

Principi programiranja. Ogladni primjeri pitanja za Prvi kolokvijum (20 poena). Od naslova 1. Pojam registarskog prenosa i pojam mikro-operacije do naslova 23. 8086 – flagovi. Doći će četiri pitanja. Prvo i drugo pitanje – teorija, treće – program za osnovni računar i četvrto – program na Paskalu.

1. Registri MAR i MBR (uopšte), operacije čitanja i upisivanja u memoriju.
2. Uopšte o vrstama mikro-operacija (četiri grupe).
3. Nabrojati i opisati glavne registre osnovnog računara (PC, ..., AC).
4. O podjeli naredbi osnovnog računara na tri vrste.
5. Šta je to obično (neposredno) adresiranje a šta je posredno (indirektno).
6. Vremenski ciklusi osnovnog računara.
7. Šema kontrolne jedinice osnovnog računara i objasniti.
8. Mikro-operacije prijemnog ciklusa (osnovni računar).
9. Mikro-operacije indirektnog ciklusa (osnovni računar).
10. Mikro-operacije izvršnog ciklusa u slučaju naredbe ADD (osnovni računar).
11. Nabrojati i definisati naredbe koje se odnose na memoriju (osnovni računar).
12. Nabrojati i opisati naredbe koje se odnose na registre (osnovni računar).
13. Registri za ulaz–izlaz i prekide (osnovni računar).
14. Ulazno–izlazno prenošenje može da bude programski upravljano ili prekidom vođeno (osnovni računar).
15. Ulazno–izlazne naredbe (osnovni računar).
16. Mikro-operacije prekidnog ciklusa (osnovni računar).
17. Primjeri ulaznog odnosno izlaznog prenošenja koje je programski kontrolisano (osnovni računar).
18. Rutina za obradu prekida R i dva bafera u memoriji (osnovni računar).
19. Primjer izlaznog prenošenja koje je prekidom vođeno (osnovni računar).
20. Novi izlazni uređaj: linijski štampač (osnovni računar).
21. Apsolutni loader L (osnovni računar).
22. Loader koji vrši relokaciju (osnovni računar).
23. Jezik assemblera za osnovni računar.
24. Računarski softver odnosno tri nivoa programa za računar (uopšte).
25. Realizacija sabiranja i množenja u slučaju nepokretnog odnosno pokretnog zarez (uopšte).
26. Organizacija ulazno–izlazne jedinice (uopšte).
27. Organizacija unutrašnje i spoljašnje memorije (uopšte).
28. Skup registara procesora Intel 8080 uključujući i registar F – flagovi.
29. O pinovima procesora Intel 8080.
30. Glavni registri procesora Intel 8086 (AH, ..., IP), obrazovanje fizičke adrese po formuli $a = 16s + d$.
31. Registar PSW procesora Intel 8086 tj. flagovi.
32. Sastaviti program za osnovni računar za računanje vrijednosti izraza $y = 2a + 4b + 4$.
33. Sastaviti program za osnovni računar za računanje vrijednosti izraza $y = (a/16 - b)/16$. Uputstvo: pomoću CMA i INC ostvaruje se $AC \leftarrow -AC$.
34. Koliko je neparnih brojeva među brojevima a , b i c . Sastaviti program za osnovni računar koji će promjenljivoj y dodijeliti odgovarajuću vrijednost. Uputstvo: parnost broja vidi se po posljednjem bitu.
35. Sastaviti program za osnovni računar za računanje vrijednosti izraza

$$y = \begin{cases} 2a + 2b, & \text{ako je } c < 0 \\ 8a + 8b, & \text{ako je } c \geq 0 \end{cases}$$

36. Sastaviti program za osnovni računar za računanje vrijednosti izraza $y = \sum_{k=a}^b k$ za date a i b .

37. Izračunati zbir brojeva koji se nalaze na adresama n , $n + 1$ i $n + 2$. Veličina n nalazi se na adresi 200 (program za osnovni računar).

38. Izračunati aritmetičku sredinu brojeva a , b , c i d . Uputstvo: pomoću CIR i CIR ostvaruje se $AC \leftarrow AC/4$, u smislu $AC \leftarrow [AC/4]$ (program za osnovni računar).

39. Ostatak pri dijeljenju $a : 16$. Uputstvo: znamo da je $a \bmod 16 = a - (a/16) \cdot 16$, u smislu $a \bmod 16 = a - [a/16] \cdot 16$. Program za osnovni računar.

40. Neka y pokazuje koliko ima nenegativnih brojeva na adresama od 200 do 209. Brojevi se već nalaze upisani na tim adresama. Treba izračunati y . Uputstvo: predznak broja vidi se po njegovom prvom bitu. Ili koristiti neki skip (naredbe za skokove). Program za osnovni računar.

41. Treba izračunati $y \leftarrow 2c(m) + c(n)$, gdje se brojevi m i n nalaze na adresama 200 odnosno 201. Napisati program za osnovni računar.

42. Napisati program na Paskalu za računanje i štampanje najvećeg zajedničkog djelioca (NZD) prirodnih brojeva a i b koji se učitavaju.

43. Program na Paskalu koji štampa prvih 30 članova niza $p_n = \frac{\lambda^n}{n!} e^{-\lambda}$. Broj λ je ulazni podatak.

44. Program na Paskalu za računanje $\binom{n}{k} = \frac{n \cdot (n-1) \cdot \dots \cdot (n-k+1)}{k!}$. Učitavaju se n i k .

45. Učitava se broj n ($1 \leq n \leq 1000$), a računa se i štampa zbir njegovih cifara. Sastaviti odgovarajući program na Paskalu. Na primjer, ako je $n = 349$ onda je zbir = 16.

46. Štampati n -ti Fibonaccijev broj f_n . Broj n se učitava. Sastaviti program na Paskalu. Znamo da je $f_1 = 1$, $f_2 = 1$ i $f_n = f_{n-2} + f_{n-1}$ za $n \geq 3$.

47. Štampati vrijednost zbira $S_n = 1 - \frac{1}{2} + \frac{1}{3} - \dots + \frac{(-1)^{n+1}}{n}$. Broj n je ulazni podatak. Program na Paskalu.

48. Sastaviti program na Paskalu u kome se učitava 100 brojeva, a određuje se (i štampa) najmanji među tim brojevima, kao i najveći.

49. Učitava se 100 karaktera, svaki je veliko slovo (A ili B ili ... ili Z). Koliko različitih slova ima ukupno, među učitanim karakterima? Program na Paskalu.

50. Učitava se broj n ($0 \leq n \leq 10000$), a štampa se njegov binarni oblik. Program na Paskalu. Na primjer, ako je $n = 24$ onda program treba da odštampa 11000.

51. Učitava se broj n ($-10000 \leq n \leq 10000$), a treba štampati njegov 16-bitni prikaz u obliku PK (potpuni komplement). Program na Paskalu. Na primjer, ako je $n = 24$ onda treba da bude odštampano 0000 0000 0001 1000.

52. Učitava se broj n ($1 \leq n \leq 366$) čiji je smisao: n -ti po redu dan u 2016. godini. Program na Paskalu treba da odštampa koji je to datum, u kom mjesecu, kao i koji je to dan u sedmici.

53. Učitava se prirodan broj. Program na Paskalu treba da saopšti da li je taj broj prost ili je pak (suprotno) složen.

54. Koliko ima parova blizanaca (parova prostih brojeva koji se razlikuju za dva) do 1000? Napisati program na Paskalu.

55. Učitava se paran broj n . Neka program na Paskalu odgovori sa "da" ili "ne" na pitanje: da li n može da se prikaže kao zbir dva prosta broja.

Znamo da Goldbachova hipoteza glasi: svaki paran broj veći od dva može da se prikaže kao zbir dva prosta broja.

(drugi komplement = prvi komplement + 1)

34. y = koliko je neparnih među a , b i c .

a_1 = najjači bit akumulatora (bit najveće težine), a_{16} – najslabiji bit akumulatora (bit najmanje težine). Znamo da se prilikom CIR dešava (između ostalog) $a_1 \leftarrow e$ i $e \leftarrow a_{16}$. Sadržaj od E dolazi na najjače mjesto u akumulatoru tj. stari sadržaj od E postaje novi najjači bit akumulatora. Stari najslabiji bit akumulatora postaje novi sadržaj flipflopa E. Znamo da se prilikom CIL dešava (između ostalog) $e \leftarrow a_1$ i $a_{16} \leftarrow e$. Znači e novo $\leftarrow a_1$ staro i a_{16} novo $\leftarrow e$ staro. {CIR, a_i novo $\leftarrow a_{i-1}$ staro}

```

200 LDA 221 AC ← a
201 CIR    e ← a16
202 CLA    AC ← 0
203 CIL    AC ← zadnji bit broja a
204 STA 224 ostavi na 224
205 LDA 222 AC ← b
206 CIR    e ← a16
207 CLA    AC ← 0
208 CIL    AC ← zadnji bit broja b
209 STA 225 ostavi na 225
210 LDA 223 AC ← c
211 CIR    e ← a16
212 CLA    AC ← 0
213 CIL    AC ← zadnji bit broja c
214 ADD 224 AC ← AC + c(224)
215 ADD 225 AC ← AC + c(225)
216 STA 220 y = c(220) ← AC
217 HLT    stop
220 DEC 0  mjesto za y (za rezultat)
221 DEC ... a
222 DEC ... b
223 DEC ... c

```

Pretpostavlja se da su ulazni podaci upisani (već upisani), u trenutku kada počinje izvršavanje/rad programa. Ulazni podaci su a , b i c .

Vrijednost rezultata (y) nalaziće se na predviđenom mjestu, nakon HLT.

Adrese 224 i 225 služe za među-rezultate.

$$35. y = \begin{cases} 2a + 2b, & c < 0 \\ 8a + 8b, & c \geq 0 \end{cases}$$

```

200 LDA 221 AC ← a
201 ADD 222 AC ← AC + b = a + b
202 CIL    AC ← 2 · AC = 2a + 2b

```

```

203 STA 223 y1 ← AC = 2a + 2b
204 CIL    AC ← 2 · AC = 4a + 4b
205 CIL    AC ← 2 · AC = 8a + 8b
206 STA 224 y2 ← AC = 8a + 8b
207 LDA 220 AC ← c
208 SPA    skip if AC ≥ 0
209 BUN 213 goto 213
210 LDA 224 AC ← y2
211 STA 219 y ← AC
212 HLT    stop
213 LDA 223 AC ← y1
214 STA 219 y ← AC
215 HLT    stop
219 DEC 0  mjesto za y (za rezultat)
220 DEC ... c
221 DEC ... a
222 DEC ... b

```

Vidimo da se y_1 drži na adresi 223, vidimo da se y_2 čuva na 224, $c(223) = y_1$, $c(224) = y_2$.

Obratiti pažnju na naredbu SPA (ona se nalazi na adresi 208). U slučaju da nije $AC \geq 0$ za njom će se izvršiti naredba na adresi 209, a u slučaju da je $AC \geq 0$ za njom će se izvršiti naredba 210.

$$36. y = \sum_{k=a}^b k.$$

Plan:

$y \leftarrow 0$, $x \leftarrow a$, $k \leftarrow -b - 1 + a$,
208: $y \leftarrow y + x$, $x \leftarrow x + 1$, $k \leftarrow k + 1$,
if $k = 0$ stop, if $k \neq 0$ 208

```

300 DEC ... a
301 DEC ... b
302 DEC 0  y (rezultat)
303 DEC 0  x
304 DEC 0  k
→ 200 LDA 300 AC ← a
201 STA 303 x ← AC = a
202 LDA 301 AC ← b
203 INC    AC ← AC + 1 = b + 1
204 CMA    complement
205 INC    increment, sada je AC = -b - 1
206 ADD 300 AC ← AC + a = -b - 1 + a
207 STA 304 k ← AC = -b - 1 + a
208 LDA 302 AC ← y
209 ADD 303 AC ← AC + x
210 STA 302 y ← AC
211 LDA 303 AC ← x
212 INC    AC ← AC + 1
213 STA 303 x ← AC
214 LDA 304 AC ← k
215 INC    AC ← AC + 1

```

```

216 STA 304  k ← AC
217 SZA      skip if AC = 0
( ako je k ≠ 0 onda idi na 218 )
( ako je k = 0 onda idi na 219 )
218 BUN 208  goto 208
219 HLT      stop

```

37. $y = c(n) + c(n + 1) + c(n + 2)$, gdje je $n = c(200)$.

```

→ 100 LDA 200  AC ← n
    101 INC     AC ← AC + 1
    102 STA 201  n1 ← AC = n + 1
    103 INC     AC ← AC + 1
    104 STA 202  n2 ← AC = n + 2
    105 LDA* 200 AC ← c(n)
    106 ADD* 201 AC ← AC + c(n1)
    107 ADD* 202 AC ← AC + c(n2)
    108 STA 199  y ← AC
    109 HLT     stop
    199 DEC 0   y (rezultat), zbir
    200 DEC ... n (ulazni podatak)

```

Vidimo da je $c(201) = n_1$ i $c(202) = n_2$, to su dvije pomoćne promjenljive, $c(200) = n$.

Podrazumijeva se:

```

n      DEC ... prvi sabirak x1
n + 1  DEC ... drugi sabirak x2
n + 2  DEC ... treći sabirak x3

```

Radi boljeg razumijevanja, pogledajmo konkretan slučaj (primjer). Uzmimo da je $n = 300$. Tada program računa $y = c(300) + c(301) + c(302)$, odnosno podrazumijeva se:

```

300 DEC ... prvi sabirak
301 DEC ... drugi sabirak
302 DEC ... treći sabirak

```

Ako želimo da sve bude konkretno, ako je:

```

300 DEC 4100 prvi sabirak
301 DEC 7100 drugi sabirak
302 DEC 9100 treći sabirak

```

onda će na kraju rada programa biti $c(199) = 20300$.

Rješenje zadatka na drugi način:

```

→ 100 LDA* 200 AC ← c(c(200)) = x1
    101 STA 199  c(199) ← AC
    102 LDA 200  AC ← c(200)
    103 INC     AC ← AC + 1

```

```

104 STA 200  c(200) ← AC
105 LDA 199  AC ← c(199)
106 ADD* 200 AC ← AC + c(c(200))
           = AC + x2
107 STA 199  c(199) ← AC
108 LDA 200  AC ← c(200)
109 INC     AC ← AC + 1
110 STA 200  c(200) ← AC
111 LDA 199  AC ← c(199)
112 ADD* 200 AC ← AC + c(c(200))
           = AC + x3
113 STA 199  c(199) ← AC
114 HLT     stop
199 DEC 0   y (rezultat)
200 DEC ... n (ulazni podatak)

```

38. $y = \frac{1}{4}(a + b + c + d)$.

Uputstvo: Pomoću naredbe CIR postiže se $AC \leftarrow [AC/2]$ (cio dio).

39. $y = a \bmod 16$.

Uputstvo: Pomoću četiri naredbe CIR postiže se $AC \leftarrow [AC/16]$.

40. $y =$ koliko je nenegativnih među $c(200)$, $c(201)$, ..., $c(209)$.

Uputstvo: Neka adresa (uzmimo) 199 služi za promjenljivu n čije će vrijednosti tokom rada programa biti 200, 201, ..., 209. Broj (za koji se gleda da li je nenegativan) donosi se u akumulator pomoću naredbe LDA* 199.

41. $y = 2c(m) + c(n) = 2c(c(200)) + c(c(201))$.

Uputstvo: Naredbe LDA* 200, CIL i ADD* 201.

**Principi programiranja
Konsultacije za Prvi kolokvijum
Programi na Paskalu**

42. Napisati program na Paskalu za računanje i štampanje najvećeg zajedničkog djelioca (NZD) prirodnih brojeva a i b koji se učitavaju.

$$\text{NZD}(a, b) = \text{NZD}(b, a \bmod b),$$

$$\text{NZD}(a, 0) = a \quad (a \neq 0)$$

```
program p42(input,output);
var a,b,x,y: integer;
begin
read(a);
read(b);
while b>0 do
begin
x:=a mod b;
y:=b;
a:=y;
b:=x
end;
write(a)
end.
```

Tokom rada programa, u slučaju ulaznih podataka 120 i 210, dobiće se kao rezultat 30.

43. Program na Paskalu koji štampa prvih 30 članova niza $p_n = \frac{\lambda^n}{n!} e^{-\lambda}$. Broj λ je ulazni podatak.

```
program p43(input,output);
var i: integer;
lamda,broj,imen,p: real;
begin
readln(lamda);
broj:=1.0;
imen:=1.0;
for i:=1 to 30 do
begin
broj:=broj*lamda;
imen:=imen*i;
p:=broj*exp(-lamda)/imen;
writeln(i,p)
end
end.
```

Na primjer, za ulazni podatak 4, dobija se kao rezultat:

```
1 7.3262555555E-02
2 1.4652511111E-01
3 1.9536681481E-01
4 1.9536681481E-01
5 1.5629345185E-01
6 1.0419563457E-01
7 5.9540362610E-02
8 2.9770181305E-02
... ..
29 5.9706690704E-16
30 7.9608920938E-17
```

Lako se vidi da je $\sum_{n=0}^{\infty} p_n = 1, \forall \lambda \in R$.

44. Program na Paskalu za računanje $\binom{n}{k} = \frac{n \cdot (n-1) \cdot \dots \cdot (n-k+1)}{k!}$. Učitavaju se n i k .

Smatramo da je $1 \leq k \leq n$.

```
program p44(input,output);
var n,k,i,broj,imen,komb: integer;
begin
read(n);
read(k);
broj:=n;
imen:=1;
komb:=1;
for i:=1 to k do
begin
komb:=komb*broj div imen;
broj:=broj-1;
imen:=imen+1
end;
write(komb)
end.
```

Na primjer: ulazni podaci: 10 3, rezultat: 120.

45. Učitava se broj n ($1 \leq n \leq 1000$), a računa se i štampa zbir njegovih cifara. Sastaviti odgovarajući program na Paskalu. Na primjer, ako je $n = 349$ onda je zbir = 16.

Oznake: a – cifra hiljada, b – cifra stotina, c – cifra desetica, d – cifra jedinica.

```
program p45(input,output);
var n,a,b,c,d,zbir: integer;
begin
```

```

read(n);
d:=n mod 10;
n:=n div 10;
c:=n mod 10;
n:=n div 10;
b:=n mod 10;
n:=n div 10;
a:=n mod 10;
zbir:=a+b+c+d;
write(zbir)
end.

```

Npr. ulazni podatak: 349, rezultat: 16.

46. Štampati n -ti Fibonaccijev broj f_n . Broj n se učitava. Sastaviti program na Paskalu. Znamo da je $f_1 = 1$, $f_2 = 1$ i $f_n = f_{n-2} + f_{n-1}$ za $n \geq 3$.

```

program Fibonacci(input,output);
label 1;
var n,a,b,c,i: integer;
begin
read(n);
if (n=1) or (n=2) then
begin c:=1; goto 1 end;
a:=1; b:=1; c:=a+b;
if n=3 then goto 1;
for i:=4 to n do
begin a:=b; b:=c; c:=a+b end;
1: write(c)
end.

```

Iz rješenja: $a = f_1$, $b = f_2$, $c = f_3$, itd., $a = f_{n-2}$, $b = f_{n-1}$, $c = f_n$.

Ulazni podatak: 12, rezultat: 144.

47. Štampati vrijednost zbira $S_n = 1 - \frac{1}{2} + \frac{1}{3} - \dots + \frac{(-1)^{n+1}}{n}$. Broj n je ulazni podatak. Program na Paskalu.

```

program p47(input,output);
var i,n: integer;
s,x: real;
begin
read(n);
s:=0.0;
for i:=1 to n do
begin

```

```

x:=1/i;
if i mod 2<>0 then
s:=s+x else s:=s-x
end;
write(s)
end.

```

Vidimo da je sabirak pozitivan kada je broj i neparan, odnosno negativan kada je paran. Može se zamijeniti sa `if i div 2<>0 then` ili se takođe može zamijeniti sa `if odd(i) then` gdje je `odd(i) = false` ako je i paran i `odd(i) = true` ako je i neparan. Engl. odd – neparan. Znano da je $\lim_{n \rightarrow \infty} S_n = \ln 2 = 0,693$.

Ulazni podatak: 10
Rezultat: 6.4563492063E-01

48. Sastaviti program na Paskalu u kome se učitava 100 brojeva, a određuje se (i štampa) najmanji među tim brojevima, kao i najveći.

```

program p48(input,output);
var x: array[1..100]of real;
min,max: real;
n: integer;
begin
for n:=1 to 100 do read(x[n]);
min:=x[1]; max:=x[1];
for n:=2 to 100 do
if x[n]<min then min:=x[n];
for n:=2 to 100 do
if x[n]>max then max:=x[n];
writeln('min=',min);
writeln('max=',max)
end.

```

Tokom rada programa pratimo veličine $\min\{x_1, \dots, x_n\}$ i $\max\{x_1, \dots, x_n\}$ i postepeno napredujemo od $n = 1$ do $n = 100$.

Ulazni podaci: 7 7 7 7 8 8 8 8 ... 26 26 26
26 26, rezultati: 7 26.

49. Učitava se 100 karaktera, svaki je veliko slovo (A ili B ili ... ili Z). Koliko različitih slova ima ukupno, među učitanim karakterima? Program na Paskalu.

```

program p49(input,output);
var x: array[1..100]of char;

```

```

niz: array[65..90] of integer;
m,n,ascii,rez: integer;
ch: char;
begin
for n:=1 to 100 do read(x[n]);
for m:=65 to 90 do niz[m]:=0;
for n:=1 to 100 do
begin
ch:=x[n];
ascii:=ord(ch);
niz[ascii]:=niz[ascii]+1
end;
rez:=0;
for m:=65 to 90 do
if niz[m]>0 then rez:=rez+1;
write(rez)
end.

```

Po ASCII kodu važi ord('A')=65, ord('B')=66, ..., ord('Z')=90 (slično chr(65)='A' i drugo), tako da niz služi da pokaže koliko se puta koje slovo pojavljuje.

Ulaz: MATEMATIKA..MATEMATIKA
Izlaz: 6

50. Učitava se broj n ($0 \leq n \leq 10000$), a štampa se njegov binarni oblik. Program na Paskalu. Na primjer, ako je $n = 24$ onda program treba da odštampa 11000.

$x[1]$ je cifra najveće težine, a $x[15]$ je cifra najmanje težine (cifra jedinica). Ne treba štampati vodeće nule. Pojedina cifra treba da zauzima samo jednu poziciju. Prilikom štampanja, cifre treba da budu jedna do druge.

```

program p50(input,output);
var i,j,n: integer;
x: array[1..15] of integer;
begin
read(n);
for i:=15 downto 1 do
begin
x[i]:=n mod 2;
n:=n div 2
end;
j:=1;
while x[j]=0 do j:=j+1;
for i:=j to 15 do write(x[i]:1)
end.

```

Ulaz 24, izlaz 11000.

51. Učitava se broj n ($-10000 \leq n \leq 10000$), a treba štampati njegov 16-bitni prikaz u obliku PK (potpuni komplement). Program na Paskalu. Na primjer, ako je $n = 24$ onda treba da bude odštampano 0000 0000 0001 1000.

```

program p51(input,output);
label 1;
var x: array[1..16] of integer;
n,k: integer;
begin
read(n);
if n=0 then
begin for k:=1 to 16 do x[k]:=0;
goto 1 end;
if n>0 then
begin
x[1]:=0;
for k:=16 downto 2 do
begin x[k]:=n mod 2;
n:=n div 2 end;
goto 1
end;
if n<0 then
begin x[1]:=1;
for k:=2 to 16 do x[k]:=0;
goto 1 end;
{n=-2^15 poseban slucaj}
x[1]:=1; {n je negativan, prvi bit 1}
n:=-n;
for k:=16 downto 2 do
begin x[k]:=n mod 2;
n:=n div 2 end; {sada je u binarnom}
for k:=2 to 16 do x[k]:=1-x[k];
{sada je u prvom komplementu}
x[16]:=x[16]+1; {dodali smo 1}
{1111+1=10000 i slicno}
{k:=16;
while x[k]=2 do
begin x[k]:=0;
x[k-1]:=x[k-1]+1;
k:=k-1 end;
1: for k:=1 to 16 do write(x[k]:1)
end.

```

Ako unesemo $n = -24$ onda će nam biti saopšteno $x_1, \dots, x_{16} = 1111111111101000$ kao rezultat.

Drugo rješenje bi bilo: prikazati u bimarnom obliku (sa 16 cifara) broj n u slučaju da je $n \geq 0$, odnosno broj $n + 2^{16} = n + 65536$ u slučaju da je $n < 0$.

52. Učitava se broj n ($1 \leq n \leq 366$) čiji je smisao: n -ti po redu dan u 2008. godini. Program na Paskalu treba da odštampa koji je to datum, u kom mjesecu, kao i koji je to dan u sedmici.

Smisao promjenljivih: d – datum, m – mjesec, $a[i]$ – koliko ima dana koji mjesec, $b[i] = a[1] + \dots + a[i]$.

```

program p52(input,output);
var a,b: array[1..12]of integer;
n,m,d,dan,i: integer;
begin
read(n);
for i:=1 to 12 do a[i]:=31;
a[2]:=29; a[4]:=30; a[6]:=30;
a[9]:=30; a[11]:=30;
b[1]:=a[1];
for i:=2 to 12 do b[i]:=b[i-1]+a[i];
m:=1;
for i:=1 to 11 do if n>b[i] then m:=i+1;
d:=n;
for i:=1 to m-1 do d:=d-a[i];
dan:=n mod 7;
if dan=0 then dan:=7;
write(d); write(' ');
case m of
1: write('januar'); 2: write('februar');
3: write('mart'); 4: write('april');
5: write('maj'); 6: write('jun');
7: write('jul'); 8: write('avgust');
9: write('septembar');
10: write('oktobar');
11: write('novembar');
12: write('decembar') end;
write(' ');
case dan of
1: write('utorak'); 2: write('srijeda');
3: write('cetvrtak'); 4: write('petak');
5: write('subota'); 6: write('nedjelja');
7: write('ponedjeljak') end
end.

```

Razmatrana godina je prestupna pa je zato $a[2] = 29$. Prvi dan (1. januar) u razmatranoj

godini je utorak pa zato slučajevi (case) počinju sa utorkom.

```

ulaz 1 izlaz 1 januar utorak
ulaz 2 izlaz 2 januar srijeda
ulaz 31 izlaz 31 januar cetvrtak
ulaz 32 izlaz 1 februar petak
ulaz 122 izlaz 1 maj cetvrtak
ulaz 366 izlaz 31 decembar srijeda

```

53. Učitava se prirodan broj. Program na Paskalu treba da saopšti da li je taj broj prost ili je pak (suprotno) složen.

Smatramo da je $n \geq 2$ (za $n = 1$ ne kaže se ni da je prost ni da je složen). U funkcijskom potprogramu prost ispituje se da li je njegov argument n prost broj. Ispituje se uslov: za svako k , n nije djeljiv sa k , gdje je $2 \leq k \leq \sqrt{n}$.

```

program p53(input,output);
var n: integer;
function prost(n: integer): boolean;
var k: integer; b: boolean;
begin
b:=true;
k:=2;
while (k*k<=n) and b do
begin
if n mod k=0 then b:=false;
k:=k+1
end;
prost:=b
end;
begin
read(n);
if prost(n) then write('prost')
else write('slozen')
end.

```

Npr. ulaz 2477, izlaz prost ili: ulaz 2479, izlaz slozen ili: ulaz 2481, izlaz slozen i slično.

54. Koliko ima parova blizanaca (parova prostih brojeva koji se razlikuju za dva) do 1000? Napisati program na Paskalu.

Kandidati su parovi (3, 5), (5, 7), ..., (997, 999). Par $(n, n + 2)$ je par blizanaca ako su n i $n + 2$ prosti. Promjenljiva broj služi da izbroji parove blizanaca.

```

program p54(input,output);
var n,broj: integer;
  function prost(n: integer): boolean;
  var k: integer; b: boolean;
  begin
    b:=true;
    k:=2;
    while (k*k<=n) and b do
      begin
        if n mod k=0 then b:=false;
        k:=k+1
        end;
    prost:=b
  end;
begin
  broj:=0;
  n:=3;
  repeat
    if prost(n) and prost(n+2) then
      broj:=broj+1;
      n:=n+2
  until n=999;
  write(broj)
end.

```

Rezultat: 35.

55. Učitava se paran broj n . Neka program na Paskalu odgovori sa "da" ili "ne" na pitanje: da li n može da se prikaže kao zbir dva prosta broja.

Znamo da Goldbachova hipoteza glasi: svaki paran broj veći od dva može da se prikaže kao zbir dva prosta broja.

Smatramo da je $n \geq 4$. Goldbachova hipoteza glasi: svaki paran broj $n \geq 4$ može da se prikaže u obliku $n = p + q$, gdje su p i q prosti brojevi. U programu, kada se učita n , mi ispitujemo da li su oba broja p i $n - p$ prosti za $p = 3, p = 5, \dots$ sve dok je $p \leq \frac{n}{2}$ odnosno (prije će se desiti) dok ne nađemo oba prosta.

```

program p55(input,output);
label 10,20,30;
var n,p,q: integer;

function prost(n: integer): boolean;
var k: integer; b: boolean;
begin
  b:=true; k:=2;
  while (k*k<=n) and b do

```

```

  begin if n mod k=0 then b:=false;
        k:=k+1 end;
  prost:=b
end;

BEGIN
  read(n);
  if n=4 then
    begin p:=2; q:=2; goto 10 end;
  p:=3;
  repeat
    if prost(p) and prost(n-p)
    then goto 20;
    p:=p+2
  until p>n div 2;
  write('nema rastavljanja');
  goto 30;
20: q:=n-p;
10: write(p,'+',q);
30: write('Goldbach')
  END.

```

Ulaz 1080 izlaz 11 + 1069, ulaz 1082 izlaz 13 + 1069, ulaz 1084 izlaz 23 + 1061 i slično.