

2.6 Azpiprogramak

2.6.1 Sarrera

2.6.2 Azpiprograma motak, definizioa eta erabilera

2.6.3 Parametroen erabilera

2.6.4 Parametroak erreferentziaz edo kopiaz pasatzea

2.6.5 Azpiprogramen zehaztapena

Ariketak

2.6.1 Sarrera (I)

Problema bat ebazteko:

- Problema azpi-problematan zatitu
- Azpi-problema bakoitza bere aldetik ebatzi
- Azpi-problema bakoitza berriro azpi-problematan zatitu
- Horrela jarraitu azpi-problemak ebazteko errazak diren arte

Sarrera (II)

➤ Adibidea: a, b bi zenbaki positibo emanda, $a \geq b$, idatzi $\binom{a}{b} = \frac{a!}{b! (a-b)!}$

➤ Soluzioa:

1. a eta b balioak irakurri

2. emaitza kalkulatu

3. emaitza idatzi

➤ 2. emaitza kalkulatu

2.1 kalkulatu a! eta utzi *a_fakt* aldagaian

2.2 kalkulatu b! eta utzi *b_fakt* aldagaian

2.3 kalkulatu (a-b)! eta utzi *a_ken_b_fakt* aldagaian

2.4 emaitza =

$$a_fakt / (b_fakt * a_ken_b_fakt)$$

Sarrera (III)

2. emaitza kalkulatu

```
a_fakt =1  
for i=1 to a step 1  
    a_fakt = a_fakt * i  
next i
```

 } **2.1**

```
b_fakt =1  
for i=1 to b step 1  
    b_fakt = b_fakt * i  
next i
```

 } **2.2**

```
a_ken_b_fakt =1  
for i=1 to (a-b) step 1  
    a_ken_b_fakt = a_ken_b_fakt * i  
next i
```

 } **2.3**

emaitza = a_fakt / (b_fakt * a_ken_b_fakt)

Sarrera (IV)

- 2.1, 2.2, 2.3 problema berdinak dira:

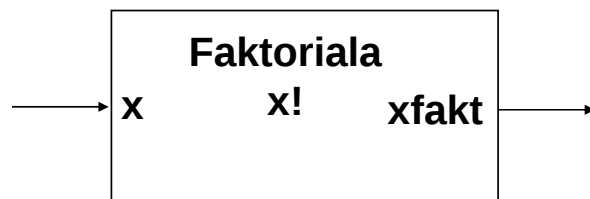
kalkulatu $x!$ eta utzi xfakt aldagaian

xfakt =1

for i=1 to x step 1

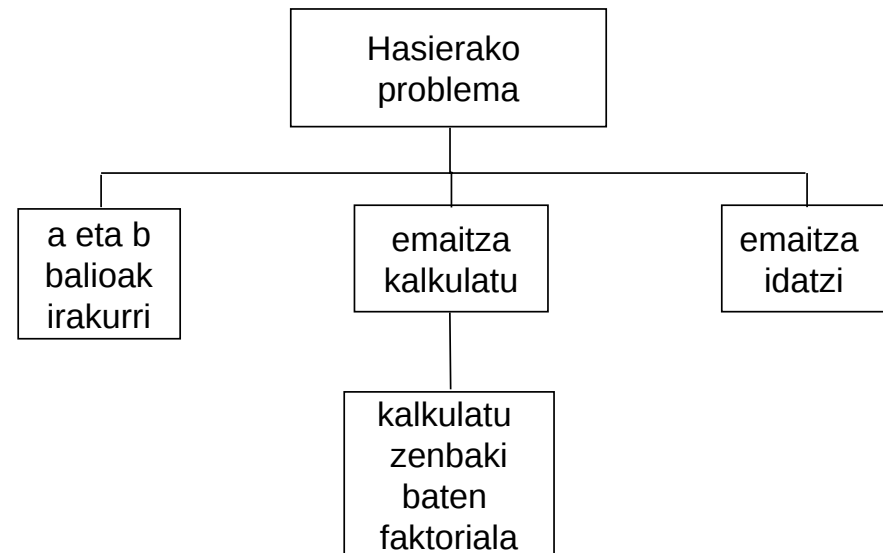
 xfakt = xfakt * i

next i



Sarrera (V)

- Soluzioa era hierarkikoan antolatuta:



➤ Modularizazioa:

- azpiprograma bakoitzak helburu jakin bat du

➤ Garapen denbora murriztea:

- Programen lerro kopuru osoa murrizten da
- Erroreak egiteko probabilitatea ere jaisten da

➤ Datuen independentzia eta ezkutatzea:

- Programaren beste zatiekiko independentea da
- Bere datuak mantentzen ditu
- Deitzen dion programa zatiarekin komunikatzeko interfaze argia definituz
- Beharrezko EZ duen informaziorako atzipenik ez dauka

➤ Bi mota dauzkagu VisualBasic-en:

➤ Funtzioa:

- Funtzio baten deiak BETI balio bat bueltatzen du
- Beraz, adierazpenetan erabil daiteke
- Exekuzioan, funtzioaren deia bueltatzen duen balioaz ordezkutzen da
- Adib: $y = \sin(x)$
 - ‘ y aldagaiari $\sin(x)$ funtzioa
 - ‘ ebaluatu ondoren lortzen
 - ‘ den balioa esleitzen zaio

➤ **Prozedura:**

- Prozeduraren deiak ez du inungo baliorik bueltatzen
- Beraz, ezin dira adierazpenetan erabili
- Parametroen balioak aldatu ditzake balioen bat itzultzeko (baina ez daude derrigortuta)

Funtzioaren definizioa

➤ Sintaxia VBasic-en:

```
[Private] Function izena ([parametroak]) [As datu_mota]
    [aldagai lokalen erazagupena]
    [aginduak]
    [izena = adierazpena]
```

End Function

➤ Adibidea:

```
Function Faktoriala(X As Integer) As Long
    DIM i As Integer, xfakt As Long
    xfakt = 1
    for i=1 to x step 1
        xfakt = xfakt * i
    next i
    Faktoriala = xfakt
End Function
```

Ikusi: t.VB-funtzioak.pdf

Funtzioaren erabilera

```
Private Sub Exekutatu_Click()
    DIM A As Integer, B As Integer
    DIM A_Fakt, B_Fakt, A_Ken_B_Fakt As Long
    REM Programa nagusia hemen hasten da
    A = InputBox("Sartu zenbaki bat:")
    B = InputBox("Beste zenbaki bat:")
    A_Fakt = Faktoriala(A)
    B_Fakt = Faktoriala(B)
    A_Ken_B_Fakt = Faktoriala(A-B)
    Emaiza = A_fakt / (B_Fakt * A_Ken_B_Fakt)
    MsgBox("Konbinazio kopurua: " & emaitza)
End Sub
```

```
Function Faktoriala(X As Integer) As Long
    REM Aurrebaldintza: X >= 0
    REM Postbaldintza: X-en faktoriala itzultzen du
    DIM i As Integer, xFakt As Long
    xFakt = 1
    FOR i= 1 TO X STEP 1
        xFakt = xFakt * i
    NEXT i
    Faktoriala = xFakt
End Function
```

Prozeduraren definizioa

➤ Sintaxia VBasic-en:

```
[Private] Sub izena ([ parametroak])  
    [ aldagai lokalen erazagupena]  
    [ aginduak]
```

End Sub

➤ Adibidea:

```
Sub Faktoriala(ByVal X As Integer, xFakt As Long)  
    DIM i As Integer  
    xFakt = 1  
    for i=1 to x step 1  
        xFakt = xFakt * i  
    next i  
End Sub
```

Prozeduraren erabilera: Call sententzia

```
Private Sub Exekutatu_Click()  
    DIM A As Integer, B As Integer  
    DIM A_Fakt, B_Fakt, A_ken_B_Fakt As Long  
    REM Programa nagusia hemen hasten da  
    A = InputBox("Sartu zenbaki bat:")  
    B = InputBox("Beste zenbaki bat:")  
    Call Faktoriala(A, A_Fakt)  
    Call Faktoriala(B, B_Fakt)  
    Call Faktoriala(A-B, A_Ken_B_Fakt)  
    Emaita = A_fakt / (B_Fakt * A_Ken_B_Fakt)  
    MsgBox("Konbinazio kopurua: " & emaitza)  
End Sub
```

```
Sub Faktoriala(ByVal X As Integer, xFakt As Long)  
    REM Aurrebaldintza: X >= 0  
    REM Postbaldintza: xFakt aldagaian X-en  
    REM faktoriala dago  
    DIM i As Integer  
    xFakt = 1  
    FOR i= 1 TO X STEP 1  
        xFakt = xFakt * i  
    NEXT i  
End Sub
```

2.6.3 Prozedura-dei baten exekuzioa (I)

```

Sub Faktoriala(X As Integer, xFakt As Long)
REM Aurrebaldintza: X >= 0
REM Postbaldintza: Fakt aldagaian X-en
REM          faktoriala dago
    DIM i As Integer
    xFakt = 1
    FOR i= 1 TO X STEP 1
        xFakt = xFakt * i
    NEXT i
End Sub

```

- Programa nagusiko aldagaiak: U eta V As Integer eta W As Long
- Suposatu hasierako balioak hauek direla:
{ U: 4, V: 2, W: 15 }
- eta honako deia egiten dela:

Faktoriala(U + V, W)

Prozedura-dei baten exekuzioa (II)

```

REM programa nagusia
DIM U, V As Integer, W As Long
U = InputBox("Sartu datu bat")
V = InputBox("Beste datu bat")
W = InputBox("Beste datu bat")
...

```

Call Faktoriala (U+V, W)

MsgBox("Faktoriala: " & W & " da")

```

SUB Faktoriala (X AS Integer,
Fakt AS Long)
DIM I AS INTEGER

```

e. Azpiprogramako objektuen desagertzea

- Tokia hartzen du prozeduraren parametroak eta aldagai lokalak gordetzeko

- Parametro errealak ebaluatzen dira, eta

d. Buelta programa nagusira

- Emaita parametroen balioak, deian agertu diren aldagaiei asignatzen zaizkie

Memoria Nagusia

U	V	W
4	2	15

Faktoriala(U+V, W)

X	Fakt
?	720

Prozedura-dei baten exekuzioa (II)

a. Hasierako egoera deiaren aurretik.

Aldagaiak memoria nagusian daude
Parametro formalek ez dute baliorik

b. Prozedura-deia.

Tokia hartzen da prozeduraren parametroak eta
aldagai lokalak gordetzeko

Parametro errealak ebaluatzen dira, eta
bakoitza dagokion parametro formalari asignatzen
zaio.

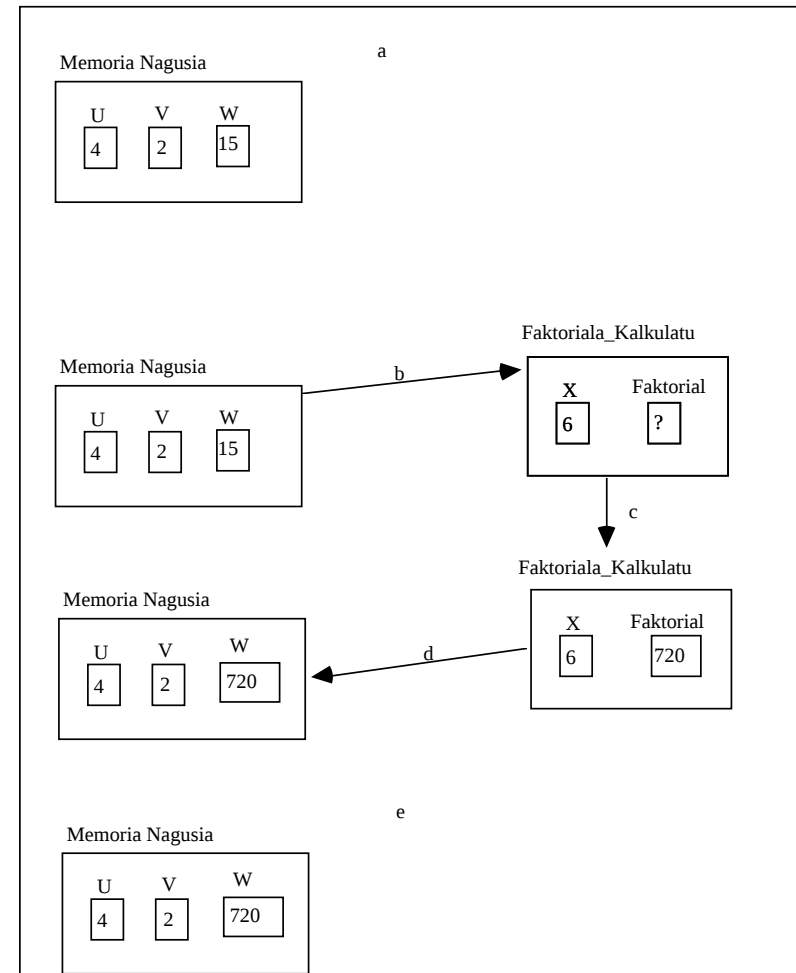
c. Azpiprogramaren exekuzioa.

d. Buelta programa nagusira.

Emaitza parametroen balioak deian agertu diren
aldagaiei asignatzen zaizkie.

e. Azpiprogramako objektuen desagertpena.

Prozedura-dei baten exekuzioa (III)



Noiz erabili Funtzioak edo prozedurak?

Eragiketak	Funtzioak	Prozedurak
Teklatutik irakurri	Ez	Bai
Pantailan idatzi	Ez	Bai
0 emaitza	Ez	Bai
emaitza 1	Bai	Bai
Emaitza anitz	Ez	Bai

2.6.4 Parametroak. Balioen pasatzea

➤ Parametroak: Azpiprograma eta programa nagusiaren artean datuak elkartrukatzeko balio dute

➤ VisualBasic-en bi mota daude:

- Sarrerakoak: programa nagusiak azpiprograma bati datuak pasa ahal izateko. Definizioan **ByVal** erabili.
- Sarrera-irteerakoak: sarrera bezala erabiltzeaz gain, parametro berdinean, azpiprogramak balezake emaitzaren bat bueltatzea programa deitzaileari.

- Adibidea:

Sarrera
↓

Sarrera-irteera
↙

Sub Absolutua(**ByVal** X as Integer, Abs As Integer)

‘Prozedura honek ezin dezake X parametroan idatzi.

‘Prozedura honek Abs parametroan idatzi dezake

- **Konbenioz:** azpiprogramaren definizioaren hasieran sarrerako parametroak definitu eta amaieran sarrera-irteerakoak
- Azpiprogramaren deian sarrerako parametroei balioak pasatzeko:
 - Balio konstantea: 5 3.4
 - Aldagai baten edukia: Zenb
 - Adierazpen bat: Zenb+5
- Azpiprogramak programa nagusiari emaitza bat bueltatzeko (irteera):
 - Aldagai bat

- Sarrerako datuak hasieran eta emaitzak bukaeran
- Adibidea:
 - Balio_Absolutua(-7, Zenb)

↑
datua

↑
emaitza
- Grafikoki:

-7 → X

Balio_Absolutua Abs

← Zenb
- Adibide gehiago:
 - Balio_Absolutua(A-B, Zenb)

↑
datua

↑
emaitza
 - Kalkulatu_Faktoriala(A-B, F)

↑
datua

↑
emaitza
 - Kalkulatu_Max(A, B, C, Max)

↙ ↗
datuak

↑
emaitza

► Definizio hau emanda:

```
- SUB Kalkulatu_Max(ByVal Zenb1 As Integer,
  ByVal Zenb2 As Integer, Max As Integer)
```

➤ Programa nagusian erabiltzeko:

A, B, M, Emaiza eta Y, prog. Nag-ko aldagaiak dira

```

Call Kalkulatu_Max(A, B, M)
           ↓      ↓      ↘
        Zenb1 Zenb2  Max

```

Call Kalkulatu_Max(5, B-2, Emaitzza)

Zenb1 *Zenb2* *Max*

➤ Definizio hau emanda:

- Function **Faktoriala** (ByVal X As Integer) As Long

Y = Faktoriala(5)

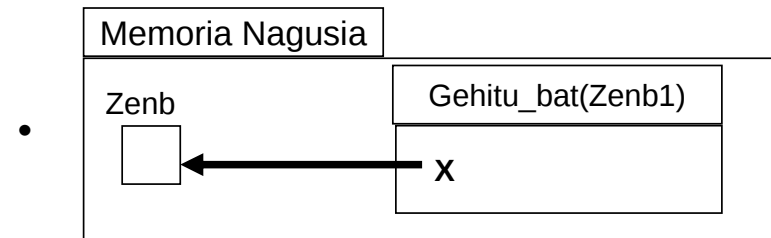


Zenb

➤ **Erreferentziaz:**

- Memoriako erreferentzia da kopiatzen dena. Hau da: deian erabilitako aldagaia eta parametroak gelaska berdina identifikatzen dute

Adib: Call Gehitu_bat(Zenb)

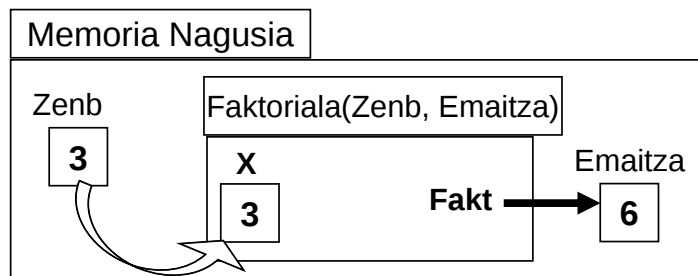


- Azpiprogramaren barruan **X** parametroa aldatzen bada, deian erabilitako **Zenb** aldagaia (programa nagusiko aldagaia) ere aldatuko da
- VBasic-en defektuz erreferentziaz pasatzen dira
- **NOTA: Taulak eta matrizeak BETI erreferentziaz pasatzen dira**

Parametroak erreferentziaz edo kopiaz pasatzea (II)

➤ Kopiaz:

- Azpiprogramaren deian erabilitako **aldagaiaren balioa parametrora kopiatzen** da
- Bi aukera:
 1. Deia egiterakoan, parentesiak jarri:
Call Faktoriala ((**Zenb**), Eraitza)
 2. Azpiprogramaren definizioan **ByVal** jarritz:
Sub Faktoriala(**ByVal** X As Integer, Fakt As Long)



- Azpiprogramak, deian erabilitako aldagaiak EZ aldatzea nahi dugunean erabiliko dugu. (Adibidean: X aldatzen bada, Zenb aldagaia ez da aldatzen)

2.6.5 Azpiprogramaren zehaztapena

Formalki:

➤ GOIBURUKOA + BALDINTZAK

GOIBURUKOA

➤ Hiru elementu nagusi dira:

1. Azpiprograma **mota**: *Function*, *Sub*
2. Azpiprogramaren **izena**: *Absolutua*
3. **Parametroen erazagupena** (aldagaien antzera):
 - a. Hasieran sarrerakoak eta ondoren sarrera-irteerakoak
 - b. Parametro bakoitzeko adierazi:
 - **Izena**: *Zenb*
 - **Mota**: *Integer*

➤ Adibidea:

Function Absolutua(ByVal Zenb As Integer) As Integer

Azpiprogramaren zehaztapena (II)

BALDINTZAK

- Bi baldintza adierazi (iruzkinak erabiliz):
- **Aurrebaldintza** (azpiprograma **exekutatu aurretik** betetzen den baldintza):
 - azpiprogramari pasatzen zaizkion datuek zer bete behar duten adierazten da
 - azpiprogramak datuak teklatutik irakurri behar dituenean hemen adierazten da
- **Postbaldintza** (azpiprograma **exekutatu ondoren** betetzen den baldintza):
 - emaitza-parametro bakoitzean zer lortu den adierazten da
 - azpiprogramak emaitzak pantailan idatziko dituenean hemen adierazten da

Funtzio baten zehaztapena (III)

➤ Adibideak:

```
Function Kalkulatu_Max(ByVal Z1 As Integer,  
    ByVal Z2 As Integer) As Integer  
REM Aurrebaldintza: Z1, Z2-k zenbaki oso bana  
REM                                     dute  
REM Postbaldintza: Z1 eta Z2 balioen  
REM                                     maximoa itzultzen du
```

```
Function Batura_Kalkulatu(T() As Integer) As  
    Integer  
REM Aurrebaldintza: T taulak 20 zenbaki oso ditu  
REM Postbaldintza: T taulako zenbakien batura  
REM                                     itzultzen du
```

Prozedura baten zehaztapena (III)

➤ Adibideak:

```
SUB Kalkulatu_Max(ByVal Z1 As Integer,
    ByVal Z2 As Integer, Z3 As Integer)
REM Aurrebaldintza: Z1, Z2-k osoko bana dute
REM Postbaldintza: Z1 eta Z2 balioen
REM          maximoa Z3-n itzultzen du

SUB Kalkulatu_Max_eta_Min(ByVal Zenb1 as
    Integer, ByVal Zenb2 as Integer, Zenb3
    As Integer, Max As Integer, Min As
    Integer)
REM Aurrebaldintza: 3 osoko ditugu
REM Postbaldintza: Max aldagaian Zenb1, Zenb2
REM          eta Zenb3 balioen maximoa dago.
REM          Min aldagaian Zenb1, Zenb2 eta
REM          Zenb3 balioen minimoa dago

SUB Batura_Kalkulatu(T() As Integer, Batura As
    Integer)
REM Aurrebaldintza: T taulak 20 osoko ditu
REM Postbaldintza: Batura parametroan T taulako
REM          elementuen batura dago
```

Prozeduren zehaztapenak. Ariketak

- Ondoko azpiprogramen zehaztapenak idatzi:
1. Zenbaki bat emanda, kapikua den esango duen azpiprograma
 2. Z zenbaki bat emanda, bere batukaria $(1+2+3+\dots+Z)$ kalkulatu duen azpiprograma
 3. Teklatutik 0-z bukatutako zenbakien sekuentzia dator. Azpiprogramak horietatik zenbat diren lehenak bueltatu behar du, eta gainera pantailan idatziko ditu.

Laburpena

- Azpiprograma baten definizio osoak ondoko elementuak ditu:
 - Azpiprogramaren **zehaztapena** (goiburukoa, aurrebaldintza eta postbaldintza)
 - Azpiprogramaren **aldagai lokalak**
 - Azpiprogramaren **aginduak**
- Bi parametro mota:
 - Sarrera: datuak pasatzeko
 - Sarrera-irteera: emaitzak bueltatzeko
 - **Konbenioz**: azpiprogramaren definizioan lehenengo sarrera parametroak jarri eta amaieran sarrera-irteerakoak
- Parametroen balio pasatzea
 - Erreferentziaz: memoriako helbidea kopiazen da (Kontuz!!!)
 - Kopiaz: balioa edo datua da kopiazen dena (**ByVal** erabili)

Adibideak. Prozeduren definizioak

```
SUB Kalkulatu_Max(ByVal Zenb1 AS INTEGER,  
                  ByVal Zenb2 AS INTEGER, Max AS INTEGER)  
REM Aurrebaldintza: 2 osoko ditugu  
REM Postbaldintza: Max aldagaian Zenb1 eta  
REM                      Zenb2 balioen maximoa dago  
    IF Zenb1 > Zenb2 THEN  
        Max = Zenb1  
    ELSE Max = Zenb2  
    END IF  
END SUB
```

```
SUB Faktoriala_Kalkulatu(ByVal X AS Integer,  
                        Fakt AS Long)  
REM Aurrebaldintza: X >= 0  
REM Postbaldintza: Fakt aldagaian X-en  
REM faktoriala dago  
DIM i AS Integer  
    REM aldagai lokala/lagungarria  
    Fakt = 1  
    FOR i= 1 TO X STEP 1  
        Fakt = Fakt * i  
    NEXT i  
END SUB
```

Adibideak. Prozeduren definizioak

```
SUB Irakurri_Data(Eguna AS INTEGER, Hila AS  
    INTEGER, Urtea AS INTEGER)  
REM Aurrebaldintza: hiru balio osoko daude  
REM      irakurtzeko, balioek data zuzena  
REM      adierazten dute(eguna, hila, urtea)  
REM Postbaldintza: balioak hiru aldagaitan  
    gorde dira  
    Eguna = InputBox("Sartu eguna");  
    Hila = InputBox("Sartu hilabetea")  
    Urtea = InputBox("Sartu urtea")  
END SUB  
  
SUB Bat_Gehitu(X AS INTEGER)  
    REM Postbaldintza: aldagaiaren balioa  
    REM      hasierakoa gehi bat da  
    X = X + 1  
END SUB
```

Beste adibide bat (I)

```
Private Sub Exekutatu_Click()  
  
    DIM A, B, C As Integer  
    DIM A_Lehena, BC_Lehena, Zenb_Lehena As Boolean  
  
    REM Programa nagusia hemen hasten da  
  
    A = InputBox("Sartu A zenbakia")  
    B = InputBox("Sartu B zenbakia")  
    C = InputBox("Sartu C zenbakia")  
    Call Esan_Lehena_Den (A, A_Lehena)  
    IF A_Lehena THEN  
        PRINT A; " Lehena da"  
    END IF  
  
    Call Esan_Lehena_Den(B+C, BC_Lehena)  
    Call Esan_Lehena_Den(1431, Zenb_Lehena)  
    IF BC_Lehena and Zenb_Lehena THEN  
        PRINT "biak dira lehenak "  
    ELSE  
        PRINT " biak ez dira lehenak"  
    END IF  
END Sub
```

Beste adibide bat (II)

```
SUB Esan_Lehena_Den(ByVal X As Integer,
                    Lehena AS Boolean)
    REM Aurrebaldintza: X positiboa da
    REM Postbaldintza: Lehena True da X lehena
    REM      bada eta False bestela

    DIM Zatitzaileak, K AS INTEGER
    IF X <= 3 THEN
        Lehena = TRUE
    ELSE
        Zatitzaileak = 0
        FOR K = 2 TO X \ 2 STEP 1
            IF (X mod K) = 0 THEN
                Zatitzaileak = Zatitzaileak+1
            END IF
        NEXT k
        IF Zatitzaileak = 0 THEN
            Lehena = TRUE
        ELSE
            Lehena = FALSE
        END IF
    END IF
END SUB
```

Azpiprogramak taula eta matrizeekin (I)

```
REM programa Adibidea
const N as Integer = 10,
      M as Integer= 20
DIM B1 (1 TO N) As Integer
DIM B2 (1 TO N) As Integer
DIM B1_PosMax, B2_PosMax As Integer
DIM Mat1 (1 TO N, 1 TO M) As Integer
DIM Mat2 (1 TO N, 1 TO M) As Integer
DIM Mat_Emaitza (1 TO N, 1 TO M) As Integer

SUB MaximoaLortu(B() As Integer,
                 PosMax As Integer)
    'Aurrebaldintza: B-k balioak sartuta ditu
    'Postbaldintza: PosMax bektoreko maximoaren
    '                posizioa da
    DIM K, Max As Integer

    Max = 1
    for K = 2 to N
        if B(K) > B(Max) then
            Max = K
        end if
    next k
    PosMax = Max
end SUB
```

Azpiprogramak taula eta matritzeekin (II)

```

Const N As Integer = 10,
      M As Integer = 20
SUB Irakurri_Bektorea(Bekt() As Integer)
    'Aurrebaldintza: 10 balio daude irakurtzeko
    'Postbaldintza: 10 balio irakurri dira eta
    '                      Bekt taulan(1xN) utzi dira
    DIM I AS Integer
    MsgBox ('Idatzi ' & N & ' balio osoko')
    FOR I = 1 to N
        Bekt(I) = InputBox(I & ". Zenbakia")
    NEXT I
END SUB

SUB Irakurri_Matrizea(Mat() As Integer)
    'Aurrebaldintza: 10x20 balio erreal daude
    '                      irakurtzeko
    'Postbaldintza: balioak irakurri dira eta
    '                      Mat matrizean(10x20) utzi dira
    DIM L, Z AS Integer
    MsgBox 'Idatzi '; N; ' bider '; M;
    ' balio errealen matrizea'
    FOR L = 1 to N
        FOR Z = 1 to M
            Mat(L, Z) = InputBox(I & ". Zenbakia")
        NEXT Z
    NEXT L
END SUB

```

Azpiprogramak taula eta matritzeekin(III)

```

Const N as integer = 10
SUB Batura_Kalkulatu(M1() AS integer,
                    M2() AS integer,
                    M3() AS integer)
    'Aurrebaldintza: M1 eta M2-k balioak sartuta dauzkate
    '                      eta NxN-ko matrizeak dira
    'Postbaldintza: M3 matrizeen batura da, NxN-koa
    DIM L, Z AS Integer
    FOR L = 1 to N
        FOR Z = 1 to N
            M3(L,Z) = M1(L,Z) + M2(L,Z)
        NEXT Z
    NEXT L
END SUB

REM programa nagusiaren gorputza
Call Irakurri_Bektorea(B1)
Call Maximoa_Lortu(B1, B1_PosMax)
Call Irakurri_Bektorea(B2)
Call Maximoa_Lortu(B2, B2_PosMax)
...
Call Irakurri_Matrizea(Mat1)
Call Irakurri_Matrizea(Mat2)
Call Batura_Kalkulatu(Mat1, Mat2, Mat_Emaitza)
Call Idatzi_Matrizea(Mat_Emaitza)

```