



Š i f r a k a n d i d a t a :

Državni izpitni center



JESENSKI ROK

RAČUNALNIŠTVO

Izpitna pola 1

Petek, 31. avgust 2007 / 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 izpuščajte ničesar.

Ne obračajte strani in ne začenjajte reševati nalog, dokler vam nadzorni učitelj tega ne dovoli.

Prilepite kodo oziroma vpišite svojo šifro (v okvirček desno zgoraj na tej strani in na ocenjevalna obrazca).

Rešitve vpisujte v za to predvideni prostor v izpitni poli z nalivnim peresom ali kemičnim svinčnikom.

V izpitno polo vpisujte le končne rešitve algoritmov. Konceptna lista uporabite za zapisovanje osnutkov rešitev. Zapisi na konceptnih listih se pri ocenjevanju ne upoštevajo.

Skice lahko rišete prostoročno. Pišite urejeno in čitljivo.

Zaupajte vase in v svoje sposobnosti.

Želimo vam veliko uspeha.

Ta pola ima 20 strani, od tega 1 prazno.

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

Obkrožite pravilno trditev.

1. Osnovni gradniki računalniškega sistema so (obkrožite najpopolnejši odgovor):

(2 točki)

- A RAM, ROM, CPE;
- B CPE, pomnilnik, disketna enota;
- C monitor, tipkovnica, CPE;
- D CPE, pomnilnik, vhodno-izhodne enote;
- E monitor, ROM, CPE.

2. Pomnilniška hierarhija pomeni določeno organizacijo in medsebojno povezavo pomnilnikov znotraj računalnika. Pomnilniki so v naslednjem vrstnem redu, če gledamo naraščajočo velikost in padajočo hitrost dostopa procesorja:

(2 točki)

- A disk, registri, predpomnilnik, primarni pomnilnik;
- B registri, predpomnilnik, primarni pomnilnik, disk;
- C predpomnilnik, registri, disk, primarni pomnilnik;
- D primarni pomnilnik, registri, predpomnilnik, disk;
- E disk, predpomnilnik, registri, primarni pomnilnik.

3. Zakaj je potreben mehanizem prekinitev s prioritetami?

(2 točki)

- A Zaradi procesov, ki zahtevajo takojšen začetek procesiranja.
- B Zato, da lahko procese razvrstimo skladno z zahtevanim časom začetka procesiranja.
- C Samo mehanizem prekinitev zagotavlja začetek in končanje procesiranja v odvisnosti od prioritete procesov.
- D Samo mehanizem prekinitev zagotavlja končanje procesiranja v odvisnosti od prioritete procesov.
- E Samo ta mehanizem zagotavlja, da dobijo vsi procesi ustrezen čas procesorja.

4. Dolžina naslovnih registrov procesorja določa:*(2 točki)*

- A največjo točnost (precizijo) računanja;
- B širino podatkovnega vodila med predpomnilnikom in glavnim pomnilnikom;
- C število možnih prekinitev;
- D dolžino besede;
- E obseg naslovnega prostora.

5. Za registrsko arhitekturo procesorjev RISC je značilno, da:*(2 točki)*

- A program sploh ne uporablja pomnilnika za shranjevanje spremenljivk;
- B prevajalniki lažje prevedejo program, ki ne uporablja pomnilniških naslovov;
- C je arhitektura tako poenostavljena, da procesor v ukazih dela le s kratkimi operandi;
- D pozna le vsebovano naslavljanje;
- E se vse aritmetične in logične operacije izvajajo nad vsebinami registrov.

6. Preliv (overflow) pri računanju razlikujemo od prenosa (carry), ker:*(2 točki)*

- A se prenos pojavi pri seštevanju, preliv pa pri odštevanju;
- B se prenos pojavi pri računanju s 16-bitnimi števili, preliv pa z 8-bitnimi;
- C se prenos uporablja pri računanju s celimi števili, preliv pa pri računanju z realnimi števili;
- D prenos pomeni, da smo presegli maksimalno vrednost, preliv pa minimalno;
- E preliv upoštevamo le pri računanju s predznačenimi števili.

7. Preslikava logičnih naslovov v fizične je pri uporabi navideznega pomnilnika potrebna, da:*(2 točki)*

- A se neki del programa lahko naloži na različne naslove v fizičnem pomnilniku;
- B se lahko vsak ukaz ali spremenljivka poljubno premešča po fizičnem pomnilniku;
- C se lahko program (software) poveže s strojno opremo (hardware) na fizičnem nivoju;
- D ni fragmentacije pomnilnika;
- E so deli programa, ki so med seboj povezani, v istem naslovnem področju.

8. Pri katerem izmed danih primerov vsebin je pravilno izračunana prečna in vzdolžna parnost (pariteta), če izberemo liho parnost (pariteto)?

(2 točki)

A

1	1	0	0	1	0	0	1	1
1	0	0	0	1	1	1	1	0
0	0	0	1	1	0	1	0	0
1	1	1	1	1	0	1	1	0
0	1	0	1	1	0	0	0	0

B

1	1	0	0	1	0	0	1	1
1	0	0	0	1	1	1	1	0
0	0	0	1	1	0	1	0	0
1	1	1	1	1	0	1	1	0
1	0	1	0	0	1	1	1	1

C

1	1	0	0	1	0	0	1	0
1	0	0	0	1	1	1	1	1
0	0	0	1	1	0	1	0	1
1	1	1	1	1	0	1	1	1
0	1	0	1	1	0	0	0	1

D

1	1	0	0	1	0	0	1	1
1	0	0	0	1	1	1	1	1
0	0	0	1	1	0	1	0	1
1	1	1	1	1	0	1	1	0
0	0	0	1	0	1	0	1	0

E

1	1	0	0	1	0	0	1	0
1	0	0	0	1	1	1	1	1
0	0	0	1	1	0	1	0	1
1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
NALOGE IZBIRNEGA TIPA

Obkrožite pravilno trditev.

9. Kaj se izpiše med izvajanjem naslednjega zaporedja stavkov:

Pascal:

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

C:

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

(2 točki)

- A Produkt števil rezultat in x.
- B n-ta potenca števila x.
- C Vsota števil n in x.
- D Produkt števil rezultat in i.
- E Vsi večkratniki števila x.

10. V programskem jeziku Pascal oziroma C sta dani deklaraciji spremenljivk:**Pascal:**

```
var tabela=array[1..10] of integer;
```

C:

```
int tabela[10];
```

Katera trditev je pravilna?*(2 točki)*

- A Za deklaracijo spremenljivke `tabela` je uporabljen enostavni podatkovni tip.
- B Za deklaracijo spremenljivke `tabela` je uporabljen sestavljeni homogeni podatkovni tip.
- C Za deklaracijo spremenljivke `tabela` je uporabljen sestavljeni nehomogeni podatkovni tip.
- D Za deklaracijo spremenljivke `tabela` je uporabljen dinamični podatkovni tip.
- E Navedena deklaracija je neveljavna oziroma nepravilna.

11. Kaj se izpiše po izvedbi naslednjega zaporedja stavkov:**Pascal:**

```
var cena:integer;
procedure vnesi_ceno(var nova_cena:integer);
begin
    nova_cena:=nova_cena+20;
end;
begin
    cena:=510;
    vnesi_ceno(cena);
    writeln('Cena je: ',cena,' SIT');
end.
```

C:

```
#include <stdio.h>
void vnesi_ceno(int *nova_cena);
int main(){
    int cena=510;
    vnesi_ceno(&cena);
    printf("Cena je: %d SIT \n",cena);
    return 0;
}

void vnesi_ceno(int *nova_cena){
    *nova_cena+=20;
}
```

(2 točki)

- A Cena je: 510 SIT
- B Cena je: 520 SIT
- C Cena je: 530 SIT
- D Nič od naštetega.
- E Cena je: 480 SIT

12. Deklarirane so celoštevilске spremenljivke x , y in z . Dan je tudi izsek iz programa

Pascal:

```
y:=37; z:=4;  
x:= y mod z;
```

C:

```
y=37; z=4;  
x= y % z;
```

Kakšna je vrednost spremenljivke x po izvedbi teh programskih stavkov?

(2 točki)

- A 374
- B 9
- C 33
- D 4
- E 1

13. Primerjalnega operatorja $=$ v programskem jeziku Pascal oziroma $==$ v programskem jeziku C ne moremo uporabljati za primerjavo vsebin dveh spremenljivk, če sta spremenljivki deklarirani:

(2 točki)

- A kot predznačeni celi števili;
- B kot nepredznačeni celi števili;
- C kot tabeli;
- D kot znaka;
- E primerjalni operator za ugotavljanje enakosti spremenljivk lahko uporabljamo v vseh zgoraj navedenih primerih.

14. Deklaracija glave funkcije v programskem jeziku Pascal oziroma prototip funkcije v programskem jeziku C vsebuje (obkrožite vse pravilne odgovore):

(2 točki)

(vsi pravilni odgovori 2 točki, dva pravilna odgovora ena točka, že en nepravilen odgovor nič točk)

- A ime funkcije;
- B podatkovni tip vrednosti, ki jo funkcija vrne;
- C pomnilniški naslov, na katerem je funkcija;
- D argumente funkcije;
- E blok stavkov, ki opredeljuje funkcionalnost podprograma.

15. Stavki `case` v programskem jeziku Pascal oziroma stavki `switch` v programskem jeziku C lahko zamenja:

(2 točki)

- A zaporedje prireditev;
- B zaporedje stavkov `for`;
- C zaporedje stavkov `while`;
- D zaporedje stavkov `if`;
- E zaporedje vhodno-izhodnih stavkov.

16. Izvorna koda programa, napisanega v višjem programskem jeziku, se shrani:

(2 točki)

- A kot binarna datoteka;
- B v formatu, ki je določen s programskim jezikom;
- C v formatu, ki se določi pri shranjevanju datoteke;
- D v neznanem formatu;
- E kot tekstovna datoteka ASCII.

SKLOP PROGRAMSKI JEZIKI IN PROGRAMIRANJE
NALOGE S KRATKIMI ODGOVORI**17. Kaj izpiše naslednji program?***(2 točki)***Pascal:**

```
Var x,y : Integer;
Begin
  x:= 140; y:=50;
  While (x<>y) Do
  Begin
    If (x>y) Then
      x:=x-y
    Else
      y:=y-x;
    WriteLn(x, '|', y);
  End;
  WriteLn(x);
End.
```

C:

```
#include "stdafx.h"
#include <stdio.h>
int _tmain(int argc, _TCHAR* argv[])
{
  int x=140, y=50;
  while (x!=y) {
    if (x>y)
      x=x-y;
    else
      y=y-x;
    printf("%d| %d\n", x, y);
  }
  printf("%d\n", x);
  return 0;
}
```

Rešitev:

18. Napišite sled izvajanja funkcije za razvrščanje podatkov tabele x , če je uporabljen algoritem za razvrščanje z izbiranjem.

(2 točki)

Tabela x :

14	21	26	3	11	12	15
----	----	----	---	----	----	----

Rešitev:

19. Napišite program, ki izpiše vsa naravna števila iz intervala $[1...1000]$, ki niso deljiva s 14.

(2 točki)

Rešitev:

20. Kaj vrne naslednja funkcija, če kot argument x navedemo:a) $x='1234321'$;b) $x='1234567'$.

(2 točki)

Pascal:

```
function pali(x:string):integer;
var i : integer;
    pogoj: boolean;
begin
    i:=1;
    pogoj:=true;
    while (i <= length(x) div 2) and pogoj do
    begin
        if x[i]<>x[length(x)-i+1] then
            pogoj:=false
        else
            inc(i);
    end;
    pali:= integer(pogoj);
end;
```

C:

```
#include <string.h>
int pali(char *x)
{
    for (int i=0;i<(strlen(x)/2);)
    { if (x[i]!=x[strlen(x)-i-1])
        break;
      else i++;
    }
    return(i==strlen(x)/2);
}
```

Rešitev:

PRAZNA STRAN