

②

AZPIPROGRAMAK

• FUNTzioAK (FUNCTION)

- Balio bakera itzultzen dute BETI.
- Ez da erreferentziat pasatutako parametrok sartzen (hau da, ezin da azpiprogramaren baten aldatuko den parametrok sartu).

• PROZEDURAK (PROCEDURE)

- Erreferentziat pasatutako parametroak erabili daitezke.
- Ez dute datuak itzultzen edo datu bat bakera zeluzko itzultzen dute.
- Jazotako datuak alda ditzakete.

• ERREFERENTZIAT PASATUTAKO PARAMETROAK 2 KASUTAN ERABILITEN

DIRA:

- A) Prozedura batek bakera bat bakera zeluzko itzuli behar duen.
- B) Prozedura batek jazotako parametro baten balioa aldatu behar duen.

ADIBIDEA

Bi tenbaki jaso eta eragiketak egiten dituen programa

- 1) Jazotako tenbakiak baten dituen FUNTzioA
- 2) Tenbakiak eskatzen duen FUNTzioA
- 3) Batuketaren eraketa ERAKUSTEN duen PROZEDURA
- 4) Tenbakiaren arteko batuketa eta kenketa egiten dituen PROZEDURA
- 5) Tenbakiaren bat bikoizten duen PROZEDURA

PROGRAM Adibidea;

IRTENGO DEN
DATUAREN DATU-MOTA

1) FUNCTION fn Batu (i1bk1, i1bk2 : INTEGER) : INTEGER;
VAR
iBatuketa : INTEGER; → UNEKO PARAMETROA (Funtzioa soilik erabili)
BEGIN
iBatuketa := i1bk1 + i1bk2;
fn Batu := iBatuketa; → Funtzio honek iBatuketa-ren emaitza itzuliko du.
END;

2) FUNCTION fn Zenbakia Eskatu : INTEGER;
VAR
i1bk : INTEGER;
BEGIN
Write ('Sor ezazu tenbaki oso bat: ');
Readln (i1bk);
fn Zenbakia Eskatu := i1bk;
END;

→ INTEGER-ekin lanetan baliokide

→ Funtzio deklaratuak eta programa nagusian deklaratuak aldatzeki eta dute bertan iten berdina iten.

3) PROCEDURE p Emaitza Erakutsi (iBatuketa : INTEGER);
BEGIN
WriteLn ('Batuketaren emaitza: ', iBatuketa);
END;

Kasu honetan
prozedura erabili
dugu eta baita datuak
aldatzen

4) PROCEDURE p Engiketak (i1bk1, i1bk2 : INTEGER, VAR iBatuketa, iKenketa :
INTEGER);
BEGIN
iBatuketa := i1bk1 + i1bk2;
iKenketa := i1bk1 - i1bk2;
END;

Prozedura erabi dugu 2 datu
intertan diralako.

5) PROCEDURE pBiderkatu x 2 (VAR i7bk : INTEGER);

BEGIN

i7bk := i7bk * 2;

END;

Procedure esabildi duzu sartutako parametroz atpi programaren berron aldatzen delako.

VAR → PROGRAMA NAGUSIKO (PN) ALDAGAIEN DEKLARAZIOA

iBatuketa, i7bk1, i7bk2, iKenketa : INTEGER;

BEGIN

① iBatuketa := fnBatu (3, 4);

pEmaitza Erakutsi (iBatuketa);

② i7bk1 := fnZenbakiEskatu;

WriteLn (i7bk1)

③ i7bk2 := fnZenbakiEskatu;

WriteLn (i7bk2);

④ iBatuketa := fnBatu (i7bk1, i7bk2);

pEmaitza Erakutsi (iBatuketa);

pEraketa (i7bk1, i7bk2, iBatuketa, iKenketa);

⑤ pEmaitza Erakutsi (iBatuketa);

WriteLn (iKenketa);

⑥ pBiderkatu x 2 (i7bk1);

WriteLn (i7bk1);

ReadLn;

END.

ADIBIDEA

EJEKUZIOA

1. Batuketaren emaitza: 7

2. Sor ezazu zenbaki

oso bat: 2

3. Sor ezazu zenbaki

oso bat: 4

4. Batuketaren emaitza: 6

5. Batuketaren emaitza: 6

-2

6. 4

⑧ ARIKETAK

14

PROGRAM Arik14;

PROCEDURE pEmaitzaEtaHondara (i7bk1, i7bk2: INTEGER);

VAR

iKont: INTEGER;

Prozedura erabilhen dugu i7bk1-en balioa azpiprogramaren berran aldatzen delako

BEGIN

iKont := 0;

WHILE i7bk1 > i7bk2 DO

BEGIN

i7bk1 := i7bk1 - i7bk2;

iKont := iKont + 1; → Kenketa egun daitereen bakoitzean iKont-i +1 egia

END;

VAR

i7bk1, i7bk2: INTEGER;

BEGIN

REPEAT

Write('Ser ezazu zenbaki positibo bat: ');

Readln (i7bk1);

UNTIL i7bk1 > 0;

REPEAT

Write('Ser ezazu zenbaki positibo bat: ');

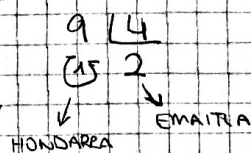
Readln (i7bk2);

UNTIL i7bk2 > 0;

pEmaitzaEtaHondara (i7bk1, i7bk2);

Readln;

END.



iKont = 1
iKont = 2
i7bk1 = 9 - 4 = 5 → HONDARRA
i7bk2 = 4
iKont = 2 + EMAITZA

15.

PROGRAM Arik 15;

FUNCTION fn 2frak terbato (li7bk: LONGINT): INTEGER;

VAR

Funksia erabilerika dugu datu bakarra irudika dugu bakarra.

ikont: INTEGER;

BEGIN

ikont := 0;

REPEAT \rightarrow REPEAT-ek eta du BEGIN-END eskaren.

li7bk := li7bk DIV 10; \rightarrow 10-eratik zatitu aldi den balantzen ikont + 1.

ikont := ikont + 1 \rightarrow Et da balantzea

UNTIL li7bk = 0;

fn 2frak terbato := ikont;

END;

VAR

li7bk: LONGINT;

i3zifro Bano Gehiago, x: INTEGER;

BEGIN

i3zifro Bano Gehiago := 0; \rightarrow Hasieratu balantza dugu.

REPEAT

Write ('Zer balantza: ');

Readln (li7bk);

x := fn 2frak terbato (li7bk);

IF x > 3 THEN

i3zifro Bano Gehiago := i3zifro Bano Gehiago + 1;

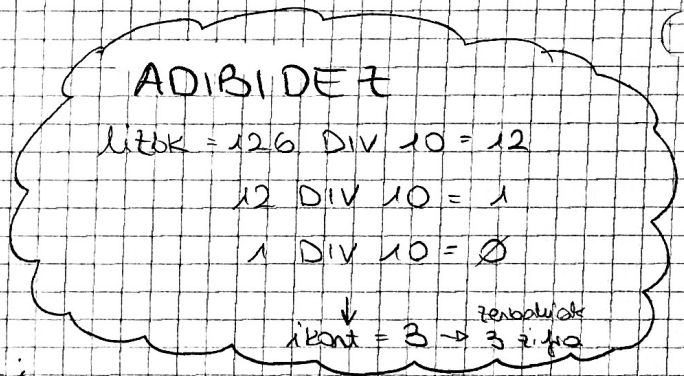
UNTIL li7bk = 0;

WriteLn ('3 zifro bano gehiagoko', i3zifro Bano Gehiago, ' zer balantza sortu dituzu');

ReadLn;

END.

\rightarrow Programa gutxiets amaitzen BEHARRERIKOA



\rightarrow Aldagai horiek 3 zifro bano gehiago zer balantza kopurua zer balantza du.

16.

PROGRAM Arik 16;

FUNCTION fn BueltaEman (liZbk: LONGINT): LONGINT;

VAR

liEmante: LONGINT;

BEGIN

liEmante := 0;

WHILE liZbk <> 0 DO terbalukan angka

BEGIN

liEmante := liEmante * 10 + liZbk MOD 10;

liZbk := liZbk DIV 10;

END;

fn BueltaEman := liEmante;

END;

VAR

liZbk, x: INTEGER;

BEGIN

REPEAT

Write ('terbalukan: ');

Readln (liZbk);

x := fn BueltaEman (liZbk);

IF x = liZbk THEN

ikant := ikant + 1;

UNTIL liZbk = 0;

Writeln (ikant, 'terbaku kapikue sortu datus');

END.

$$liNum = 123 \text{ DIV } 10 = 12$$

$$12 \text{ DIV } 10 = 1$$

$$1 \text{ DIV } 10 = 0$$

$$liEmante = 0 * 10 + 3 = 3$$

$$3 * 10 + 2 = 32$$

$$32 * 10 + 1 = 321$$

17

PROGRAM Arik 17;

FUNCTION fn fatiketaTxikitu (a, b: INTEGER): INTEGER;

BEGIN

WHILE a <> b DO

BEGIN

IF a > b THEN

a := a - b

ELSE

b := b - a;

END;

fn fatiketaTxikitu := a; → a edo b ja daiteke hemen, bala berdina izango baitute.

END;

VAR → zenbaki naturalak

i7bk, i7zen, i7atiketa: INTEGER;

↳ zenbaki naturalak

BEGIN

REPEAT

Write ('zenbaki naturalak:');

Readln(i7bk);

UNTIL i7bk > 0;

REPEAT

Write ('zenbaki naturalak:');

Readln(i7zen);

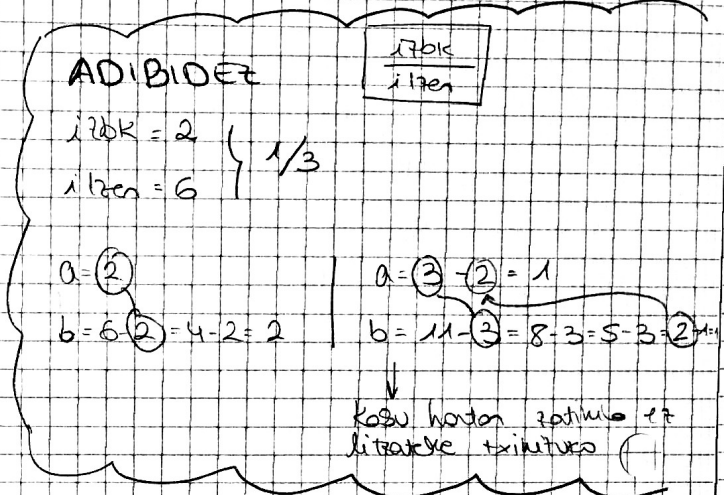
UNTIL i7zen > 0;

i7atiketa := fn fatiketaTxikitu (i7bk, i7zen);

Write ('Txikietara den zatikua: ', i7bk / i7atiketa, ' / ', i7zen / i7atiketa);

Readln;

END.



18.

PROGRAM Arik 18;

Urae bisiestoe den egiartaren duen funtzioa

FUNCTION fnBisiesto (iUrte: INTEGER): BOOLEAN;

VAR

boEmaitza: BOOLEAN;

BEGIN

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

boEmaitza := TRUE -> Aurrik etarritako baldintzak betetzen badira,
urte bisiestoez bago da

ELSE

boEmaitza := FALSE

fnBisiesto := boEmaitza;

END;

FUNCTION fnHilabetekoEgnak (iHila, iUrte: INTEGER): INTEGER;

VAR

iEgnak: INTEGER;

BEGIN

CASE iHila OF

1, 3, 5, 7, 8, 10, 12: iEgnak := 31;

4, 6, 9, 11: iEgnak := 30;

2: IF fnBisiesto (iUrte) = TRUE THEN

iEgnak := 29

ELSE

iEgnak := 28;

END;

fnHilabetekoEgnak := iEgnak;

END;

Hilabete bakoitzaren egun kopurua jakiteko funtzioa

→ Aldatuak aldatuko dituzulako.

Data futuruak
duen
prozedura

```
PROCEDURE pData Futura (VAR iEguna, iHila, iUrtea: INTEGER);
BEGIN
  WHILE iEguna > fn HilabetekoEgunak (iHila, iUrtea) DO
  BEGIN
    iEguna := iUrtea - fn HilabetekoEgunak (iHila, iUrtea);
    iHila := iHila + 1;
    IF iHila > 12 THEN
    BEGIN
      iUrtea := iUrtea + 1;
      iHila := 1;
    END;
  END;
END;
```

ADIBIDEZ

$$\begin{aligned} \text{Eguna} &= 96 - 31 = 65 \\ 65 - 30 &= 35 \\ 35 - 31 &= 4 \end{aligned}$$
$$\begin{aligned} \text{Hilabetea} &= 10 + 1 = 11 \\ 11 + 1 &= 12 \\ 12 + 1 &= 13 \end{aligned}$$
$$\text{Urtea} = 2000 + 1 = 2001$$

VAR

iEguna, iHila, iUrtea, iEgunkop: INTEGER;

BEGIN

REPEAT

Write ('Eguna: ');

Readln (iEguna);

Write ('Hilabetea: ');

Readln (iHila);

Write ('Urtea: ');

Readln (iUrtea);

UNTIL (iUrtea > 0) AND (iHila >= 1) AND (iHila <= 12) AND (iEguna >= 1)
AND (iEguna <= fn HilabetekoEgunak (iHila, iUrtea));

REPEAT

Write (' Zerbat opa egun irago dituzu? ');

Readln (iEgunkop);

UNTIL (iEgunkop > 0);

iEguna := iEguna + iEgunkop;

pData Futura (iEguna, iHila, iUrtea);

Write (' OPORREN AMAIERAKO EGUNA: ', iEguna, '-', iHila, '-', iUrtea);

Readln;

END.