek	letnik
100	Jure	Novak	2
200	Jan	Novak	1
300	Franci	Lump	1
400	Micka	Žolna	1
500	Lea	Kralj	1

Izpit			
Sifra	predmet	datum	ocena
100	MAT	13.1.02	9
100	FIZ	20.2.02	9
300	MAT	20.1.03	7
500	MAT	20.1.03	10
500	FIZ	18.1.03	10

Napišite stavek SQL, ki izpiše priimke študentov prvega letnika.

(1 točka)

Rešitev:

```
SELECT priimek
```

```
FROM Student
```

```
WHERE letnik=1
```

Napišite ustavek SQL, ki izpiše povprečno oceno po predmetih.

(3 točke)

Rešitev:

```
SELECT predmet,AVG(ocena)
```

```
FROM izpit
```

```
GROUP BY predmet
```

16. Na diagramih toka podatkov imamo atomarni proces **Odobren_popust**, za katerega izdelajte minispecifikacije z odločitveno tabelo ali odločitvenim drevesom. Kot vhod v proces dobimo **Plačilo_z_gotovino** in podatek o ceni izdelka, ki ga želimo kupiti. Kot izhod dobimo podatkovni tok **Količina_popusta**. Upoštevajte, da je popust odobren, če plačamo z gotovino. Popust izdelka, ki je dražji od 10.000 SIT, je 5 %, popust izdelka, ki je dražji od 50.000 SIT, pa je 10 %. Če stane izdelek več kakor 100.000 SIT, je popust 20 %.

(4 točke)

Rešitev:

Pogoji akcije/ukrepi	Pravila					
<i>Cena_izdelka</i>	>=10000	>=10000	>50000	>50000	>=100000	>=100000
<i>Plačilo_z_gotovino</i>	D	N	D	N	D	N
<i>Količina_popusta</i>	5%	0%	10%	0%	20%	0%

