



QBASIC ZA POČETNIKE

Uvod

Dakle, kao što vam sâm naslov govori, ovo je vodič za totalne početnike u programiranju. Dobro, ne baš TOTALNE... Trebali biste već znati neke PRAVE osnove, tipa "Što je programiranje?" ili "A compiler?" ili nešto slično (npr. da se program izvršava red po red odozgo prema dolje). Takve osnove prepuštamo nekome drugome. Ovo je za one koji ne znaju skoro ništa, a ono nešto što znaju je samo u teoriji. I odmah vam dajem prvo **upozorenje**: s QBasicom nećete moći ništa. To je samo za učenje, od njega ništa upotrebljivog. Dobra vijest je da je poslije poprilično lako prijeći na Visual Basic (osobito .Net verzija), što je itekako upotrebljivo. Eto, to je to od uvoda, znači, krećemo...

Naredba CLS

Svaki programski jezik, pa tako i QBasic, ima komad kôda koji ide na sami početak programa jer... jer to tako jednostavno je. Kasnije ćete saznati što to točno znači i zašto to tako je, za sada morate znati samo da na početku SVAKOG programa ide:

```
CLS
```

Reći ću vam da je **CLS** kratica od **Clear Screen** (Očisti Ekran), tek toliko da je lakše zapamtite. Ajmo dalje...

Naredba PRINT

Prva "prava" naredba koju ćete učiti je naredba **PRINT**. Ona služi za ispis teksta na ekran. I nema **nikakve veze** ni sa kakvim printerom niti printanjem (da ne bi bilo nisam znao/la)! Ono što piše poslije naredbe **PRINT** bit će ispisano na ekranu kad se program pokrene. Pa napišite:

```
CLS
```

```
PRINT "Hello world!"
```

i pokrenite program pritiskom na tipku F5. Ekran će pocrnjeti i na njemu će biti napisano: Hello world!

Qbasic izvršava programe kao u DOS-u, pa na raspolaganju imate 80 znakova u svakom od 25 redova na ekranu. Kao što možete vidjeti, da biste nešto ispisali, morate to staviti u navodnike. Ako želite koristiti novu naredbu **PRINT**, to ćete napraviti u novom redu. I kad pokrenete program, to što ste napisali će se pojaviti u novom redu. Ako vam je sve to još malo mutno, igrajte se. To mi je savjet za svaki put kad vam nešto bude "malo mutno" (osim u subotu navečer), pa ga zapamtite.

Napomena: Sve možete pisati malim slovima. Kad prijedete u novi red, Qbasic će automatski naredbe napisati velikim slovima, a znakove matematičkih operacija razdvojiti (npr. **3+2** će postati **3 + 2**). Naravno, ono pod navodnicima ostaje isto.



created by *Dejan Drabić*

Sad napišite:

```
CLS  
PRINT 3
```

Pogledate li rezultat (F5), vidjet ćete da radi i bez navodnika. No, ako umjesto **3** napišete **Hello**, definitivno ne radi (probajte). A ako napišete **3 + 2...** sad vam je već jasnije. Matematičke operacije bivaju izračunate (u, al profesionalno zvučim!) i njihov rezultat ispisan (rezultat od 3 je 3, da ne bi bilo zabune). A slova moraju ići u navodnike. Inače ih Qbasic pokušava shvatiti kao brojeve, a to onda nikako ne valja.

Osnove varijabli

Varijable su (zapravo) prilično komplicirane stvari koje imaju veze s RAM-om i onom nečemu u procesoru itd. (neću napisati ništa više da ne ispadnem glup). Ono što vama treba je da znate da su to... kao posude u kojima čuvate podatke. Svaka od njih ima svoje ime (naljepnica na posudi) i svoju vrijednost (ono što je u posudi). Npr. Ja želim spremiti podatak (koji će mi poslije trebati) da Mirko ima 23 godine. Znači, uzmem posudu, na nju nalijepim naljepnicu "Mirkove godine" i unutra ubacim broj 23 (ne pitajte kako, samo zamislite!). U Qbasicu bi to bilo da kreiram **varijablu** naziva **Mirko.godine** i dodijelim joj vrijednost **23**, odnosno:

```
Mirko.godine = 23
```

Onda kreiram još jednu varijablu u koju napišem da Slavko ima 26 godina:

```
Slavko.godine = 26
```

Sada s tim varijablama mogu raditi kao da su one - njihove vrijednosti, odnosno, kad napišem **Mirko.godine**, Qbasic će to razumjeti kao **23**. Shodno tome (evo me opet profesionalnog), ako napišemo

```
PRINT Mirko.godine + Slavko.godine
```

, računalo će, kad se program pokrene, ispisati **49**.

A možemo kreirati još jednu varijablu koja će biti zbroj Mirkovih i Slavkovih godina:

```
MiS.Godine = Mirko.godine + Slavko.godine
```

Sada je u varijabli **MiS.Godine** (ovo je slučajno ispalo slično kao miss of the year, šta da radim) vrijednost **49**, i s njom možemo raditi što nas volja. O imenima varijabli više na kraju ovog dijela, inače, ona uopće nisu važna – ove tri varijable su se isto tako mogle zvati **a**, **b** i **c** ili **Jozo**, **Pero** i **Djuro** ili bilo kako drugačije (dobro, ne bilo kako... napisat ću na kraju).

A što ako želimo sačuvati podatak (staviti u posudu) da se Mirkova žena zove Fata? (dobro, malo miješam viceve, ali to nema veze). Kod naredbe **PRINT** sam rekao da brojke (koje služe za matematičke operacije) ne idu pod navodnike, a slova idu. To vrijedi i ovdje. Jedini je problem što odmah poslije varijable u koju idu slova uvijek treba napisati znak \$ ([Shift] + [4]). Dakle:

```
Mirkova.zena$ = "Fata"
```



created by *Dejan Drabić*

O varijablama će se pisati još, ali to ćemo poslije.

Naredba INPUT

Služi za to da bi korisnik (onaj koji se služi vašim programom) mogao nešto upisati. Nakon upisivanja riječi **INPUT**, obično se piše poruka pod navodnicima (tipa "Upiši jedan troznamenasti broj."). Ako se napišu navodnici, poslije njih ide zarez (,) pa zatim varijabla u koju će biti upisana vrijednost koju korisnik napiše (opet, ako se očekuje unos slova, poslije varijable treba ići znak \$).

```
CLS
INPUT "Koliko Mirko ima godina?", Mirko.godine
INPUT "Koliko Slavko ima godina?", Slavko.godine
MiS.Godine = Mirko.godine + Slavko.godine
PRINT "Mirko i Slavko zajedno imaju ovoliko godina:"
PRINT MiS.Godine
PRINT
INPUT "Kako se zove Mirkova žena?", Mirkova.zena$
PRINT "Ime Mirkove žene:"
PRINT Mirkova.zena$
```

Napomene:

- Onaj jedan **PRINT** koji je sâm u jednom redu služi za to da bi dobili jedan red razmaka (rekoh da svaki **PRINT** piše svoje u sljedeći red, pa je i ovaj napisao svoje ništa u sljedeći red).
- Pošto s imenom Mirkove žene ne mogu ništa, samo sam ga ispisao (zadnji **PRINT**)

Još samo da kažem da s jednom naredbom **INPUT** možete "natjerati" korisnika da upiše više stvari (ali bi bilo dobro da mu to i kažete - za to služe navodnici pokraj **INPUT**). To se radi tako da umjesto jedne varijable na kraju, napišete više varijabli odvojenih zarezom. Kad korisnik to bude morao upisati, i on će varijable odvajati zarezom. Tako će naredba

```
INPUT "Upiši tri broja: ", a, b, c
```

zahtijevati od korisnika da upiše npr.

23, 1, 58

i stisne tipku [Enter]. (Brojevi su proizvoljni, naravno, a razmaci u upisivanju nisu nužni. To je moglo biti i **23,1,58.**)

Evo vam i primjer...

```
CLS
INPUT "Upiši tri broja koja želiš zbrojiti: ", a, b, c
zbroj = a + b + c
PRINT "Zbroj ta tri broja je:"
PRINT zbroj
```

Sve kužite? Na ekranu bi (na kraju programa) trebalo biti nešto tipa:



created by *Dejan Drabić*

Upiši tri broja koja želiš zbrojiti: 23, 1, 58
Zbroj ta tri broja je:
82

Dodaci

- Naredba PRINT

Kad sam trebao napisati zbroj Mirkovih i Slavkovih godina, u kôdu sam napisao:

```
PRINT "Mirko i Slavko zajedno imaju ovoliko godina:"  
PRINT MiS.Godine
```

No, to se moglo napraviti i ljepše, ovako:

```
PRINT "Mirko i Slavko zajedno imaju"; MiS.Godine; "godina."
```

Pa bi ispalo:

Mirko i Slavko zajedno imaju 49 godina.

Tu se nema što objašnjavati... sami skužite. Broj mijenjanja između texta i varijable je neograničen, samo treba na prijelazu staviti točku-zarez (;). Zapravo, ne treba. Qbasic će to napraviti umjesto vas! (fuj, ko reklama za usisavač, multipraktik i tenk u jednom!). Prema tome, ako napišete

```
PRINT "Mirko i Slavko zajedno imaju" MiS.Godine "godina."
```

, nakon prelaska u novi red, to će postati

```
PRINT "Mirko i Slavko zajedno imaju"; MiS.Godine; "godina."
```

Kao što ćete vidjeti, Qbasic automatski dodaje razmake između varijabli i texta. Moguće je staviti i više varijabli zaredom, pa i više textova zaredom, makar to i nema previše smisla (mislim na textove, više varijabli ima smisla). Ako stavite zarez (a ne točku-zarez), bit će kao da ste stisnuli tipku TAB (malo žešće razdvojeno – omogućava pisanje u stupcima). Isprobavajte!

Umjesto da pišete **PRINT** svako malo, možete napisati i upitnik [?]. Kad prijedete u novi red, upitnik će automatski postati **PRINT**.

- Varijable

Ime varijable može imati **do 256 znakova**, ne smije počinjati brojem, velika i mala slova nisu važna, smije sadržavati znakove:

abcdefghijklmnopqrstuvwxyz.0123456789 (ova točka se isto ubraja)

i mislim da je to sve.

Dajte imena varijablama pametno, tako da znate što je što (kad budete radili program s parsto varijabli, znat ćete o čemu pričam). Sad vam je vjerojatno bezveze svako malo pisati

Mirko.godine i **Slavko.godine** – zašto ne biste pisali **a** i **b** kad je lakše? – ali, poslije će vam to dobro doći.

- Ostalo...



created by *Dejan Drabić*

Decimalna točka je točka, ne zarez.

Umjesto **0.25**, možete napisati samo **.25**. Štoviše, samo će se promijeniti u **.25**.

U vašem kodu će vam često trebati vaši komentari (tipa: "ovo je tu tak da bi mi poslije bilo laše napraviti ono tam" ili sl.). A pogotovo ako budete radili na nekom projektu zajedno s nekim. Komentari se u QBasicu pišu tako da ispred njih (tj. lijevo od njih) jednostavno stavite apostrof (') {kao [?], ali bez tipke [Shift]}. Sve desno od apostrofa QBasic ne gleda. Primjere ćete moći vidjeti u rješenjima zadataka uz ovaj vodič.



created by *Dejan Drabić*

IF-THEN-ELSE naredbe

Ako znate imalo engleskog, shvatili ste o čemu se radi. **IF-THEN-ELSE** znači **AKO-ONDA-INAČE**, dakle imamo posla s uvjetima. Evo jednostavnog primjera za početak. Treba nam program koji će nam reći je li učitani broj veći od nule.

```
CLS
INPUT "Upiši broj: ", x
IF x > 0 THEN PRINT "Je."
```

Stvarno je jednostavno. Naredbu **IF** (zadnji red) čitamo onako kako je napisano: **AKO** je x veći od nule, **ONDA** ispiši "Je".

I sad vama čudno... pričalo se o nekom **ELSE**u... Pa evo vam! Prvom primjeru dodajte da napiše je li veći **ili manji** od nule.

```
CLS
INPUT "Upiši broj: ", x
IF x > 0 THEN PRINT "Je." ELSE PRINT "Nije."
```

Ovo čitate kao i prvi primjer, samo mislim da bolje zvuči kad umjesto **INAČE** kažete **A AKO NIJE**:

AKO je x veći od nule, **ONDA** ispiši "Je", **A AKO NIJE (INAČE)**, ispiši "Nije".

A sad zamislite da smo to sve htjeli napisati lijepo: "Broj (taj i taj) je veći od nule" (ili nije).
Gledajte:

```
CLS
INPUT "Upiši broj: ", x
IF x > 0 THEN PRINT "Broj"; x; "je veći od nule." ELSE PRINT "Broj"; x; "nije veći od nule."
```

Ajme gužve u jednom redu!!! Zato je bolje pisati u više redova, a naredbe uvlačiti (TAB):

```
CLS
INPUT "Upiši broj: ", x
IF x > 0 THEN
[color=white]TAB PRINT "Broj"; x; "je veći od nule."
ELSE
    PRINT "Broj"; x; "nije veći od nule."
END IF[/color]
```

Napomena: Pošto je u web stranici poprilično nemoguće napisati nešto uvučeno (tipkom TAB), a uvlačenja su (bar meni) podosta važna u programiranju, koristio sam bijelu boju teksta da dobijete privid uvučenog. Žao mi je ako budete morali kopirati neki kod, onda ćete svaki bijeli "TAB" morati brisati i ...

Eto, sad je ljepše, jedino morate zapamtiti da, kad pišete u više redova, na kraju mora pisati **END IF**. Kad pišete u jednom redu, **NE SMIJE!!!**



created by *Dejan Drabić*

Još jedan jednostavan primjer: password. Evo programčića koji kaže je li lozinka ispravna (u ovom primjeru, lozinka je KjuBejzik).

```
CLS
INPUT "Password? ", a$
IF a$ = "KjuBejzik" THEN
[color=white]TAB PRINT "Password confirmed"
ELSE
    PRINT "Error: wrong password"
END IF[/color]
```

Vratimo se sad drugom primjeru. Tamo imamo grešku. Ako upišemo broj 0, program će napisati da je nula manja od nule!!! Sad nam treba naredba **ELSEIF** (ne ELSE IF!!!):

```
CLS
INPUT "Upiši broj: ", x
IF x > 0 THEN
[color=white]TAB PRINT "Broj"; x; "je veći od nule."
ELSEIF x < 0 THEN
    PRINT "Broj"; x; "nije veći od nule."
ELSE
    PRINT "Upisali ste nulu."
END IF[/color]
```

Čitamo:

```
AKO je x veći od nule, ONDA
    ispiši "Je"
INAČE, AKO je x manji od nule (odnosno, "AKO nije veći od nule, nego je
manji")
    ispiši "Nije".
INAČE
    ispiši "Upisali ste nulu."
KRAJ
```

Napomena: broj **ELSEIF** naredbi je neograničen.

Evo još jednog (stvarno debilnog) primjera koji postoji samo da bi se ljudi zapitali ima li smisla nastaviti s učenjem programiranja. Uglavnom, program pita pada li kiša i kaže je li potrebno ponjeti kišobran.

```
CLS
INPUT "Pada li kiša? ", a$
IF a$ = "Da" THEN
[color=white]TAB PRINT "Ponesi kišobran."
ELSEIF a$ = "Ne" THEN
    PRINT "Nemoj ponijeti kišobran."
ELSE
    PRINT "Daj, piši razgovjetnije!!!"
```



created by *Dejan Drabić*

```
END IF  
[/color]
```

Ovaj zadnji **PRINT** je tu jer, ako korisnik napiše "ne" (s malim n), QBasic to neće pročitati kao "Ne", i to neće valjat.

CASE naredba

Pogledajte ovaj primjer. Recimo da vam jednom u nekom programu za rad s datumima zatreba dio programa koji će ispisivati hrvatski naziv mjeseca (a poznat je redni broj):

```
CLS  
INPUT "Broj mjeseca: ", x  
  
IF x = 1 THEN  
[color=white]TAB PRINT "Siječanj"  
ELSEIF x = 2 THEN  
    PRINT "Veljača"  
ELSEIF x = 3 THEN  
    PRINT "Ožujak"  
ELSEIF x = 4 THEN  
    PRINT "Travanj"  
ELSEIF x = 5 THEN  
    PRINT "Svibanj"  
ELSEIF x = 6 THEN  
    PRINT "Lipanj"  
ELSEIF x = 7 THEN  
    PRINT "Srpanj"  
ELSEIF x = 8 THEN  
    PRINT "Kolovoz"  
ELSEIF x = 9 THEN  
    PRINT "Rujan"  
ELSEIF x = 10 THEN  
    PRINT "Listopad"  
ELSEIF x = 11 THEN  
    PRINT "Studeni"  
ELSEIF x = 12 THEN  
    PRINT "Prosinac"  
ELSE  
    PRINT "Taj mjesec ne postoji."  
END IF[/color]
```

Puno pisanja, jel'da? Za to služi **CASE** naredba. Najbolje ćete skužiti na primjeru, pa evo vam isto ovo pomoću naredbe **CASE**:

```
CLS  
INPUT "Upiši broj mjeseca: ", x  
  
SELECT CASE x  
CASE 1  
[color=white]TAB PRINT "Siječanj"
```




created by *Dejan Drabić*

```
CASE 2
  PRINT "Veljača"
CASE 3
  PRINT "Ožujak"
CASE 4
  PRINT "Travanj"
CASE 5
  PRINT "Svibanj"
CASE 6
  PRINT "Lipanj"
CASE 7
  PRINT "Srpanj"
CASE 8
  PRINT "Kolovoz"
CASE 9
  PRINT "Rujan"
CASE 10
  PRINT "Listopad"
CASE 11
  PRINT "Studeni"
CASE 12
  PRINT "Prosinac"
CASE ELSE
  PRINT "Taj mjesec ne postoji."
END SELECT[/color]
```

Manje texta, preglednije, lakše za copy/paste... Znači, na početku naredbe ide:
`SELECT CASE varijabla`

Poslije toga za svaku vrijednost:

```
CASE vrijednost
(vrijednost može biti i niz slova, onda se to stavlja u navodnike i treba biti $ poslije varijable)
```

Umjesto **ELSE**:

```
CASE ELSE
```

I na kraju:

```
END SELECT
```

GOTO naredba

GOTO se sastoji od dvije riječi: **GO TO** (ko bi reko!), odnosno **IDI NA**. Služi za preskakanje dijela kôda ili vraćanje nazad, ali nema neku pretjeranu primjenu. Mislim da se svako korištenje naredbe **GOTO** može napisati na drugačiji način (bilo grananjem, petljama...). Uglavnom, na početku jednog reda se napiše broj, a kad bilo gdje u programu bude napisano **GOTO** taj broj, QBasic se vraća nazad ili preskače kôd do linije na čijem je početku taj broj. (sad sam ga malo



created by *Dejan Drabić*

zakomplicirao, al skužit ćete.)

Primjer: program upisuje brojeve dok se ne upiše nula. Onda ispiše zbroj svih upisanih brojeva:

```
CLS
zbroj = 0

20 INPUT "Upiši broj: ", x

IF x = 0 THEN
[color=white]TAB GOTO 10
ELSE
zbroj = zbroj + x
    GOTO 20
END IF

10 PRINT zbroj[/color]
```

Znači, kad QBasic dođe na **GOTO 10**, on ode na 10, odnosno, izvrši naredbu **PRINT zbroj**.

Onda ide dalje, a dalje nema ništa, pa je program gotov.

Kad dođe na **GOTO 20**, ode na 20, odnosno, vrati se na **INPUT**, izvrši ga i ide dalje, na naredbu **IF**.

Isprobavajte, pa ćete skužiti.

Matematika i logika

Uskoro će vam trebati neke žešće matematičke operacije (i funkcije itd.) osim +, -, *, /, <, > i =. Evo vam pregled svih za koje se sad mogu sjetiti.

+, -, *, /, =
--- to znate

<, >
--- manje od, veće od

<=, >=, <>
--- manje ili jednako, veće ili jednako, nejednako

MOD
--- daje ostatak pri dijeljenju.
13 MOD 5 = 3
(13 / 5 = 2 i ostatak 3)

ABS
--- daje apsolutnu vrijednost broja.
ABS(-5) = 5
(matematički napisano: $|-5| = 5$)



created by *Dejan Drabić*

^ ([ALT] + [94])
--- **exponencija.**
2 ^ 3 = 8

SQR
--- **kratica od square root - kvadratni korijen**
SQR(16) = 4

SIN, COS, TAN
--- **daje sin, cos i tan nekog kuta U RADIJANIMA.**
COS(pi) = -1

Napomene:

- Da biste pretvorili stupnjeve u radijane, pomnožite stupnjeve sa (pi / 180).
[color=gray]SIN(90 * (pi / 180)) = 1
- kad radite s brojem pi, bilo bi dobro da na početku programa napišete
pi = 3.141592654[/color]

EXP
--- **e na n-tu.**
EXP(1) = 2.718281828

LOG
--- **limes.**
LOG(EXP(1)) = 1

INT
--- **zaokružuje na prvi manji cijeli broj**
INT(5.76) = 5
INT(-5.2) = -6

FIX
--- **miče decimalni dio broja**
FIX(1.8) = 1
FIX(-5.2) = -5

CINT
--- **zaokružuje na najbliži cijeli broj**
CINT(2.9) = 3
CINT(-2.2) = -2
CINT(2.5) = 2

Logički operateri (tako ih svi zovu, nemam pojma šta točno znači "operater", i nije važno) su male riječi koje nam pomažu u nekim stvarima. Prvo pregled, onda primjer. Dobro ga proučite.

AND - i
obje strane moraju biti istinite



created by *Dejan Drabić*

OR - ili

bar jedna strana mora biti istinita

XOR

jedna strana mora biti istinita, ali ne i druga.

(iskreno, ovo nikad nisam koristio)

Primjer:

```
x = 2
y = 7
IF x = 2 AND y > 5 THEN PRINT "Da." ELSE PRINT "Ne.";
IF x = 3 AND y > 5 THEN PRINT "Da." ELSE PRINT "Ne.";
IF x <> 3 AND y >= 5 THEN PRINT "Da." ELSE PRINT "Ne."

IF x = 3 OR y > 5 THEN PRINT "Da." ELSE PRINT "Ne.";
IF x = 2 OR y = 5 THEN PRINT "Da." ELSE PRINT "Ne.";
IF x = 1 OR y = 5 THEN PRINT "Da." ELSE PRINT "Ne."

IF x => 2 XOR y = 5 THEN PRINT "Da." ELSE PRINT "Ne.";
IF x > 2 XOR y = 7 THEN PRINT "Da." ELSE PRINT "Ne.";
IF x = 2 XOR y < 8 THEN PRINT "Da." ELSE PRINT "Ne."
```

Ispis:

```
Da. Ne. Da.
Da. Da. Ne.
Da. Da. Ne.
```



created by *Dejan Drabić*

FOR-NEXT petlja

Zamislite da dobijete zadatak da morate ispisati prvih 10 prirodnih brojeva. Postoji nekoliko rješenja (ja ću napisati tri):

1.) Koristeći naredbu **PRINT** (i copy/paste). Najidiotskije rješenje.

```
CLS
PRINT 1
PRINT 2
PRINT 3
PRINT 4
PRINT 5
PRINT 6
PRINT 7
PRINT 8
PRINT 9
PRINT 10
```

2.) Koristeći naredbu **PRINT** i **GOTO**. To je najpametnije što možete napraviti s trenutnim znanjem (iz prošla 2 vodiča):

```
CLS
i = 1
10 PRINT i
i = i + 1
IF i <= 10 THEN GOTO 10 ELSE END
```

Pojašnjenje: **i** je 1. To se ispiše. Tada **i** postaje 2. Ako je još uvijek u skupu brojeva do 10, vraća se na ispis **i** i ponovo povećava za 1. Kad **i** postane 11, ne ispisuje se, nego se program završava.

3.) Koristeći petlju **FOR-NEXT**:

```
CLS
FOR i = 1 TO 10
    PRINT i
NEXT i
```

Pojašnjenje:

- Naredba **FOR** označava početak **FOR-NEXT** petlje; naredba **NEXT** označava kraj.
- Sve između te dvije naredbe se najčešće zove "niz naredbi koji se ponavlja".
- Prvi put kad se ponavlja (u ovom slučaju naredba **PRINT i**), **i** je 1. Sljedeći put je 2, pa 3... do 10. To piše odmah poslije naredbe **FOR** – **i** ide od 1 do 10.
- Dakle, program ide odozgo prema dolje. Kad dođe do **NEXT**, povećava varijablu **i** i vraća se nazad na red poslije **FOR** (ako varijabla još nije dosegla konačnu vrijednost) ili nastavlja dalje (ako jest).

Još jedan primjer – program zbraja prvih **n** prirodnih brojeva:



created by *Dejan Drabić*

```
CLS
INPUT "Koliko brojeva? ", n
zbroj = 0

FOR i = 1 TO n
    zbroj = zbroj + i
NEXT i

PRINT zbroj
```

Naredba **zbroj = zbroj + i** će se ponavljati **n** puta, **i** će prvi put biti 1, pa 2... i tako sve do **n**. Kad **i** dostigne broj **n**, zbroj će se ispisati. Jednostavno je!!!

Napišite program koji će množiti samo prirodne parne brojeve do broja **n**. A šta sad? Jedna je opcije da unutar **FOR** petlje provjeravate je li taj broj paran ili ne (**IF i MOD 2 = 0 THEN... ELSE...**). Ali, lakše je koristiti naredbu **STEP**:

```
CLS
INPUT "Do koliko? ", n
umnozak = 1

FOR i = 2 TO n STEP 2
    umnozak = umnozak * i
NEXT i
PRINT umnozak
```

Prvi put će **i** biti 2 (jer je **i = 2 TO n...**). Dalje će se uzimati svaki drugi broj (...**STEP 2**), dakle 4, 6, 8...

Broj poslije **STEP** naredbe može biti i negativan. Npr. ispiši sve brojeve od -5 do 5 unazad:

```
CLS
FOR i = 5 TO -5 STEP -1
    PRINT i
NEXT i
```

Napomena: varijabla koja se mijenja (ona poslije **FOR** i **NEXT**) ne mora uvijek biti **i**. Može biti i humuhumunukunukuapuaa, ali je **i** uobičajeno.

DO-LOOP petlja

Koristi se za ponavljanje nekih naredbi ("niz naredbi koji se ponavlja") dok neki uvjet ne bude zadovoljen / dok je neki uvjet istinit. Recimo, primjer s početka ovog članka (ispisati brojeve do 10) se može napisati ovako:

```
CLS
i = 1
DO
```



created by *Dejan Drabić*

```
PRINT i
i = i + 1
LOOP UNTIL i > 10
```

Naredbe unutar petlje (**PRINT i & i = i + 1**) će se ponavljati dok **i** ne postane veće od 10 (**UNTIL i > 10**). Opet, kad program dođe do naredbe **LOOP**, provjerava uvjet **i** ovisno o tome se vraća na red poslije **DO** ili nastavlja dalje.

Ovo isto se moglo napraviti malčice drukčije:

```
CLS
i = 1
DO
    PRINT i
    i = i + 1
LOOP WHILE i <= 10
```

Opet ista priča, jedino što se te dvije naredbe ponavljaju dok god je **i <= 10** (**WHILE i <= 10**). Pomaže ako znate značenje engleskih riječi "until" i "while", problem je što se one na hrvatski prevode skoro jednako ("dok").

Napomena: naredbe UNTIL ili WHILE se mogu staviti poslije naredbe DO umjesto poslije LOOP. Razlika je u tome što se onda uvjet provjerava na početku. Logikom ćete lako doći do rješenja na bilo koji način (obično možete na sva četiri - UNTIL kod LOOP, UNTIL kod DO, WHILE kod LOOP, WHILE kod DO).

No, ovo je puno lakše napraviti pomoću naredbe **FOR-NEXT**. Čemu onda služi **DO-LOOP** petlja? Evo vam opet odgovor pomoću primjera: Program upisuje brojeve dok se ne upiše 0. Onda ispiše zbroj.

```
CLS
zbroj = 0

DO
    INPUT "Upiši broj: ", x
    zbroj = zbroj + x
LOOP UNTIL x = 0

PRINT zbroj
```

Sve vam je jasno: petlja se vrti i vrti i vrti dok god **x nije = 0**. Ne smeta što se ta nula na kraju zbroji – rezultat je isti.

Evo vam još: Program upisuje niz brojeva dok se ne upiše broj veći od 999 i ispisuje najmanji od njih.

```
CLS
min = 1000
DO
    INPUT "Upiši broj: ", x
```



created by *Dejan Drabić*

```
IF x < min THEN min = x
LOOP UNTIL x > 999
PRINT min
```

Skušajte sami.

WHILE-WEND petlja

Potpuno bespotreban dio QBasica. Prvi primjer (s početka članka):

```
CLS
i = 1
WHILE i <= 10
    PRINT i
    i = i + 1
WEND
```

Čitajući i prevodeći na hrvatski: dok god je $i \leq 10$, ispiši i i povećaj ga za 1. Potpuno isto kao:

```
CLS
i = 1
DO WHILE i <= 10
    PRINT i
    i = i + 1
LOOP
```

Jedino što se s **WHILE-WEND** petljom uvjet uvijek provjerava na početku i nema korištenja riječi **UNTIL**. Jadno. Dakle, koristite vi petlju **DO-LOOP** i sve će biti OK.

"Ugniježdene" petlje (nested loops)

Nitko vam ne brani da napravite jednu petlju unutar druge. Recimo da hoćete napisati tablicu množenja (do 10x10):

```
CLS
FOR i = 1 TO 10
    FOR j = 1 TO 10
        PRINT i * j;
    NEXT j
    PRINT
NEXT i
```

Malo je grbavo, al nema veze...

Svaki put kad se izvršava **PRINT i * j**, varijable su drugačije:

- 1.) $i = 1, j = 1$
- 2.) $i = 1, j = 2$
- 3.) $i = 1, j = 3$
- (...)



created by *Dejan Drabić*

10.) $i = 1, j = 10$

11.) $i = 2, j = 1$

12.) $i = 2, j = 2$

(...)

100.) $i = 10, j = 10$

Poslije **PRINT i * j** stoji točka-zarez (;) da bi se sljedeći put ispisivalo u istom retku. Nakon **NEXT j** piše samo **PRINT** da bi se ispisalo ništa, i prešlo u novi red.



created by *Dejan Drabić*

Tipovi varijabli

Dosad su varijable koje smo koristili bile brojevi ili slova (ostali znakovi). Sad ćemo upoznati ostale oblike brojevnih varijabli (i malo produbiti znanje o varijablama koje sadrže slova).

QBasic koristi pet vrsta varijabli:

integer: cijeli broj između -32,768 i 32,767

long: cijeli broj između -2,147,483,648 i 2,147,483,647

single: decimalni broj "jednostruke preciznosti" (što god to značilo)

double: decimalni broj "dvostruke preciznosti" (isto ko i iznad)

string: znakovi koji se ne shvaćaju kao brojevi (mislim da najviše 256 znakova)

Naredba DIM

U većini programskih jezika, potrebno je "deklarirati" varijablu prije nego što ju upotrijebite. To znači da morate napisati naredbu u kojoj ćete reći programu ime varijable i njezin tip. U QBasicu se to radi naredbom **DIM**:

DIM ime_varijable AS tip_varijable

Dakle, da bi rekli QBasicu da je varijabla blah tipa integer, moramo napisati:

DIM blah AS INTEGER

NAPOMENA: Ako se to ne napiše, QBasic pretpostavlja da je varijabla tipa single.

Drugi način

Sjećate se da ste, kad vam je trebala tekstualna varijabla, poslije njezinog imena uvijek stavljali znak \$? E, to ima svoj razlog koji ćete sada saznati. Svaki tip podataka ima takav znak, tako da naredba DIM zapravo ne služi skoro ničemu – ne morate deklarirati varijable, samo ih počnite koristiti i upotrebljavajte ove znakove:

integer: %

long: &

single: !

double: #

string: \$

Put a deset na...

Ako je neki broj prevelik (ili premalen) za tipove single ili double, on će biti napisan u obliku iz podnaslova – broj 1234567890123456789 će se pretvoriti u:

- ako je tip **single**: 1.234568E+18 (1.234568 x 10 ^ 18)

- ako je tip **double**: 1.234567890123457D+18 (1.234567890123457 x 10 ^ 18)



U tom obliku broj možete upisati i kad pišete kôd i kad korisnik upisuje nešto (naredba INPUT).

Čemu to služi?

U "pravim" programima (koji se ne pišu u QBasicu ;) , to služi za to da bi se smanjila količina memorije (RAM-a) koju program ždere. Znači, ako vam treba varijabla za broj ljudi nekih, nećete koristiti tip single ili double jer vam ne trebaju decimale (ne može negdje biti 156 i pol ljudi!), pa ćete koristiti integer (ili, u nekim ekstremnim slučajevima, long. Pa koliko koji tip podataka zauzima memorije?

integer: 16 bitova (2 bajta)

long: 32 bita (4 bajta)

single: 32 bita (4 bajta)

double: 64 bitova (8 bajtova)

string: 8 bitova (1 bajt) po znaku, a pošto je dozvoljeno 256 znakova – 2048 bitova (2 kilobita (256 bajtova))

Zato se koristi "tip podataka" `STRING * n`. Ako znate broj znakova koji neki string smije sadržavati, koristite ovo `DIM a AS STRING * 10`

Ovime će se duljina niza ograničiti na 10 znakova. Ovaj se tip podataka koristi i za stringove duže od 256 znakova.

Pošto QBasic ne služi za "prave" programe, tipovi podataka služe samo za programe u kojima dobivate malo žešće rezultate. Primjer:

Znate onu priču kad je neki tip nekom Kinezu (ko zna zašto) rekao da hoće onoliko riže koliko bi bilo na šahovskoj ploči kad bi se na prvo polje stavilo 1 zrno, a na svako slijedeće polje duplo više zrna nego na prethodno? E, pa koliko bi na ploči bilo zrnja?

```
zbroj = 0
```

```
FOR i = 1 to 64
```

```
  x = 2 ^ (i - 1) ' polje 1: x = 2 ^ 0 (1); polje 2: x = 2 itd.
```

```
  zbroj = zbroj + x
```

```
NEXT i
```

```
PRINT zbroj
```

Kad bi varijabla **zbroj** bila tipa integer ili long, dobili biste grešku "Overflow". Da je single, dobili bi rezultat, ali malo jadan: 1.844674E+19 (1.844674 x 10¹⁹). Tipom double, dobivate pristojniju vrijednost: 1.844674407370955D+19 (1.844674407370955 x 10¹⁹).

Brojke i slova

Za tipove podataka važne su još dvije naredbe koje služe za pretvaranje iz brojevnog u znakovni tip i obrnuto. Recimo da u nekoj varijabli tipa string imate "234", i tome sad želite pribrojiti 1. To neće ići jednostavnom naredbom zbrajanja. Prvo vrijednost iz prve varijable ("234") morate

MLADI INFORMATIČARI STRAHONINCA



created by *Dejan Drabić*

prepisati u drugu, s tim da je ona druga brojčana (bilo koji brojčani tip). Za to služi naredba **VAL** (od eng. "value").

```
a$ = "234"
```

```
x% = VAL(a$)
```

Za obrnutu radnju koristi se naredba **STR** (od "string").

```
x% = 25
```

```
y$ = STR(x)
```



Matematika u QBasicu

OPERACIJA	OBJAŠNENJE	PRIMJER
+	zbrajanje	$3 + 2 = 5$
-	oduzimanje	$3 - 2 = 1$
*	množenje	$3 * 2 = 6$
/	dijeljenje	$3 / 2 = 1.5$
=	jednako / pridruživanje	$x = 3$
<	manje od	$x < 3$
>	veće od	$x > 3$
<=	manje ili jednako	$x <= 3$
>=	veće ili jednako	$x >= 3$
<>	nejednako	$x <> 3$
MOD	daje ostatak pri dijeljenju	$8 \text{ MOD } 5 = 3$
ABS(n)	apsolutna vrijednost broja n	$\text{ABS}(-6) = 6$
^	exponiranje [alt+94]	$2 \wedge 3 = 8$
SQR(n)	drugi (kvadratni) korijen broja n	$\text{SQR}(144) = 12$
SIN(n)	sinus kuta n u radijanima*	$\text{SIN}(\text{pi}) = 0$ **
COS(n)	kosinus kuta n u radijanima*	$\text{COS}(\text{pi}) = -1$ **
TAN(n)	tangens kuta n u radijanima*	$\text{TAN}(\text{pi}) = 0$ **
EXP(n)	e na n-tu	$\text{EXP}(1) = 2.71828$
LOG(n)	limes	$\text{LOG}(\text{EXP}(1)) = 1$
INT(n)	zaokružuje na prvi manji cijeli broj	$\text{INT}(-3.2) = -4$
FIX(n)	briše decimalni dio broja	$\text{FIX}(-3.2) = -3$
CINT(n)	zaokružuje na najbliži cijeli broj	$\text{CINT}(-3.2) = -3$

** Da biste pretvorili stupnjeve u radijane, pomnožite stupnjeve sa $(\text{pi} / 180)$.

** Npr. $\sin(90^\circ)$: $\text{SIN}(90 * (\text{pi} / 180))$

** Kad radite s brojem π , bilo bi dobro da na početku programa napišete

** $\text{pi} = 3.141592654$



ZADACI I RJEŠENJA

NAPOMENE:

- Ako ne znate neke matematičke postupke koji su vam potrebni u ovim zadacima (npr. ako još niste učili Pitagorin poučak), nemojte pokušavati to sad naučiti. Jedan zadatak više-manje...
- Ispis u vašem rješenju ne mora biti isti ni kao ispis u mom rješenju, ni kao primjer u zadatku. Glavno da je točno.
- Postoji više rješenja. Ako je vaše točno, a nije kao moje, nema nikakve potrebe da ga mijenjate!

ZADACI 1.-dio:

1.) Napišite program koji će unositi (naredba INPUT) duljine stranica pravokutnika i ispisivati (naredba PRINT) njegovu površinu!

Primjer:

Ulaz: Ispis:

2, 3 6

2.) Napišite program koji će unositi jedan prirodan broj i ispisivati njegov kvadrat!

Primjer:

Ulaz: Ispis:

11 121

3.) Napišite program koji će unositi dva prirodna broja i ispisivati njihov zbroj, razliku, umnožak i kvocjent!

Primjer:

Ulaz: Ispis:

10, 2 12, 8, 20, 5

4.) Napišite program koji unosi polumjere dviju koncentričnih kružnica (znači da imaju isto središte) (prvo polumjer manje pa veće) i ispisuje površinu kružnog vijenca između te dvije kružnice (vidi sliku)! (površina kruga: $r \pi$)

Primjer:

Ulaz: Ispis:

17, 19 226.08

5.) Napišite program koji će unositi koordinate gornjeg lijevog i donjeg desnog vrha pravokutnika (stranice su paralelne s osima x i y). Program treba ispisivati površinu i opseg tog pravokutnika. (Cijeli pravokutnik je u prvom kvadrantu, odnosno sve su vrijednosti pozitivne)

Primjer:

Ulaz: Ispis:

17, 19 12, 16

23,

21

6.) Napišite program koji vam govori koliko najviše kutijica stane u veliku kutiju. Sve kutijice su iste veličine i stavljaju se u istom smjeru. Program upisuje širinu, duljinu pa visinu kutijice i širinu, duljinu pa visinu velike kutije.



created by *Dejan Drabić*

NAPOMENA: za ovo vam treba funkcija INT. Recimo da je $a = 3.26$. Ako napišete $b = \text{INT}(a)$, b će postati prvi cijeli manji broj od a , dakle **3**. Toliko za sad.

Primjer:

Ulaz: Ispis:

3, 4, 2 20

15, 11, 5

7.) Kad završava 5. sat u školi? U program se upisuje vrijeme početka nastave, trajanje sata i odmora (između svakog sata je odmor, to valjda znate). Sva se vremena upisuju na način: sat i minute. Ispis je proizvoljan. Velikog odmora nema.

NAPOMENA: za ovo vam treba funkcija INT. Recimo da je $a = 3.26$. Ako napišete $b = \text{INT}(a)$, b će postati prvi cijeli manji broj od a , dakle **3**. Toliko za sad.

Primjer:

Ulaz: Ispis:

8, 0 12:05

0, 45

0, 5



created by *Dejan Drabić*

RJEŠENJA:

1.)

```
CLS
INPUT "Upiši stranice pravokutnika: ", a, b
PRINT "Površina pravokutnika je"; a * b; "."
```

2.)

```
CLS
INPUT "Upiši prirodan broj: ", a
PRINT "Kvadrat broja"; a; "je"; a * a; "."
```

3.)

```
CLS
INPUT "Upiši dva prirodna broja: ", a, b
PRINT "Zbroj:", a + b ' Ovdje sam stavljao
PRINT "Razlika:", a - b ' zarez, a ne točka-zarez
PRINT "Umnožak:", a * b ' da bi rezultati bilo lijepo
PRINT "Količnik:", a / b ' poredani jedan ispod drugog.
```

4.)

```
CLS
INPUT "Upiši polumjere kružnica: ", a, b
PolumjerA = a * a * 3.14 ' to je ernakvadratpi :)
PolumjerB = b * b * 3.14
PRINT "Površina vijenca je"; PolumjerB - PolumjerA
```




created by *Dejan Drabić*

5.)

```
CLS
INPUT "Gornji lijevi vrh: ", x1, y1
INPUT "Gornji desni vrh: ", x2, y2
DuljinaX = x2 - x1 'duljina stranice usporedne s x-osi
DuljinaY = y2 - y1 'duljina stranice usporedne s y-osi
PRINT "Površina je"; DuljinaX * DuljinaY; "."
PRINT "Opseg je"; 2 * (DuljinaX + DuljinaY); "."
```

6.)

```
CLS
INPUT "Širina, duljina i visina kutijice: ", a, b, c
INPUT "Širina, duljina i visina kutije: ", x, y, z
RezX = INT(x / a)
RezY = INT(y / b)
RezZ = INT(z / c)
PRINT "Broj kutijica: "; RezX * RezY * RezZ
```

7.)

```
CLS
INPUT "Početak nastave: ", PocH, PocM
INPUT "Trajanje sata: ", SatH, SatM
INPUT "Trajanje odmora: ", OdmH, OdmM
SatMIN = 60 * SatH + SatM 'trajanje sata u minutama
OdmMIN = 60 * OdmH + OdmM 'trajanje odmora u minutama
TrajMIN = 5 * (SatMIN + OdmMIN) - OdmMIN 'trajanje svih sati i odmora u minutama
'ima pet sati, poslije svakog je
'odmor, osim poslije zadnjeg.
TrajH = INT(TrajMIN / 60) 'koliko sati to sve skupa traje? Vjerojatno će
'ispasti decimalni broj, a treba nam broj sati
'(minute ćemo poslije), i zato naredba INT:
'da bi maknuli decimalni dio.
TrajM = TrajMIN - TrajH * 60 'Od ukupnih minuta oduzmemo sate i dobijemo
'koliko sati (TrajH) i minuta (TrajM)
'to sve traje.
DoH = PocH + TrajH
DoM = PocM + TrajM
PRINT "Peti sat završava u"; DoH; ":"; DoM
```

NAPOMENE:

- Ako ne znate neke matematičke postupke koji su vam potrebni u ovim zadacima (npr. ako još niste učili Pitagorin poučak), nemojte pokušavati to sad naučiti. Jedan zadatak više-manje...
- Ispis u vašem rješenju ne mora biti isti ni kao ispis u mom rješenju, ni kao primjer u zadatku. Glavno da je točno.
- Postoji više rješenja. Ako je vaše točno, a nije kao moje, nema nikakve potrebe da ga mijenjate!

ZADACI:

1.) Napišite program koji će unositi (naredba INPUT) duljine stranica pravokutnika i ispisivati (naredba PRINT) njegovu površinu!



created by *Dejan Drabić*

Primjer:

Ulaz: Ispis:

2, 3 6

2.) Napišite program koji će unositi jedan prirodan broj i ispisivati njegov kvadrat!

Primjer:

Ulaz: Ispis:

11 121

3.) Napišite program koji će unositi dva prirodna broja i ispisivati njihov zbroj, razliku, umnožak i kvocjent!

Primjer:

Ulaz: Ispis:

10, 2 12, 8, 20, 5

4.) Napišite program koji unosi polumjere dviju koncentričnih kružnica (znači da imaju isto središte) (prvo polumjer manje pa veće) i ispisuje površinu kružnog vijenca između te dvije kružnice (vidi sliku)! (površina kruga: $r^2 \pi$)

Primjer:

Ulaz: Ispis:

17, 19 226.08

5.) Napišite program koji će unositi koordinate gornjeg lijevog i donjeg desnog vrha pravokutnika (stranice su paralelne s osima x i y). Program treba ispisivati površinu i opseg tog pravokutnika. (Cijeli pravokutnik je u prvom kvadrantu, odnosno sve su vrijednosti pozitivne)

Primjer:

Ulaz: Ispis:

17, 19 12, 16

23,

21



created by *Dejan Drabić*

6.) Napišite program koji vam govori koliko najviše kutijica stane u veliku kutiju. Sve kutijice su iste veličine i stavljaju se u istom smjeru. Program upisuje širinu, duljinu pa visinu kutijice i širinu, duljinu pa visinu velike kutije.

NAPOMENA: za ovo vam treba funkcija INT. Recimo da je $a = 3.26$. Ako napišete $b = \text{INT}(a)$, b će postati prvi cijeli manji broj od a , dakle **3**. Toliko za sad.

Primjer:

Ulaz: Ispis:

3, 4, 2 20

15, 11, 5

7.) Kad završava 5. sat u školi? U program se upisuje vrijeme početka nastave, trajanje sata i odmora (između svakog sata je odmor, to valjda znate). Sva se vremena upisuju na način: sati , minute. Ispis je proizvoljan. Velikog odmora nema.

NAPOMENA: za ovo vam treba funkcija INT. Recimo da je $a = 3.26$. Ako napišete $b = \text{INT}(a)$, b će postati prvi cijeli manji broj od a , dakle **3**. Toliko za sad.

Primjer:

Ulaz: Ispis:

8, 0 12:05

0, 45

0, 5

RJEŠENJA:

1.)

CLS

```
INPUT "Upiši stranice pravokutnika: ", a, b
PRINT "Površina pravokutnika je"; a * b; "."
```

2.)

CLS

```
INPUT "Upiši prirodan broj: ", a
PRINT "Kvadrat broja"; a; "je"; a * a; "."
```

3.)

CLS

```
INPUT "Upiši dva prirodna broja: ", a, b
PRINT "Zbroj:", a + b ' Ovdje sam stavljao
PRINT "Razlika:", a - b ' zarez, a ne točka-zarez
PRINT "Umnožak:", a * b ' da bi rezultati bilo lijepo
PRINT "Količnik:", a / b ' poredani jedan ispod drugog.
```

4.)

CLS

```
INPUT "Upiši polumjere kružnica: ", a, b
PolumjerA = a * a * 3.14 ' to je ernakvadratpi :)
PolumjerB = b * b * 3.14
PRINT "Površina vijenca je"; PolumjerB - PolumjerA
```

5.)

CLS

```
INPUT "Gornji lijevi vrh: ", x1, y1
INPUT "Gornji desni vrh: ", x2, y2
```



created by *Dejan Drabić*

```
DuljinaX = x2 - x1 'duljina stranice usporedne s x-osi
DuljinaY = y2 - y1 'duljina stranice usporedne s y-osi
PRINT "Površina je"; DuljinaX * DuljinaY; "."
PRINT "Opseg je"; 2 * (DuljinaX + DuljinaY); "."
```

6.)

```
CLS
INPUT "Širina, duljina i visina kutijice: ", a, b, c
INPUT "Širina, duljina i visina kutije: ", x, y, z
RezX = INT(x / a)
RezY = INT(y / b)
RezZ = INT(z / c)
PRINT "Broj kutijica: "; RezX * RezY * RezZ
```

7.)

```
CLS
INPUT "Početak nastave: ", PocH, PocM
INPUT "Trajanje sata: ", SatH, SatM
INPUT "Trajanje odmora: ", OdmH, OdmM
SatMIN = 60 * SatH + SatM 'trajanje sata u minutama
OdmMIN = 60 * OdmH + OdmM 'trajanje odmora u minutama
TrajMIN = 5 * (SatMIN + OdmMIN) - OdmMIN 'trajanje svih sati i odmora u
minutama
'ima pet sati, poslije svakog je
'odmor, osim poslije zadnjeg.
TrajH = INT(TrajMIN / 60) 'koliko sati to sve skupa traje? Vjerojatno će
'ispasti decimalni broj, a treba nam broj sati
'(minute ćemo poslije), i zato naredba INT:
'da bi maknuli decimalni dio.
TrajM = TrajMIN - TrajH * 60 'Od ukupnih minuta oduzmemo sate i dobijemo
'koliko sati (TrajH) i minuta (TrajM)
'to sve traje.
DoH = PocH + TrajH
DoM = PocM + TrajM
PRINT "Peti sat završava u"; DoH; ":"; DoM
```



created by *Dejan Drabić*

ZADACI 2.-dio:

1.) Napišite program koji će unositi dva cijela broja i ispisati veći.

Primjer:

Ulaz: Ispis:

2, 3 3

2.) Napišite program koji unosi prirodan broj i ispisuje je li djeljiv s 3, 5, 7 ili 9.

Primjer:

Ulaz: Ispis:

63 Broj je djeljiv s: 3, 7, 9

3.) Napišite program koji unosi godinu (poslije Krista) i ispisuje je li ta godina bila (će biti) prijestupna. Prijestupna je ako je djeljiva sa 4 ili 400, a nije djeljiva sa 100.

Primjer:

Ulaz: Ispis:

1997 Nije prijestupna.

2000 Prijestupna.

1900 Nije prijestupna

4.) Program za bankomat. Zatraženi iznos treba isplatiti u što manjem broju novčanica. Recimo da bankomat ima neograničen broj novčanica od 100, 50, 10 i 1 kn. (Nema veze što ne postoje novčanice od 1 kn, šta cjepidlačite!) Program ispisuje koliko kojih novčanica treba ispljunuti.

Primjer:

Ulaz: Ispis:

573 5 x 100, 1 x 50, 2 x 10, 3 x 1

5.) Napišite program koji će učitavati prirodan broj i ispisivati je li taj broj kvadrat nekog drugog prirodnog broja i kojeg!

Primjer:

Ulaz: Ispis:

16 Broj 16 je kvadrat broja 4.

24 Broj 24 nije kvadrat nijednog prirodnog broja.

6.) Program unosi dva skupa (na pravcu) i ispisuje njihov presjek. Ukoliko je presjek prazan skup, to treba napisati. Svi brojevi moraju biti različiti.

Primjer:

Ulaz: Ispis:

-15, 10 0, 10

0, 23

7.) Program upisuje tri broja (koja ne moraju biti poslagana po veličini), provjerava jesu li oni članovi nekog aritmetičkog niza (je li razlika između 2 susjedna broja uvijek jednaka) i ispisuje sljedeći član!

Primjer:

Ulaz: Ispis:

5, 3, 7 9

0, -2, 1 Brojevi nisu članovi aritmetičkog niza.



created by *Dejan Drabić*

5, -85, -40 50

RJEŠENJA:

1.)

```
CLS
INPUT "Upiši 2 cijela broja: ", x, y
IF x > y THEN PRINT x
IF y > x THEN PRINT y
```

2.)

```
CLS
INPUT "Upiši prirodan broj: ", x
PRINT "Broj"; x; "je djeljiv s ";
IF x MOD 3 = 0 THEN PRINT "3 ";
IF x MOD 5 = 0 THEN PRINT "5 ";
IF x MOD 7 = 0 THEN PRINT "7 ";
IF x MOD 9 = 0 THEN PRINT "9 ";
```

3.)

```
CLS
INPUT "Upiši godinu: ", x
IF x MOD 4 = 0 AND x MOD 100 <> 0 THEN
PRINT "Prijestupna."
ELSEIF x MOD 400 = 0 THEN
PRINT "Prijestupna."
ELSE
PRINT "Nije prijestupna."
END
```

IF

4.)

```
CLS
INPUT "Iznos za isplatu: ", x
od100 = INT(x / 100)
x = x MOD 100
od50 = INT(x / 50)
x = x MOD 50
od10 = INT(x / 10)
x = x MOD 10
od1 = INT(x)
PRINT od100; "x 100,"; od50; "x 50,"; od10; "x 10,"; od1; "x 1."
```

5.)

```
CLS
INPUT "Prirodan broj: ", x
IF SQR(x) = INT(SQR(x)) THEN '(tj. ako je SQR(x) cijeli broj)
PRINT "Broj"; x; "je kvadrat broja"; SQR(x); "."
ELSE
PRINT "Broj"; x; "nije kvadrat nijednog prirodnog broja."
END IF
```

6.)

```
CLS
INPUT "Raspon prvog skupa: ", a, b
INPUT "Raspon drugog skupa: ", c, d
IF a < d AND b < d AND a > c AND b > c THEN '---C---a===b---D---
```



created by *Dejan Drabić*

```
PRINT "<"; a; ", "; b; ">"
ELSEIF c < b AND d < b AND c > a AND d > a THEN '---a---C===D---b---
PRINT "<"; c; ", "; d; ">"
ELSEIF a < d AND a > c THEN '---C---a===D---b---
PRINT "<"; a; ", "; d; ">"
ELSEIF b < d AND b > c THEN '---a---C===b---D---
PRINT "<"; c; ", "; b; ">"
ELSEIF (a < c AND b < c) OR (a > d AND b > d) THEN '---a---b---C---D---
PRINT "Presjek je prazan skup." ' ili
END                                IF                                '---C---D---a---b---
```



created by *Dejan Drabić*

7.)

```
CLS
INPUT "Upiši tri broja: ", a, b, c
'-----> SORTIRANJE:
IF a > b THEN
x = b ' zamjena
b = a ' a sa b
a = x ' b postaje a, a postaje b
END IF
IF b > c THEN
x = c ' b je sad veći od a,
c = b ' c je možda manji od b?
b = x ' (opet zamjena)
END IF
IF a > b THEN
x = b ' sad je c 100% najveći,
b = a ' još treba složiti
a = x ' a i b.
END IF
'-----> ARITMETIČKI NIZ?
IF b - a = c - b THEN ' ta 3 broja su članovi niza
PRINT c + (c - b) ' ispisuje broj veći od zadnjeg za razliku
ELSE
PRINT "Brojevi nisu članovi aritmetičkog niza."
END IF
```




created by Dejan Drabić

ZADACI 3.-dio

1.) Napišite program koji će ispisivati svaki x-ti broj počevši s brojem y i završivši s brojem z (ili prvim manjim koji je član niza)!

Primjer:

Ulaz: Ispis:

4, 11, 25 11, 15, 19, 23

2.) Napišite program koji učitava prirodan broj i ispisuje sve prirodne brojeve manje od njega koji imaju cjelobrojne korjene!

Primjer:

Ulaz: Ispis:

95 1, 4, 9, 16, 25, 36, 49, 64, 81

3.) Napišite program koji računa izraz $xy!$ (Pomoću petlje, bez operacije $^$!!!)

Primjer:

Ulaz: Ispis:

5, 3 125

4.) Program ispisuje je li učitani broj prost ili ne. (prosti brojevi su djeljivi samo sa 1 i sa samim sobom; broj 1 nije prost broj)

Primjer:

Ulaz: Ispis:

1 Ne.

7 Da.

29 Da.

5.) Napišite program koji ispisuje sve proste brojeve do učitano broj! (prosti brojevi su djeljivi samo sa 1 i sa samim sobom; broj 1 nije prost broj)

Primjer:

Ulaz: Ispis:

30 2, 3, 5, 7, 11, 13, 17, 19, 23, 29

6.) Napišite program koji će ispisivati najveći prirodan broj čiji je kvadrat manji od upisanog broja!

Primjer:

Ulaz: Ispis:

43 6

7.) Program učitava n prirodnih brojeva i ispisuje broj i zbroj parnih i broj i zbroj neparnih.

Primjer:

Ulaz: Ispis:

4

17, 6, -5, 2 Parnih: 2, zbroj: 8; neparnih: 2, zbroj: 12

8.) Napišite program koji unosi prva dva člana nekog geometrijskog niza i ispisuje prvih n članova tog niza. [U geometrijskom nizu se svaki sljedeći član dobije tako da se onaj prošli pomnoži s nekim brojem (svaki put istim). Npr. 1, 2, 4, 8, 16, 32... (množi se s 2) ili 5, 55, 605, 6655, 73205... (množi se s 11).]



created by *Dejan Drabić*

Primjer:

Ulaz: Ispis:

1, 3 1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19683

10

9.) Program učitava brojeve (od -9999 do 9999) dok se ne upiše 0 i svaki put poslije upisa ispisuje trenutno najmanji i najveći broj. Na kraju ispisuje zbroj svih upisanih brojeva.

Primjer:

Ulaz: Ispis:

10 min.: 10, max.: 10

7 min.: 7, max.: 10

15 min.: 7, max.: 15

-50 min.: -50, max.: 15

0 zbroj: -18

10.) Program unosi prirodne brojeve dok se ne unese kvadrat nekog prirodnog broja. Zatim ispisuje zbroj svih unesenih brojeva, njihov broj (koliko ih je upisano) i kojeg je prirodnog broja zadnji broj kvadrat.

Primjer:

Ulaz: Ispis:

43, 7, 2, 25 zbroj: 77, ukupno: 4, 25 je kvadrat od 5.

11.) Računalo "zamisli" jedan broj između 1 i 100. Korisnik upisuje jedan po jedan broj, a računalo mu sipisuje je li učitani broj manji ili veći od "zamišljenog". Program završava kad korisnik pogodi broj i ispisuje koliko mu je pokušaja trebalo.

NAPOMENA: Za "zamišljanje" broja: na početku programa (poslije CLS) napišite RANDOMIZE TIMER. Poslije možete napisati: $x = \text{INT}(\text{RND} * n) + 1$ i x će postati slučajni cijeli broj između 1 i n .

Primjer:

Ulaz: Ispis:

40 Broj je veći.

70 Broj je veći.

90 Broj je manji.

80 Broj je veći.

83 Broj je manji.

81 Pogodio/la si u 6 pokušaja!!!

12.) Ušli ste u potpuno mračnu sobu upisane širine i duljine. Računalo slučajnim odabirom određuje koordinate svjetiljke. Na početku ste na položaju (1, 1) – dolje lijevo. Krećete se brojevima (4 – lijevo, 6 – desno, 8 – gore, 2 – dolje). Poslije svakog koraka, računalo vam ispisuje vaš trenutacan položaj. Kad pritisnete tipku 5, ispisuje u kojem smijeru (lijevo/desno i gore/dolje) je svjetiljka. Ako udarite u zid, računalo vam to mora napisati i **ne dozvoliti vam** da idete kroz zidove. Na kraju (kad dođete do svjetiljke) računalo mora ispisati koliko vam je koraka trebalo, te koliko ste puta pritisnuli tipku 5.



created by *Dejan Drabić*

NAPOMENA: Za "zamišljanje" broja: na početku programa (poslije CLS) napišite RANDOMIZE TIMER. Poslije možete napisati: $x = \text{INT}(\text{RND}*n)+1$ i x će postati slučajni cijeli broj između 1 i n .

Primjer:

Ulaz: Ispis:

6, 5

4 Nemoš kroz zidove!

6 Sad si na (2, 1).

8 Sad si na (2, 2).

5 Svjetiljka je desno gore.

6 Sad si na (3, 2).

5 Svjetiljka je gore.

8 Sad si na (3, 3)

8 Sad si na (3, 4)

Našao si svjetiljku u 5 koraka! 2 puta si stisnuo 5.

1 2 3 4 5 6

5
4
3
2
1



created by *Dejan Drabić*

RJEŠENJA:

1.)

```
CLS
INPUT "Svaki koji broj? ", a
INPUT "Od ", b
INPUT "Do ", c
FOR i = b TO c STEP a
PRINT i;
NEXT i
```

2.)

```
CLS
INPUT "Upiši broj: ", x
FOR i = 1 TO x
IF SQR(i) = INT(SQR(i)) THEN PRINT i; 'Ako je decimalna vrijednost broja
'jednaka zaokruženoj, broj je cijeli.
NEXT i
```

3.)

```
CLS
INPUT "Broj ", x
INPUT "Na potenciju ", y
rez = x
FOR i = 2 TO y
rez = rez * x
NEXT i
PRINT rez
```

4.)

```
CLS
INPUT "Upiši broj: ", x
IF x = 1 THEN
PRINT "Broj"; x; "nije prost."
GOTO 10
END IF
FOR i = 2 TO x / 2
IF x MOD i = 0 THEN
PRINT "Broj"; x; "nije prost."
GOTO 10
END IF
NEXT i
PRINT "Broj"; x; "je prost."
10 END
```

5.)

```
CLS
INPUT "Upiši broj: ", x
FOR i = 2 TO x
FOR j = 2 TO i / 2
IF i MOD j = 0 THEN GOTO 10
NEXT j
PRINT i;
10
NEXT i
```

6.)

```
CLS
INPUT "Upiši broj: ", x
```



created by *Dejan Drabić*

```
FOR i = x - 1 TO 1 STEP -1
IF INT(SQR(i)) = SQR(i) THEN 'Ako je decimalna vrijednost broja jednaka
PRINT SQR(i) 'zaokruženoj, broj je cijeli.
GOTO 10
END IF
NEXT i
10 END
```

7.)

```
CLS
INPUT "Koliko brojeva? ", n
bp = 0 'broj parnih
zp = 0 'zbroj parnih
bn = 0 'broj neparnih
zn = 0 'zbroj neparnih
FOR i = 1 TO n
INPUT "Upiši broj: ", x
IF x MOD 2 = 0 THEN
bp = bp + 1
zp = zp + x
ELSE
bn = bn + 1
zn = zn + x
END IF
NEXT i
PRINT " Parnih: "; bp; "; ";
PRINT "Zbroj: "; zp
PRINT "Neparnih: "; bn; "; ";
PRINT "Zbroj: "; zn
```

8.)

```
CLS
INPUT "Prva 2 člana niza: ", x, y
INPUT "Koliko da članova ispišem? ", c
koef = y / x 'koeficijent - broj s kojim se množi svaki sljedeći član.
PRINT x; 'jer će, kad je i=1, program ispisati drugi član niza.
tren = x 'trenutačni član
FOR i = 1 TO c - 1
tren = tren * koef 'trenutačni član se množi s koeficijentom
PRINT tren; 'i dobije se novi član, koji se ispiše.
NEXT i
```

9.)

```
CLS
min = 9999 ' granice postavljene
max = -9999 ' u zadatku
zbroj = 0
DO
INPUT "Upiši broj: ", x
IF x <> 0 THEN
IF x < min THEN min = x
IF x > max THEN max = x
PRINT "min: "; min; ", max: "; max
PRINT ' čisto estetski, da ima red razmaka...
END IF
zbroj = zbroj + x
LOOP UNTIL x = 0
PRINT zbroj
```



created by *Dejan Drabić*

10.)

```
CLS
zbroj = 0
broj = 0
DO
broj = broj + 1
INPUT "Upiši broj: ", x
zbroj = zbroj + x
LOOP UNTIL INT(SQR(x)) = SQR(x) 'Ako je decimalna vrijednost broja jednaka
'zaokruženoj, broj je cijeli.
PRINT "Zbroj: "; zbroj
PRINT "Ukupno: "; broj
PRINT x; "je kvadrat od"; SQR(x); "."
```

11.)

```
CLS
RANDOMIZE TIMER
x = INT(RND * 100) + 1 '"zamišljeni" broj
'x = 81
pok = 0 'broj pokušaja
DO
pok = pok + 1
INPUT "Pogodi broj! ", a
IF a < x THEN
PRINT "Broj je veći."
ELSEIF a > x THEN
PRINT "Broj je manji."
END IF
PRINT
LOOP UNTIL a = x
PRINT "Pogodio/la si u"; pok; "pokušaja!!!"
```

12.)

```
CLS
RANDOMIZE TIMER
INPUT "Širina? ", UkupX
INPUT "Duljina? ", UkupY
PRINT
PRINT
TrenX = 1 'Trenutačni položaj
TrenY = 1
Korak = 0 'Broj koraka
Pet = 0 'Ovoliko puta je stisnuta petica.
SvjX = INT(RND * UkupX) + 1
SvjY = INT(RND * UkupY) + 1
SvjX = 3
SvjY = 4
DO
INPUT "Kuda? ", a
SELECT CASE a
CASE 4 '-----Lijevo
IF TrenX = 1 THEN
PRINT "Nemoš kroz zidove!!!"
PRINT
GOTO 10
ELSE
TrenX = TrenX - 1
```



created by *Dejan Drabić*

```
Korak = Korak + 1
PRINT "Sad si na ("; TrenX; ","; TrenY; ")."
PRINT
END IF
CASE 6 '-----Desno
IF TrenX = UkupX THEN
PRINT "Nemoš kroz zidove!!!"
PRINT
GOTO 10
ELSE
TrenX = TrenX + 1
Korak = Korak + 1
PRINT "Sad si na ("; TrenX; ","; TrenY; ")."
PRINT
END IF
CASE 8 '-----Gore
IF TrenY = UkupY THEN
PRINT "Nemoš kroz zidove!!!"
PRINT
GOTO 10
ELSE
TrenY = TrenY + 1
Korak = Korak + 1
PRINT "Sad si na ("; TrenX; ","; TrenY; ")."
PRINT
END IF
CASE 2 '-----Dolje
IF Tren >= 1 THEN
PRINT "Nemoš kroz zidove!!!"
PRINT
GOTO 10
ELSE
TrenY = TrenY - 1
Korak = Korak + 1
PRINT "Sad si na ("; TrenX; ","; TrenY; ")."
PRINT
END IF
CASE 5 '-----Gdje je?
Pet = Pet + 1
PRINT "Svjetiljka je ";
IF TrenX > SvjX THEN
PRINT "lijevo ";
ELSEIF TrenX < SvjX THEN
PRINT "desno ";
END IF
IF TrenY > SvjY THEN
PRINT "dolje";
ELSEIF TrenY < SvjY THEN
PRINT "gore";
END IF
PRINT "."
PRINT
CASE ELSE
PRINT "4- lijevo, 6- desno, 8- gore, 2- dolje, 5- položaj svjetiljke"
END SELECT
10
```



created by *Dejan Drabić*

```
LOOP UNTIL TrenX = SvjX AND TrenY = SvjY
PRINT
PRINT "Našao si svjetiljku u"; Korak; "koraka!"; Pet; "puta si stisnuo 5."
```