



Šifra kandidata:

Državni izpitni center



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 obrazca 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 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 ( $2 \cdot 512$  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 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 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']:=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 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 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 celoštevilska spremenljivka x, ki je definirana v glavnem programu, po izvrstitvi 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 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:**

---

---

---

---

---

---

---

---

---

---















