

2.4 Egitura errepikakorrak

For, while, do ...loop

1

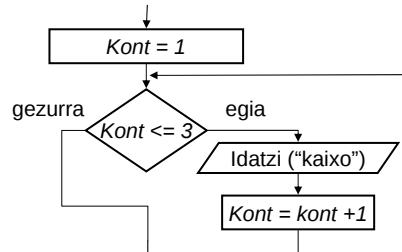
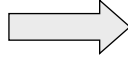
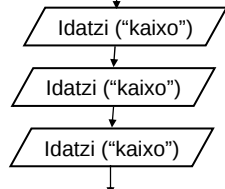
2.4. FOR, WHILE, DO..LOOP

- Orain arte ikusi diren programetan agindu bakoitza 1 edo 0 aldiz exekutatu zitekeen
- Kasu askotan aginduak errepikatzea interesatzen zaigu
 - datu berdin asko tratatu
 - gauza bat askotan errepikatu
- Ikasle baten bi noten batez bestekoa kalkulatzeko:
N1 = InputBox("Eman 1. ikasgaiaren nota: ")
N2 = InputBox("Eman 2. ikasgaiaren nota: ")
Print ((N1+N2)/2)
- 100 ikasleren noten batez bestekoa kalkulatzeko?

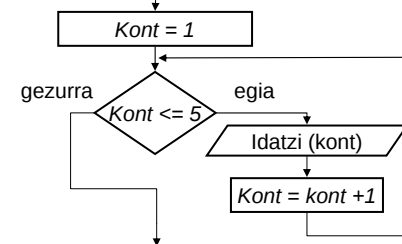
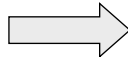
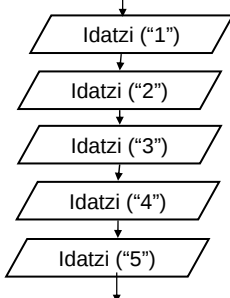
2

Errepikapena. Adibideak

1. Idatzi kaixo 3 aldiz



2. Idatzi 1-etik 5-rako zenbakiak

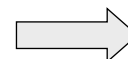


3

Errepikapena. Adibideak: FOR

1. Idatzi kaixo 3 aldiz

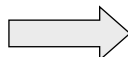
Print "kaixo"
Print "kaixo"
Print "kaixo"



FOR kont=1 to 3
 Print "kaixo"
NEXT kont

2. Idatzi 1-etik 5-rako zenbakiak

Print "1"
Print "2"
Print "3"
Print "4"
Print "5"



FOR kont=1 to 5
 Print kont
NEXT kont

4

Oinarrizko algoritmo egitura

- Ikusi 2.2.1. Algoritmoak
 - Oinarrizko algoritmo egitura (12)

5

2.4.1. FOR. Sintaxia

➤ **For** agindua::=
 for <indizea> = <hasi_balio> **to** <amaia_balio> [**step** exp]
 <agindua>

 next indizea

➤ **Oharrak:**

<indizea>: beti aldagai bat izango da

<hasi_balio>: zenbaki oso bat aldagai, literal edo adierazpen moduan

<amaia_balio>: zenbaki oso bat aldagai, literal edo adierazpen moduan

<exp>: zenbaki oso bat aldagai, literal edo adierazpen moduan

<agindua>: orain arte ikusitako edozein: esleipen-sententzia, baldintza egitura (if-then-else edo select case) edo egitura errepikakorra (for, while, do..loop)

6

FOR. ADIBIDEAK

```
Batukaria = 0;
for K = 1 to 10 step 1
    Batukaria = Batukaria+K
next K
```

$$\left\{ \text{Batukaria} = \sum_{K=1}^{10} K \right\}$$

```
Batukaria = 0;
for K = N downto 1
    Batukaria = Batukaria+K;
next K
```

$$\left\{ \text{Batukaria} = \sum_{K=N}^1 K \right\}$$

```
Batuz = 0;
for K = 1 to 10 step 1
    Batuz = Batuz+K
    print K, " pausoan "
    print "progresio "
    print "aritmetikoa: "
    print Batuz
next K
```

7

FOR-TO. SEMANTIKA

- Indize bat eta honek begiztaren pasada bakoitzean zeintzuk balio hartuko dituen adierazi behar da
- $\langle \text{hasi_balio} \rangle \leq \langle \text{amaia_balio} \rangle$, orduan step en dagoen balioa handituz joan behar du indizea. Begiztaren pasada bakoitzean indizearen balioa automatikoki "handitzen" da, *exp* -en adierazten den moduan hurrengo balioa hartuz
- $\langle \text{hasi_balio} \rangle > \langle \text{amaia_balio} \rangle$, orduan for-agindua amaitzen da eta ondorengo agindua exekutatzen da goaz. Hots, for gorputzeko aginduak ez dira behin ere egikaritu

KONTUZ! Indizearen balioak automatikoki aldatzen dira for-aren gorputzean. Erabiltzaileak EZIN DITU aldatu

8

2.4.2. WHILE

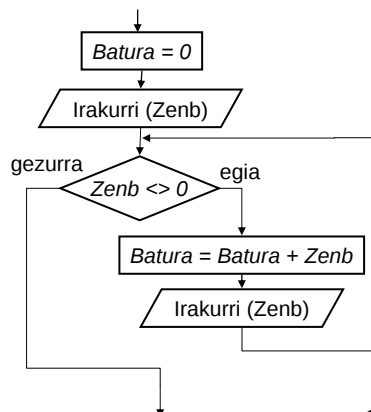
- Errepikapen kopurua finitua ez denean, zer?
Orduan, bestelako baldintza bat asmatu behar
- Adibidea: Teklatu bidez irakurritako zenbaki osoen sekuentzia baten batura kalkulatu. Sekuentziaren amaiera 0 zenbakiarekin adierazten da.
 - Analisia:
 - ❖ **Sarrerako datuak:** zenbaki sekuentzia bat
 - ❖ **Irteerako datuak:** batura, sekuentziako zenbakiena
 - ❖ **Prozesua:** Zenbaki bat irakurri eta aurreko baturari batu, beste zenbaki bat irakurri eta aurrekoari batu, ...
 - ❖ **Noiz arte??:** sarreran irakurritako zenbakia = 0 izan arte

9

Adibidearen soluzioa algoritmikoki

- Sarrerako datuen kasuak aztertu:
 - Sekuentzia hutsa izan daiteke (zenbaki bakarra eta hura 0)
 - Sekuentzia ez-hutsa, gutxienez zenbaki bat 0-ren ezberdina
 - Beraz, prozesua 0 edo n aldiz errepikatu daiteke (n sekuentziaren luzera izanik, 0 kontutan hartu gabe)

<i>Batura = 0</i>
<i>Irakurri Zenb</i>
bitartean Ez da sekuentziaren amaiera egin
<i>Batura = Batura + Zenb</i>
<i>Irakurri Zenb</i>
ambitartean



10

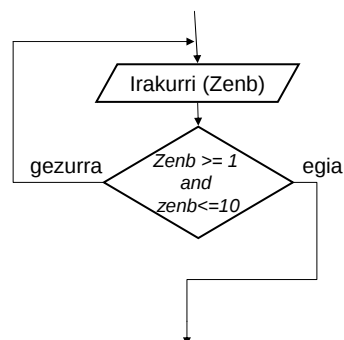
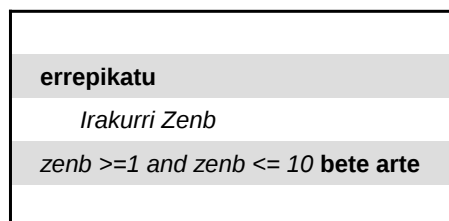
2.4.2. DO... LOOP

- **Errepikapen kopurua finitua ez denean, zer? Orduan, bestelako baldintza bat asmatu behar**
- **Adibidea:** Teklatu bidez zenbaki bat sartu eta errepikatu datu-sarrera sartutako zenbakia 1 eta 10 artean ez badago. Amaieran, emandako zenbaki ona idatzi.
 - **Analisia:**
 - ❖ **Sarrerako datuak:** zenbaki bat
 - ❖ **Irteerako datuak:** sarrerako zenbaki ona
 - ❖ **Prozesua:** Zenbaki bat irakurri, begiratu onargarria den, onargarria ez bada aurreko bi pausoak errepikatu, onargarria bada prozesua amaitu pantailan zenbakia idatziz
 - ❖ **Noiz arte??:** sarreran irakurritako zenbakia 1 eta 10 artean egon arte

11

Adibidearen soluzioa algoritmikoki

- **Sarrerako datuen kasuak aztertu:**
 - Zenbakia ez da zuzena (negatiboak, 0 eta 10 baino handiagoak)
 - Zenbakia zuzena da (1 eta 10 artekoak)



12

Oinarrizko algoritmo egiturak

- Ikusi 2.2.1. Algoritmoak
 - Oinarrizko algoritmo egitura (10)
 - Oinarrizko algoritmo egitura (11)

13

2.4.2. WHILE + DO... LOOP (Vbasic)

- **WHILE** agindua::=
 while <baldintza>
 <agindua>
 wend
- **do ... loop** agindua::=
 do
 <agindua>
 loop {while/until} <baldintza> ;
- Ikusi Oinarrizko algoritmoen egiturak:
 while (9), do..loop until (10), do..loop while (11)

14

WHILE+ DO...LOOP. SEMANTIKA

WHILE:

- Baldintza ebaluatu
 - ❖emaitza faltsua bada (False), bukatu eta jarraitu while aginduaren ondoren dagoen aginduarekin
 - ❖emaitza egiazkoa bada (True), orduan while aginduaren barneko agindu-multzoa exekutatu

DO...LOOP WHILE:

- Do...loop-aginduaren gorputzeko agindu multzoa exekutatu
- Baldintza ebaluatu
 - ❖emaitza faltsua bada (False), bukatu eta do...loop aginduaren ondoren dagoen aginduarekin jarraitu
 - ❖emaitza egiazkoa bada (True), orduan do...loop aginduaren barneko agindu-multzoa egikaritu

15

WHILE+ DO...LOOP. SEMANTIKA

DO...LOOP UNTIL:

- Do...loop-aginduaren gorputzeko agindu multzoa exekutatu
- Baldintza ebaluatu
 - ❖emaitza egiazkoa bada (True), bukatu eta do...loop aginduaren ondoren dagoen aginduarekin jarraitu
 - ❖emaitza faltsua bada (False), orduan do...loop aginduaren barneko agindu-multzoa egikaritu

16

BEGIZTEN DISEINUA

- Begiztaren kontrola
 - Noiz bukatuko da errepikapena? Bukaerako baldintza
 - Nola eguneratuko da baldintza?
 - Nola hasieratu baldintza
- Begiztaren prozesua
 - Zer errepikatu behar da?
 - Nola hasieratu?
- Diseinuaren zuzentasunaren konprobazioa trazten bidez

17

WHILE. ADIBIDEAK

Zenbaki bat irakurri eta kalkulatu 2^K moduko lehen berredura, zenbakia-ren berdina edo handiagoa dena.

REM Zenbakia > 1

Berredura = 1; $2^0 = 1$

while Berredura < Zenbakia

 Berredura = Berredura*2

wend

print "Lehen berredura handiena: "

print Berredura

Puntu karaktereaz amaitzen den karaktere sekuentzia bat irakurri eta kontatu zenbat karaktere dituen sekuentziak

Kop =0

input Kar

while Kar <> '.'

 Kop = Kop + 1

 input Kar

wend

print "karaktereak "

print "guztira: ", Kop

18

DO...LOOP. ADIBIDEAK

Zenbaki bat irakurri eta kalkulatu 2^K moduko lehen berredura, zenbakia-ren berdina edo handiagoa dena.

REM Zenbakia > 1

Berredura=1 ' $2^0 = 1$

do

 Berredura = Berredura*2

loop until Berredura>=Zenbakia

print "Berredura hurbilena: "

print Berredura

Idatzi programa bat prozesua errepikatu nahi duen galdetzen duena, erantzuna BAI den kasuan prozesua errepikatzen da eta bestela, programa amaitzen da.

Do

 erantzuna = InputBox _

 ("errepikatu nahi duzu?")

loop while erantzuna = "BAI"

19

KONTROL-EGITURA ERREPIKAKORRAK

Noiz zein erabili?

Zenbat aldiz errepikatu behar den ezaguna denean
for erabili.

Kontrolerako aldagaiak zenbaki oso bat izan behar du

Do-loop egitura begizta gutxienez behin errepikatu behar denean

Gainontzeko kasuetan **while** egitura erabili

20