

INFORMATIKA

Ⓐ

• BEGIN

• VAR → erabiliko ditugun aldagaiak izendatzeko

↳ , , , : _____ → DATU MOTAK

⊗ Aldagaiak izendatu ostean

↳ REAL → zenbaki errealeak erabiliko baditugu.

↳ INTEGER → zenbaki osoak erabiliko baditugu.

• ZATIKETAK adierazteko

- Zenbaki errealeak dividen → /

- Zenbaki osoak dividen → DIV

• Write ('m'); → zenbait idatzita ager daitezen.

• ReadLn (m) → pertsonak sartutako datua aldagai bati esleitzeko.

■ Kontzeptu bat izendatzeko

BAI

EZ

• Azpiko barra (-)

• Edozein letra

• Zenbaki baten hosi

• Gidatza (-)

• Hutsuneak erabili

• Oinarrizko kontzeptuak (Begin, Var, ..., etab.)

■ Bi kontzeptuen arteko berdintasun egitea

$$[A = B] \neq [B = A]$$

■ Irizkinak egiteko "//" formi hasieran (programan eragirik izango ez duten irizkinak jarazeko)

■ SISTEMA MOTAK

× Hamartarria → 0-9

× Zehazgarria → 0-7

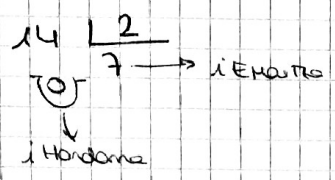
× Hexadeximala → 0-9 ABCDEF

× Bitarria → 0-1

• MOD → Hondama lotzeako (modulua)

ADIB.

iAdina = 14
 iEmaitza = iAdina DIV 2 = 7
 iHondama = iAdina MOD 2 = 0



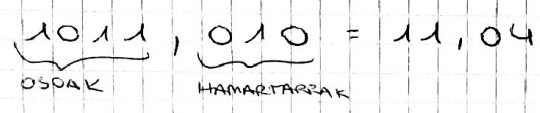
INTEGER

• DIV → zenbaki osoekin soilik
 (zenbaki errealekin EZ)

• / → zenbaki errealekin soilik

↳ 6/2 eghez zero, emaitza
 3. 0 izango da, hau da, nulua eta
 zenbaki osoak erabili eta hondama 0
 izan, emaitza REAL izango da BETI.

■ Bitarretik hamortzerera



$$(1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0) = 11$$

↳ 0 zenbaki positibotzat hartuko dira.

■ INTEGER-en → muga maximoa 32677
 ↳ muga minimoa -32678

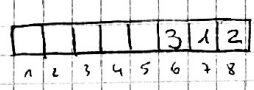
⊗ GAINEZKATzea

Kontagailu bat programatuta bogew eta 32677-na initsiko balia (balio maximoa), hurrengo zenbakia -32678 izango litzateke, eta zenbaki horietatik aurrera programatuta luke kontagailuak.

■ ZENBAKIEN NOTAZIOA

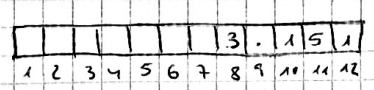
• Demagun emaitza 312 dela (iEmaitza = 312)

```
WriteLn ('Emaitza = _____', iZenbakia : 8);
```



• Demagun emaitza 3,151 dela (iEmaitza = 3,151)

```
WriteLn ('Emaitza = _____', iZenbakia : 12 : 3);
```



↑ zenbaki horietatik aurrera
 ↑ zenbaki horietatik aurrera
 ↑ zenbaki horietatik aurrera

↑ zenbaki horietatik aurrera
 ↑ zenbaki horietatik aurrera

↑ zenbaki horietatik aurrera
 ↑ zenbaki horietatik aurrera

• ZENBAIT "FUNKSI"

* SQRT → karutua

* TRUNC → komadun tenbaki oso bihurtu

* ROUND → hamartareren baliza koaduta hartuta zenbaki oso bihurtu (borobildu).

↳ ADIB. $\left\{ \begin{array}{l} rtk = 1,934 \\ \text{TRUNC}(rtk) \rightarrow 1 \\ \text{ROUND}(rtk) \rightarrow 2 \end{array} \right.$

* INT (rk) → zenbaki oso hartu

* FRAC (rk) → zenbaki hamartarak hartu

■ BOOLEAN → TRUE edo FALSE

• **OR** → $boAngia := boA \text{ OR } boB$

A sakagilua boA	TRUE	FALSE	TRUE	FALSE
B sakagilua boB	TRUE	FALSE	FALSE	TRUE
Bonbale boAngia	TRUE	FALSE	TRUE	TRUE

• **NOT**

boBale	TRUE	FALSE
NOT Bale	FALSE	TRUE

• **AND** → $boAngia := boA \text{ AND } boB$

boA	FALSE	TRUE	FALSE	TRUE
boB	FALSE	TRUE	TRUE	FALSE
boAngia	FALSE	TRUE	FALSE	FALSE

	ERAGILEA	ALDAGAI KOP.	ALDAGAIEN DATU MOTA	EMAITAREN DATU MOTA
ERAGIKETA BOOLEARRAK	NOT	1	BOOLEAN	BOOLEAN
	OR	2		
	AND	2		
KONPARAKETAK	>	2	REAL	BOOLEAN
	<			
	>=			
	<=			
	=			
	<>			

BALDINTZAKOAK

• IF THEN

① VAR iNum: INTEGER;

BEGIN

[Write ('Zer ezazu zenbaki bat?');

[ReadLn (iNum);

[IF iNum > 0 THEN

BEGIN

 WriteLn ('Positiboa da');

[END;

[ReadLn;

② IF iNum > 0 THEN

BEGIN

 IF iNum MOD 2 = 0 THEN

 BEGIN

 WriteLn ('Bakarra da');

 END

 ELSE

 BEGIN

 WriteLn ('Bakarra da');

 END;

END;

↙ ↘
ELSE-ren aurretik ez da (i) jartzen

A

```

IF iNum > 0 THEN
  WriteLn ('Positivo');
  WriteLn ('Agur');

```

-6 written badago, "Agur"
 fontai lenaduko da, et baitago IF-men
 berron.

```

IF iNum > 0 THEN
  BEGIN
    WriteLn ('Positivo');
  END

```

ELSE-ren aurretik etan da (i) jarri

```

ELSE IF iNum > 0 THEN
  BEGIN
    WriteLn ('Negatibo');
  END
ELSE
  BEGIN
    WriteLn ('zero');
  END;

```

Sententia bakarra badago, et da beharretan BEGIN-END jarria.

CASE OF

Erabilgarriaren zenbaki bat eskatu eta

- 1 → Hoba
- 2 → Aupa
- 3 → Agur

```

Write ('Sartu zenbaki bat!');
ReadLn (iNum);

```

```

IF iNum = 1 THEN
  WriteLn ('Hoba')
ELSE IF iNum = 2 THEN
  WriteLn ('Aupa')
ELSE IF iNum = 3 THEN
  WriteLn ('Agur')
ELSE
  WriteLn ('Zerbaki et da zuzena');

```

```

CASE iNum OF
  1: WriteLn ('Hoba');
  2: WriteLn ('Aupa');
  3: WriteLn ('Agur')
ELSE
  WriteLn ('Zerbaki et da zuzena');
END;

```

- MENUETAN erabilten da
- Etan da zerbaki eratekin erabili.

KARAKTEREAK

→ Ordenagailaren teklatuaren dagoen edozein simbolo.

ASCII KODEA

(256 KARAKTERE)

'Ø' → 48	'A' → 65	'a' → 97
'1' → 49	'B' → 66	'b' → 98
⋮	⋮	⋮
'9' → 57	'Z' → 90	'z' → 122

$\xrightarrow{+32}$
 $\xleftarrow{-32}$

VAR

cCar: CHAR;

iNum: INTEGER;

BEGIN

cCar := 'a';

cCar := cCar - 32

→ Ekin da eragiketa matematikarekin egun Char kostako datuekin.

cCar := CHR(ORD(cCar) - 32); ✓

97
65

ORD → karaktere → zenbaki
CHR → zenbaki → karaktere

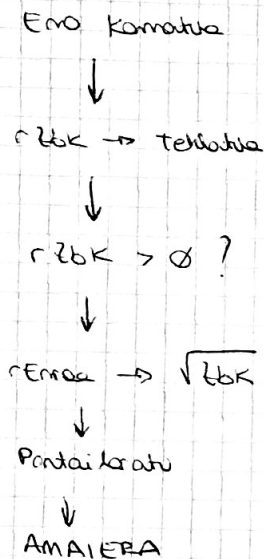
- cCar := 'a';
cCar := UPCASE(cCar); → LETRA HANDI BIHURTU
- cCar := 'A';
cCar := LOWERCASE(cCar); → LETRA XEHE BIHURTU
- cCar := 'B';
cCar := PRED(cCar); → AURREKO LETRA BIHURTU
- cCar := 'A';
cCar := SUCC(cCar); → HURRENGO LETRA BIHURTU
- INC(cCar, 3); → + 3
- DEC(cCar, 3); → - 3

ALDAGAIEN ADIERAZPENA

VAR

iNum: INTEGER;
 WNum: WORD;
 SHNum: SHORTINT;
 byNum: BYTE
 lNum: LONGINT;
 rNum: REAL;
 cChar: CHAR;
 boResul: BOOLEAN

● ERRO KARRATUA



```

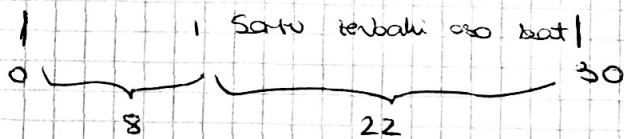
PROGRAM Erro-karratua;
VAR
  r7bk, rEmoa: REAL;
BEGIN
  Writeln ('Sartu edozein zenbaki erreala');
  Readln (r7bk);
  IF r7bk >= 0 THEN
    rEmoa := Sqrt (r7bk);
  END IF;
  Writeln ('Emaitza', rEmoa, 'irango da');
END.
  
```

Sartutako balakoa dagueraz, eta da beharrezkoa BEZIN END jarria.

● KARAKTERE KATEAK

Write ('Sartu zenbaki oso bat: 30') → TABALERA

Guere esaldiak 22 karaktere dituela jakinda:



OHARRA !!!

Beharrezko dituen bakoitza karaktere gutxiago janz gero, programak ez du kasurik egongo.

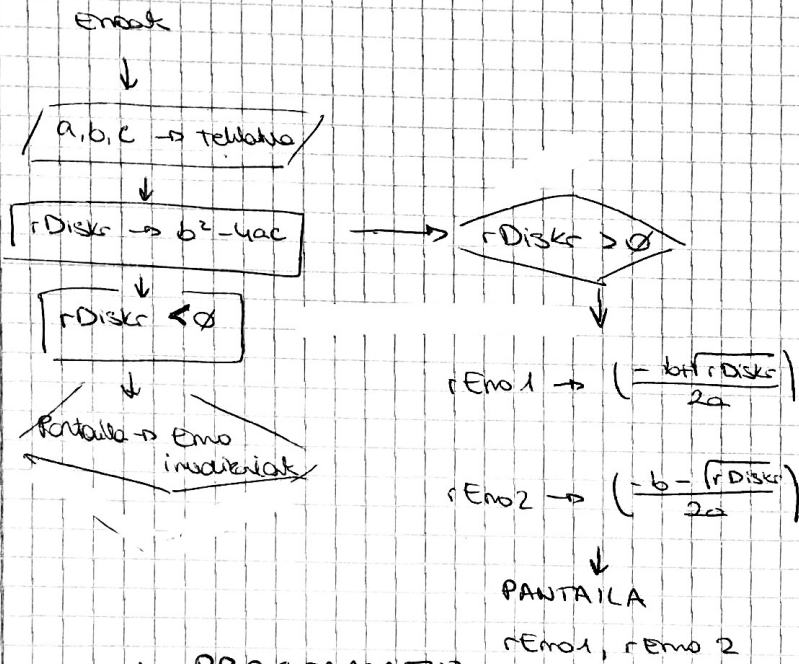
⊕ ZENBAKI ERREALAK

Write (r+bk) → zientifikoa

zelua ⊕ Potentzia ⊕ Hamartarik ⊕ E ⊕ zelua ⊕ Bermetzatzea

ADIB. 8.05503019 E004

⊙ ERROAK



PROGRAMATUZ

```

PROGRAM erroak;
VAR
  a, b, c, rDiskr, rEmo1, rEmo2: REAL;
BEGIN
  Writeln ('Sartu...');
  Readln (a, b, c);
  rDiskr := SQR(b) - 4*a*c;

  IF rDiskr < 0 THEN
    Writeln ('erroak indikatzen dira')
  ELSE IF rDiskr > 0 THEN
    BEGIN
      rEmo1 := (-b + SQR(rDiskr)) / (2*a);
      rEmo2 := (-b - SQR(rDiskr)) / (2*a);
      Writeln (rEmo1, 'eta', rEmo2);
    END
  END

```


• EKZEKUSIOA

- 1) Kontagailuen (ikent) hasierako balio bat eman (1)
- 2) While-ren amaieran $ikent := ikent + 1$ esan $ikent$ aldatzen daiteke.
- 3) While-ak funtzionatzen jarraituko du zehaztutako baldintza jada betetzen ez den arte; hau da, iterazio bakoitzean baldintza betetzen den egiaztatuko du. Betetzen ez bada, ez da berrizt egiaztatuko.

REPEAT UNTIL

↳ Et dakigu zenbatetan errepikatuko behar den kodea baina kodari buruz behin gutxienez egiaztatuko del

REPEAT

_____ ;
_____ ;
_____ ;

} Errepikatuko den kodea

UNTIL baldintza;

↳ UNTIL-aren aurreko ; ken daiteke.

- Aldagaiak aldatu behar dira

• EKZEKUSIOA

- 1) Kontagailuen (ikent) hasierako balio bat eman (1)
- 2) Repeat-aren barruan kontagailua aldatu ~~er~~. ($ikent := ikent + 1$)
- 3) Repeat-a egiaztatuko da geldirik inzipien berritu arte; hau da, kodea egiaztatuko da, geldirik inzipien betetzen den egiaztatuko du eta, betetzen ez bada, berrizt egiaztatuko da.

- Badaukizu zenbat aldiz egiaztatuko behar den?

↳ Bai: **FOR**

↳ Ez: • Gutxienez zenbat aldiz?

↳ Behin gutxienez: **REPEAT**

↳ Behar bako inoiz: **WHILE**

Ⓐ

ONARPENAK

Datu bat era jamaian eskatu DATU ZUZENA eskatu arte.

ONARTZEKO 2 ERA

(1) ERRORE MEZURIK GABE \longrightarrow REPEAT

Zenbaki \oplus bat sartu: -6

" " " " : -5

" " " " : 4 \longrightarrow Ez da gehiago eskatu

REPEAT

```
Write ('Zenbaki  $\oplus$  bat sartu: ');
```

```
ReadLn (i1bk);
```

```
UNTIL i1bk > 0
```

(2) ERRORE MEZUAREKIN \longrightarrow WHILE

Zenbaki \oplus bat sartu: -6

ERROREA! Zenbakia ez da zuzena

Zenbaki \oplus bat sartu: 8 \longrightarrow Ez da gehiago eskatu

```
Write ('Sartu zenbaki  $\oplus$  bat: ');
```

```
ReadLn (i1bk);
```

```
WHILE i1bk <= 0 DO
```

```
BEGIN
```

```
WriteLn ('ERROREA! Zenbakia ez da zuzena
```

```
Write ('Sartu zenbaki  $\oplus$  bat: ');
```

```
ReadLn (i1bk);
```

```
END;
```

(A) ARIKETAK

(4) BISIESTO

URTEA = x $\left\{ \begin{array}{l} \rightarrow 2020 \frac{4}{\emptyset} \text{ eta } 2020 \frac{100}{20} \\ \rightarrow 2020 \frac{400}{\emptyset} \end{array} \right.$

PROGRAM Bisiesto;

VAR

iUrtea: INTEGER;

BEGIN

Write ('Scr ezazu urte bat: ');

ReadLn (iUrtea);

IF (iUrtea MOD 4 = 0) AND (iUrtea MOD 100 <> 0) THEN

BEGIN

WriteLn ('Bisiesto da!')

END

ELSE IF iUrtea MOD 400 = 0 THEN

BEGIN

WriteLn ('Bisiesto da!')

END

ELSE

BEGIN

WriteLn ('Ez da bisiesto');

END;

ReadLn;

END.

BESTE ERA BATEAN

IF (iUrtea MOD 4 = 0) AND (iUrtea MOD 100 <> 0) OR (iUrtea MOD 400 = 0) THEN

WriteLn ('Bisiesto da')

ELSE

WriteLn ('Ez da bisiesto');

SENTENTZIA BAKARRA

Parentesiak jarraia BEHARRERAKOA

ELSE-ren aurritik ; ER

Sententzia bakarra dena, ez da beharrezkoa BEGIN-END jarraia

5 ZENBAKI NATURALAK

```
PROGRAM zenbakiNaturalak;  
VAR  
  n, i: INTEGER;  
  
BEGIN  
  Write ('Sartu zenbaki bat: ');  
  Readln (n);  
  FOR i:=1 TO n DO  
    Write (i, ' ');  
    Readln;   
  END.
```

↳ Renbakiak at eta HURSONEA

ADIBIDEA

```
n = 3  
1 2 3  
  
n = -6  
↳ FOR-etik IRITEN
```

6 FAKTORIALA

```
PROGRAM Faktoriada;  
VAR  
  n, iEmaitza, i: INTEGER;  
  
BEGIN  
  Write ('Sartu zenbaki bat: ');  
  Readln (n);  
  IF n < 0 THEN  
    Write ('ERROREA')  
  ELSE  
    BEGIN  
      iEmaitza := 1;  
      FOR i:=n DOWNTO 1 DO  
        BEGIN  
          iEmaitza := iEmaitza * i;  
        END;  
      Write (n, '! = ', iEmaitza);  
    END.
```

↳ Egin baita zenbaki negatibo baten faktoriada kalkulatu.

↳ "iEmaitza" aldagia erabili ahal izateko, lehenera hasieratu behar dugu.

↳ BEHERAKA

↳ Aldagai iterazio bakoitza beridatiko dugu.

ADIBIDEA

```
n = 4      i = 4 3 2 1  
iEmaitza = 1 * 4 = 4  
iEmaitza = 4 * 3 = 12  
iEmaitza = 12 * 2 = 24  
iEmaitza = 24 * 1 = 24  
4! = 24
```


7 POSITIBOAK ONARTU ETA BIDERKATU (* ERABILU GABE)

PROGRAM Biderketa;

VAR

i1bk1, i1bk2, iEmaitza : INTEGER;

BEGIN

REPEAT

Write ('Sartu zenbaki positibo bat: ');

Readln (i1bk1);

UNTIL i1bk1 > 0;

REPEAT

Write ('Sartu zenbaki positibo bat: ');

Read (i1bk2);

UNTIL i1bk2 > 0;

iEmaitza := 0;

FOR i := 1 TO i1bk2 DO

BEGIN

iEmaitza := iEmaitza + i1bk1;

END;

WriteLn (i1bk1, '*', i1bk2, '=' iEmaitza);

END.

8 DIGITU KOPURUA

PROGRAM DigituKopurua;

VAR

ckar : CHAR;

ikont : INTEGER;

BEGIN

ikont := 0;

Write ('Sar ezazu puntu batez bukatzen den zenbaki serie bat: ');

REPEAT

Read (ckar);

IF (ckar >='0') AND (ckar <='9') THEN

ikont := ikont + 1;

UNTIL ckar = '.';

WriteLn (ikont, 'Digitu sartu dituzu');

END.

9. DIGITUEN BATUKETA

PROGRAM DigituenBatuketa;

VAR

ckor : CHAR;

iBatuketa : INTEGER;

BEGIN

iBatuketa := 0;

Write ('Scr ezazu puntu batz emanaturiko serie bat: ');

REPEAT

Read (ckor);

IF (ckor >='0') AND (ckor <='9') THEN

iBatuketa := iBatuketa + ORD (ckor) - $\overbrace{\text{ORD}('0')}^{48}$;

UNTIL ckor = '.';

WriteLn ('Sartutako digituen ordezko batuketaren erantzua', iBatuketa, 'da');

ReadLn;

END.

10. EXPONENTIALA

PROGRAM Exponentziala kalkulatu;

VAR

iOinaria, iExp, iEmaitza, i : INTEGER;

BEGIN

REPEAT

Write ('Oinaria: ');

ReadLn (iOinaria);

UNTIL iOinaria > 0;

REPEAT

Write ('Exponentziala: ');

Read (iExp);

UNTIL iExp >= 0;

iEmaitza := 1;

FOR i:=1 TO iExp DO

iEmaitza := iEmaitza * iOinaria;

WriteLn ('Emaitza:', iEmaitza);

ReadLn;

END.

EXP
OINARIA
 $\begin{matrix} \downarrow \\ a^b \\ \downarrow \\ a \cdot a \cdot a \end{matrix}$

11

ADIB.

$$a=2$$

$$n=3$$

$$\sum_{k=0}^3 2^{3-k} = \frac{2^{3-0}}{2 \cdot 2 \cdot 2} + \frac{2^{3-1}}{2 \cdot 2} + \frac{2^{3-2}}{2} + \frac{2^{3-3}}{1} = 15 = \sum_{k=0}^n a^{n-k}$$

PROGRAM Batukona;

VAR

a, n, k, iBatukona, iPot, i: INTEGER;

BEGIN

REPEAT

Write ('Sesetahu a-ren balise: ');

ReadLn(a);

UNTIL a > 0;

REPEAT

Write ('Sesetahu n-ren balise: ');

ReadLn(n);

UNTIL n > 0;

iBatukona := 0;

FOR k := 0 TO n DO

BEGIN

iPot := 1;

FOR i := 1 TO n-k DO

BEGIN

iPot := iPot * a;

END;

iBatukona := iBatukona + iPot;

END;

WriteLn ('Emaitte: ', iBatukona);

ReadLn;

END.

12

KARAKTERE SERIEA

PROGRAM Karaktere Seriea;

VAR

ckor : CHAR;

BEGIN

Write ('Sor ezazu puntu batea buratunko karaktere seriea:');

REPEAT

Read (ckor);

IF (ckor >= 'a') AND (ckor <= 'z') THEN

ckor := UPCASE (ckor);

Write (ckor);

UNTIL ckor = '.';

Readln;

END.

a3! Ab, 4 l.

A3! AB, 4 L.

LETRA
LAKETI
BIHURTU