



Šifra kandidata:

Državni izpitni center



M 0 5 1 7 8 1 1 1

SPOMLADANSKI ROK

RAČUNALNIŠTVO

Izpitna pola 1

Četrtek, 16. junij 2005 / 110 minut

Dovoljeno dodatno gradivo in pripomočki: kandidat prinese s seboj nalivno pero ali kemični svinčnik. Kandidat dobi dva ocenjevalna obrazca in dva konceptna lista.

SPLOŠNA MATURA

NAVODILA KANDIDATU

Pazljivo preberite ta navodila. Ne obračajte strani in ne rešujte nalog, dokler vam nadzorni učitelj tega ne dovoli.

Prilepite kodo oziroma vpišite svojo šifro (v okvirček desno zgoraj na tej strani in na obrazcu za ocenjevanje).

Rešitve vpisujte v za to predvideni prostor v izpitni poli, z nalivnim peresom ali kemičnim svinčnikom. Odgovori, pisani z navadnim svinčnikom, bodo ocenjeni z nič (0) točkami. V izpitno polo vpisujte le končne rešitve algoritmov. Skice lahko rišete prostoročno. Pišite urejeno in čitljivo.

Število točk za posamezna vprašanja je navedeno ob nalogah v izpitni poli.

Zaupajte vase in v svoje sposobnosti.

Želimo vam veliko uspeha.

Ta pola ima 20 strani.

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 postavitev 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 ($2 \cdot 512$ 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 izvorni kodi programa 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 postavitevi 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. Katerе so značilnosti priklopa vhodno/izhodnih naprav prek pomnilniško preslikanega vhoda in izhoda?

(2 točki)

- A Registri krmilnikov V/I naprav so v posebnem naslovнем prostoru, CPE teh lokacij ne vidi, posebni vhodno/izhodni ukazi niso potrebni.
- B Registri krmilnikov V/I naprav so v posebnem naslovнем prostoru, CPE jih vidi kot druge pomnilniške lokacije, potrebni so posebni vhodno/izhodni ukazi.
- C Registri krmilnikov V/I naprav so v pomnilniškem naslovнем prostoru, CPE teh lokacij ne vidi, potrebni so posebni vhodno/izhodni ukazi.
- D Registri krmilnikov V/I naprav so v pomnilniškem naslovнем 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 8-bitnega števila z vrednostjo 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 to zagotovi.

*(2 točki)***Pascal:**

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

(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 programske 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 definiranega podatkovnega tipa.
- E Tabela je nehomogena sestavljena podatkovna struktura. Zanjo 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 celoštevilska 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 []	

18. Kaj izpiše naslednji program?

(vsaj dva pravilna odgovora 1 točka)
(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:

19. Naslednji del programa prepiši 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:

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:

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 zapisite 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:

22. Deklarirajte zapis, ki vsebuje dva znaka in celo število. Vrednost celega števila izračunamo tako, da seštejemo ASCII kodi znakov. Napišite program, ki napolni dvodimensijsko 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:

23. Imamo tekstovno 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

3

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:

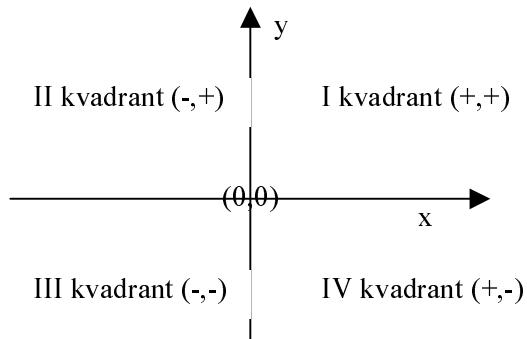
24. Podana je naslednja deklaracija točke v ravni:

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: