

# ORDENADORE PERTSONALAK PROGRAMATZEN

JESUS M. ITURRIOTZ

Eskola irakasteko programa bat idaztea ez bazara inoiz ausartu, oraingoan, artikulua hau irakurtzen baduzu, ez dut uste inolako eragozpenik izango duzunik.

Zer behar dugu *maixuaren lana* egiten duen programa bat prestatzeko?

- Lehenengo *Lezioa* irakatsi. (ordenadoreari)
- Bigarren *Lezioa hartzen* trebatu (ordenadorea)
- Hirugarren *malizia* pixka batez hornitu. (ordenadorea).

Hauexek dira bada, programa batek behar dituen osagaiak, eskola-lanak egiten eta irakasten laguntzeko.

Has gaitezen bada, atzetik aurrera goian aipatutako hiru atalak banan-banan aztertzen.

Maliziaz ari naizenean zera adierazi nahi dut; ordenadoreak, galderak egiten hasten denean, sekuentzia bat jarraitzen du. Sekuentzia luzea ala motza izan daiteke, baina, sekuentzia pixka bat jardun ondoren berehalaxe somatuko dugu zein galdera egingo digun hurrengoan eta gainera, erantzun baten ondoren beste erantzun finkoa datorrela.

Beraz, eragozpen hau gainditzeko, ordenadoreak aleatorioki egin beharko dizkigu galderak. Era aleatorio edo zorizko hau lortzeko ordenadoreek duten RND funtzioa erabiltzen da. Funtzio honen emaitza (ez du argumenturik behar) 0 eta 1 arteko zenbaki aleatorioa da beti (7

hamarreneko zenbakia), nahiz eta batzutan emaitza 0 izan inola ere ez da 1 izango, gehienez 0'9999999.

Hara:

ZX SPECTRUM	COMMODORE 64
10 PRINT RND	10 PRINT RND (1)
20 GOTO 10	20 GOTO 10

IBM-PC
10 PRINT RND
20 GOTO 10

0 eta 1 arteko zenbaki aleatorioak baino handiagoak lortu nahi badituzu, orduan RND funtzioa goi-mugaren balioarekin biderkatu behar duzu.

ZX SPECTRUM	COMMODORE 64
10 PRINT 5*RND	10 PRINT 5*RND (1)
20 GOTO 10	20 GOTO 10

IBM-PC
10 PRINT 5*RND
20 GOTO 10

Bestalde, behe-muga ezartzeko zera egin behar duzu; funtzio aleato-

rioak ematen duen emaitzari, beti, behe-mugaren balioa gehitu. Baina, kontutan eduki, batuketa emaitza guztiekin egiten dela. Beraz, goi-mugaren balioa ere kontrolatu nahi badugu formula hau jarraitu beharko dugu.

$$(RND*B) + BM$$

RND = zorizko zenbaki hamar-tarrak lortzeko funtzioa

B = biderkatzailea

BM = behe-muga

GM = goi-muga

Behe-muga zero denean biderkatzailearen balioa goi-mugarena izango da. Baina zero ez denean, biderkatzailearen balioa hauxe da

$$B = GM - BM + 1$$

Adibide batekin argituko dugu orain arte esandakoa.

Pentsa ezazu ordenadorea dadoak botatzen jarri nahi duzula. Horretarako, balio ote du ondoko programa honek?

Dadoaren mugak 6 eta 1 dira beraz biderkatzailearen balioa

$$B = GM - BM + 1 = 6 - 1 + 1 = 6$$

ZX SPECTRUM	COMMODORE 64	IBM-PC
10 PRINT (RND*6) + 1	10 PRINT (RND (1)*6) + 1	10 PRINT (RND*6) + 1
20 PAUSE 70	20 FOR I = 0 TO 300	20 FOR I = 0 TO 300
30 GOTO 10	25 NEXT I	25 NEXT I
	30 GOTO 10	30 GOTO 10

Dagoeneko, programatxo hau zure ordenadorean sartu baldin baduzu, ohartuko zinen oso dado xelebrea lortu dugula.

2.9474182 geldialditxo eta  
 3.0609436 " "  
 1.2273254 " "

ETAB.

Zer egin bada zenbakien hamarrenak ezabatzeko? Horretarako ordenadoreek duten INT funtzioa erabiltzen da. Hau da, aginte honek zenbakien zati osoa bakarrik onartzen du, hamarren guztiak baztertuz.

Kontuz! Baztertu esan bait dut eta horrek suposatzen du  $6 * \text{RND}$  (edo  $6 * \text{RND}(1)$ ) funtzioarekin lor dezakegun zenbaki handiena hau dela

$$6 * 0,9999999 = 5,9999994 = 6$$

Baldin, orain INT aginteaz hamarrenak baztertzen badizkiogu 5 gelditzen zaigu eta txikiena 0. Baina dadoak 1 eta 6 arteko zenbakiak ditu. Horregatik gehitzen diogu unitate bat biderkatzaileari zenbaki osoak nahi ditugunean.  $\text{INT}(6 * \text{RND}) + 1$

Goazen bada, aurrera jarraitzea. Ordenadoreari *lezioa* irakastera.

ZX SPECTRUM	COMMODORE 64	IBM-PC
10 PRINT INT(RND*6) + 1	10 PRINT INT(RND (1)*6) + 1	10 PRINT INT(RND*6) + 1
20 PAUSE 70	20 FOR I = 0 TO 300	20 FOR I = 0 TO 300
30 GOTO 10	25 NEXT I	25 NEXT I
	30 GOTO 10	30 GOTO 10

Ohurra: 20 eta 25. lineak geldialditxo lortzeko dira *SPECTRUM*ak zuzeneko instrukzioa dauka PAUSE 70 (50 gutxi gorabehera segundo bat da). Besteetan aldiz zeharreko bi instrukzio erabiltzen dira.

Horretarako **DATA** instrukzioa erabiltzen dugu; bertan galdera eta erantzuna jarriaz.

Adibidez geografiazko programa-txo bat antolatu nahi dugu eta ordenadoreari Euskalerriko eskualde desberdinetako hiri-buruen izenak irakatsi nahi dizkiogu.

1000 DATA IRUINEA, NAFARROA, DONOSTIA, GIPUZKOA, BILBO, BIZKAIA, DONIBANE GARAZI, BEHENA-FARROA, BAIONA, LAPURDI, GAS-TEIZ, ARABA, MAULE, ZUBEROA.

edo gramatika irakatsi nahi diogu, adibidez NOR NORI NORK aditz laguntzailearen orainaldia.

1000 DATA AITAK AMARI GONA GORRIA EKARRI, DIO,

1005 DATA NIK (ZURI) ILEA BUSTI, DIZUT,...

edo Ohm-en legea irakatsi nahi diogu

1000 DATA I \* R, V, V / R, I, V / I, R

Ikusten da beraz, edozein gai irakats diezaiokegula ordenadoreari.

Behar bada zailen gerta litekeena ondoren datorrena da:

Ordenadorea *lezioa hartzen* trebatu, eta gainera erantzunak onak ala txarrak diren bereizten jakiteko ahalmenaz hornitu. Hala nola, baldin eta, erantzunak txarrak direnean,

asmatzeko aukera mugatu eta ez bada asmatzen, erantzun ona eman.

*Lezioa hartzeko READ* instrukzioa erabiliko dugu. Instrukzio honek **DATA** saileko datuak irakurtzen ditu. Datu hauek memorian gordetzen dira etiketa berezi batzuekin eta edozein garaian eska diezaiokegu etiketa berezietatik gordetako datuak pantailan azaltzeko edo teklatutik sartutako beste datu batekin gonbaratzeko eta abar.

Goian aipatutako etiketak alfabetoko letra batez edo gehiagoz osatzen dira eta izendatu behar duten datua zenbakizkoa ez bada \$ ikurrak laguntzen dira; adibidez:

A\$ → DONOSTIA

B → 3457

DF\$ → ELHUYAR

Datuak, ordenadoreak onartzeko, koma batez separaturik ipini behar dira guztietan, baina *COMMODORE 64*-an ezin dira lerro bat baino gehiagoko **DATA**rik sartu. Aldiz *SPECTRUM*ak nahi adina datuko **DATA** onartzen du baina zenbakizko datuak ez direla azaltzeko, hauek komatxo (") artean jarri behar dira. *IBM PC* izango da behar bada hiruetatik erosoena ez bait du komatxorik behar letrazko datuak sartzeko eta gainera **DATA** lerroen luzera ez dago mugaturik.

## Adibideak:

### ZX SPECTRUM

```
80 READ A$,B
1000 DATA "DONOSTIA", 4, "IRUI-
NEA", 6,.....
....., "ELGOIBAR", 4
```

### COMMODORE 64

```
80 READ A$,B
1000 DATA DONOSTIA, 4, IRUINEA, 6,
BILBO, 3
1005 DATA.....
ELGOIBAR,4
```

### IBM PC

```
80 READ A$,B
1000 DATA DONOSTIA, 4, IRUINEA, 6,
....., ELGOIBAR,4
```

Galdera egiteko **INPUT F** erabiliko dugu, ondoren **B** etiketapean dagoen datuarekin gonbaratu eta gonbaraketaren arabera bide bat edo bestea aukeratuko dugu.

Hara:

### SPECTRUMan

```
60 INPUT "NON DAGO";A$;"?";F
80 IF F<>B THEN CLS: PRINT
"GAIZKI, APROBA EZAZU BERRIZ":
PAUSE 150: GOTO 60
85 IF F=B THEN CLS: PRINT "OSO
ONDO": PAUSE 150: GOTO 20
```

### COMMODORE 64

```
60 INPUT "NON DAGO";A$;"?";F
80 IF F<>B THEN PRINT CHR$(147):
PRINT "GAIZKI, APROBA EZAZU
BERRIZ": FOR I=0 TO 500: NEXT I:
GOTO 60
85 IF F=B THEN PRINT CHR$(147):
PRINT "OSO ONDO": FOR I=0 TO
500: NEXT I: GOTO 20
```

### IBM PC

```
60 INPUT "NON DAGO";A$;"?";F
80 IF F<>B THEN CLS: PRINT
"GAIZKI, APROBA EZAZU BERRIZ":
FOR I=0 TO 500: NEXT I: GOTO 60
85 IF F=B THEN CLS: PRINT "OSO
ONDO": FOR I=0 TO 500: NEXT I:
GOTO 20
```

Goazen bada haseran aipatutako programa antolatzena. Esan dugun bezala hiru zatitan banatzen da eta ia ez da aldaketarik egin behar gai desberdinak tratatzeko. Noski, esan beharrik ere ez dago behar-beharrezko aldaketa **DATA** saillekoa dela. Beste aldaketa nabarmena **60** lerroan dago. Hau da, galderaren presentazioa **PRINT** batez antolatuko dugu eta honekin batera, erantzun posibleak; ahal dela, gainera, zenbakizko ordainekin; horrela, tekleatzeko garaian datuak oker sartzeko arriskua izugarri gutxituaz.

Erantzun-aukerarik ez balego edo hobeki esanda, erantzunak ez baleude bitara, bostera edo... mugaturik orduan ez genuke erantzun posibleak aurkezteko aukerarik izango eta

gainera, karaktere-kate batez sartu beharko genuke erantzuna.

Jar dezagun adibide bat, azken hau errazago ulertzeko:

Demagun hau galdetu nahi dugula:

"(gaur) Nere aitak amari gona gorria ekarri"....?

"DIO" da erantzuna eta ez dirudi aproposa denik beste erantzun batzukin nahastea

- ZION = 1
- NIOKE = 2
- DIO = 3
- .
- .
- DIE = n

Baina, baldin, geografia irakasten, Euskalerriko herri edo mendi nagusienak landu nahi baditugu honako galdera batez antolatuko dugu:

"NON DAGO DONOSTIA"....?

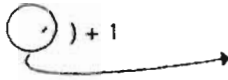
- ARABA = 1
- BEHENA = 2
- BIZKAIA = 3
- GIPUZKOA = 4
- LAPURDI = 5
- NAFARROA = 6
- ZUBEROA = 7

Bukatzeko zera esango dut. Ez ahaztu 40. lerroko **READ A\$, B** etiketak; batak letrak gordetzen ditu eta bigarrenak zenbakiak. Beraz etiketak datuen araberrakoak behar dutela izan ez ahaztu.

### PROGRAMA STANDARRA

```
10 REM "ikasgaiaren izena"
20 RESTORE:LET C = INT(RND*
30 FOR I = 1 TO C40 READ A$, B
50 NEXT I
60 PRINT "gaiari dagokion galdera"; A$; "?"
61 PRINT AT 4, 18; "Erantzuna = 1" ..... SPEKTRUMan
61 PRINT ,, "Erantzuna : 1" ..... COMMODORE 64
61 PRINT ..... "Erantzuna = 1" ..... IBM-PC
62 PRINT

70 INPUT F
```



Zenbaki hau DATAko galdera-erantzun kopuruak ematen du.

```

80 IF F <> B THEN .....
85 IF F = B THEN .....
1000 DATA ..., ..., ..., ...
1005 DATA
1010 DATA

```

} ikus 5. orrialdea

**Adibideak:**

### SPECTRUM

```
10 REM "GEOGRAFIA IKASTEN J M. ITURRIOZ 1985 ☉"
```

```
20 RESTORE: LET C = INT(RND*34) + 1
```

```
30 FOR I = 1 TO C
```

```
40 READ A$, B
```

```
50 NEXT I
```

```
60 PRINT "NON DAGO ; A$; "?"
```

←----- espazioak

```
61 PRINT AT 4, 18, "ARABA" = 1"
```

```
62 PRINT AT 5, 18; "BEHENAFARROA" = 2"
```

```
63 PRINT AT 6, 18; "BIZKAIA" = 3"
```

```
64 PRINT AT 7, 18, "GIPUZKOA" = 4"
```

```
65 PRINT AT 8, 18; "LAPURDI" = 5"
```

```
66 PRINT AT 9, 18; "NAFARROA" = 6"
```

```
67 PRINT AT 10, 18; "ZUBEROA" = 7"
```

```
70 INPUT F
```

```
80 IF F <> B THEN CLS:PRINT "GAIZKI, APROBA EZAZU BERRIZ": PAUSE 70:
GOTO 60
```

```
85 IF F = B THEN CLS: PRINT "OSO ONDO": PAUSE 70: GOTO 20
```

```
1000 DATA "IRUNEA", 6, "TUTERA", 6, "LIZARRA", 6, "OLITE", 6, "jAOIZ", 6,
"ZANGOTZA", 6, "ELIZONDO", 6, "TAFALLA", 6
```

```
1005 DATA "ALTSASU", 6, "GARAZI", 2, "BAIONA", 5, "MIARRITZE", 5,
"DONIBANE LOHITZUNE", 5, "MAULE", 7, "IRUN", 4
```

```
1010 DATA "DONOSTIA", 4, "EIBAR", 4, "AZPEITIA", 4, "DEBA", 4,
"ARRASATE", 4, "ZUMARRAGA", 4, "TOLOSA", 4, "BEASAIN", 4,
"BILBO", 3, "GERNIKA", 3, "DURANGO", 3
```

```
1015 DATA "PORTUGALETE", 3, "ELORRIO", 3, "GALDAKAO", 3, "BASAURI",
3, "AGURAIN", 1, "GASTEIZ", 1, "ARAMAIO", 1, "LAUDIO", 1
```

## COMMODORE 64

```
10 REM "GEOGRAFIA IKASTEN J.M. ITURRIOZ 1985 ©"  
20 RESTORE: LET C = INT(RND(1)*34) + 1  
30 FOR I = 1 TO C  
40 READ A$, B  
50 NEXT I  
60 PRINT "NON DAGO  ∪ "; A$; "?"  
61 PRINT "... "ARABA  ∪ ∪ ∪ ∪ = 1"  
62 PRINT "... "BEHENAFARROA  ∪ = 2"  
63 PRINT "... "BIZKAIA ∪ ∪ ∪ ∪ = 3"  
64 PRINT "... "GIPUZKOA ∪ ∪ ∪ = 4"  
65 PRINT "... "LAPURDI ∪ ∪ ∪ ∪ = 5"  
66 PRINT "... "NAFARROA ∪ ∪ ∪ = 6"  
67 PRINT "... "ZUBEROA  ∪ ∪ ∪ = 7"  
70 INPUT F  
80 IF F < > B THEN PRINT CLR: PRINT "GAIZKI, APROBA EZAZU BERRIZ": FOR  
    I = 0 TO 300: NEXT I: GOTO 60  
85 IF F = B THEN PRINT CLR: PRINT "OSO ONDO": FOR I = 0 TO 300: NEXT I:  
    GOTO 20  
1000 DATA IRUNEA, 6, TUTERA, 6, LIZARRA, 6  
1005 DATA OLITE, 6, AOIZ, 6, ZANGOTZA, 6, ELIZONDO, 6  
1010 DATA TAFALLA, 6, ALTSASU, 6, GARAZI, 2, BAIONA, 5  
1015 DATA MIARRITZE, 5, DONIBANE LOHITZUNE, 5  
1020 DATA MAULE, 7, IRUN, 4, DONOSTIA, 4  
1025 DATA EIBAR, 4, AZPEITIA, 4, DEBA, 4, ARRASATE, 4  
1030 DATA ZUMARRAGA, 4, TOLOSA, 4, BEASAIN, 4  
1035 DATA BILBO, 3, GERNIKA, 3, DWRANGO, 3  
1040 DATA PORTUGALETE, 3, ELORRIO, 3, GALDAKAO, 3, BASAURI, 3  
1045 DATA AGURAIN, 1, GASTEIZ, 1, ARAMAIO, 1, LAUDIO, 1
```



## IBM PC

```
10 REM "GEOGRAFIA IKASTEN J.M. ITURRIOZ 1985"  
20 RESTORE: LET C = INT(RND*34) + 1  
30 FOR I = 1 TO C  
40 READ A$, B  
50 NEXT I  
60 PRINT "NON DAGO ", A$, "?"  
61 PRINT ..... "ARABA " = 1"  
62 PRINT ..... "BEHENAFARROA " = 2"  
63 PRINT ..... "BIZKAIA " = 3"  
64 PRINT ..... "GIPUZKOA " = 4"  
65 PRINT ..... "LAPURDI " = 5"  
66 PRINT ..... "NAFARROA " = 6"  
67 PRINT ..... "ZUBEROA " = 7"  
70 INPUT F  
80 IF F <> B THEN CLS: PRINT "GAIZKI, APROBA EZAZU BERRIZ": FOR  
    I = 0 TO 300: NEXT I: GOTO 60  
85 IF F = B THEN CLS: PRINT "OSO ONDO": FOR I = 0 TO 300: NEXT I:  
    GOTO 20  
1000 DATA IRUNEA, 6, TUTERA, 6, LIZARRA, 6, OLITE, 6, TAFALLA, 6,  
    ALTSASU, 6  
1005 DATA , AOIZ, 6, ZANGOTZA, 6, ELIZONDO, 6, GARAZI, 2, BAIONA, 5,  
    MIARRITZE, 5  
1010 DATA DONIBANE LOHITZUNE, 5, MAULE, 7, IRUN, 4, DONOSTIA, 4  
1015 DATA EIBAR, 4, AZPEITIA, 4, DEBA, 4, ARRASATE, 4, ZUMARRAGA, 4,  
    TOLOSA, 4  
1020 DATA BEASAIN, 4, BILBO, 3, GERNIKA, 3, DURANGO, 3, PORTUGALETE, 3  
1025 DATA , ELORRIO, 3, GALDAKAO, 3, BASAURI, 3, AGURAIN, 1, GASTEIZ, 1  
1030 DATA ARAMAIO, 1, LAUDIO, 1
```

### 2. Adibidea:

Baldin eta, gramatika bada irakatsi edo landu nahi dugun gaia agian letrazko galdera-erantzunak izango ditugu soilik. Horrela bada 40. linea aldatu egin beharko dugu.

40 READ A\$, B → 40 READ A\$, \$

Era berean 70. linea aldatu

beharko dugu

70 INPUT F → 70 INPUT F\$

Beraz, orain 80. linea eta 85garrenako aldagien izenak aurreko aldaketan arabar aldatuko dira

80 IF F\$ <> B\$ THEN ...

85 IF F\$ = B\$ THEN ...

```

10 REM "ADITZA IKASTEN J.M. ITURRIOZ 1985 ©"
20 RESTORE :LET C = INT(RND* ) + 1      SPECTRUM & IBM PC
                                         Datu-parekopenua
20 RESTORE: LET C = INT(RND(1)* ) + 1    COMMODORE 64
30 FOR I = 1 TO C
40 READ A$, B$
50 NEXT I
60 PRINT "OSATU EZAZU ONDOKO ESALDIA; A$; "...
70 INPUT F$                               SPECTRUM & IBM PC
80 IF F$ <> B$ THEN CLS:PRINT "GAIZKI, APROBA EZAZU BERRIZ": FOR I = 0
    TO 300: NEXT I: GOTO 60
85 IF F$ = B$ THEN CLS: PRINT "OSO ONDO": FOR I = 0 TO 300: NEXT I: GOTO
    20
1000 DATA "NIK OGIA EKARRI", "DUT", ...   (SPECTRUMan)
1000 DATA NIK OGIA EKARRI, DUT, ...      (COMMODORE & IBM PC)

```