

6-PROGRAMAZIO METODOLOGIA Programazio Modularra

- 1) Beharra
- 2) Abantailak
- 3) Adierazpidea
- 4) Adibideak
- 5) **Exekuzioa**

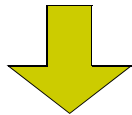


http://commons.wikimedia.org/wiki/File:Child_4_cubes.jpg



1.2 BEHARRA Kontraesana

- Berreketa ebazten duen algoritmoa sortu badugu, ezin al dugu berrerabili?



- Bai. Zenbait gauza aldatuz.
- Datu sarrera eta datu irteera. Programen arteko datu trukaketa.

1.1 BEHARRA Algoritmoen murrizpenak

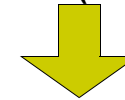
- Algoritmoetan eragile matematiko hauek erabili ditzakegu:

- -, +, *, /, %

- Ezin ditugu berreketak erabili

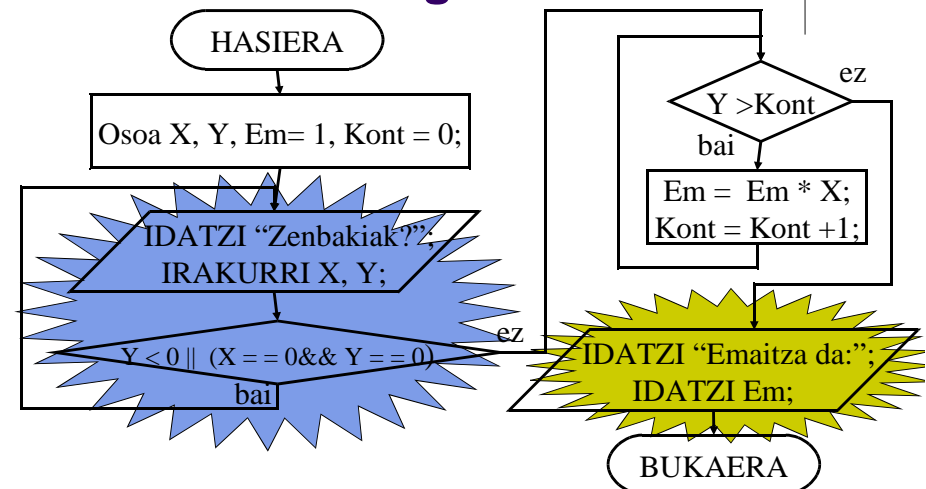
- $10^p + 1 > X \ \&\& \ X \geq 10^p$

- $(X \% 10^j) / 10^{j-1} == (X \% 10^k) / 10^{k-1}$

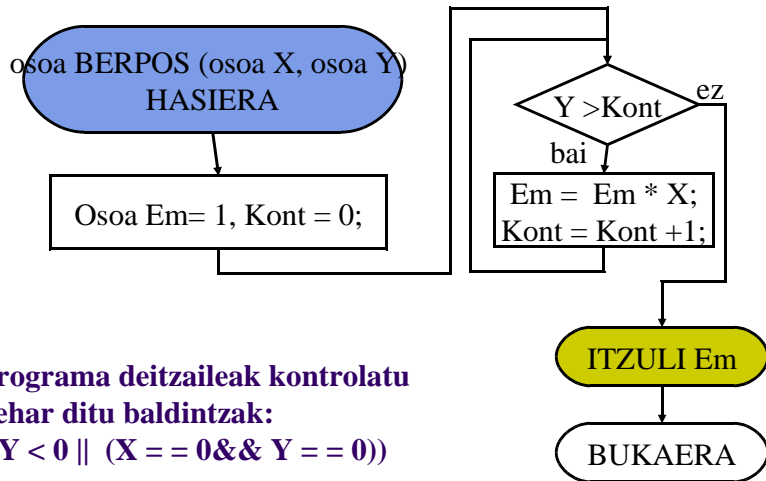


- Berreketa eragilerik gabe algoritmoak konplikatu egiten zaizkigu

1.3 BEHARRA BERREKETA algoritmoa

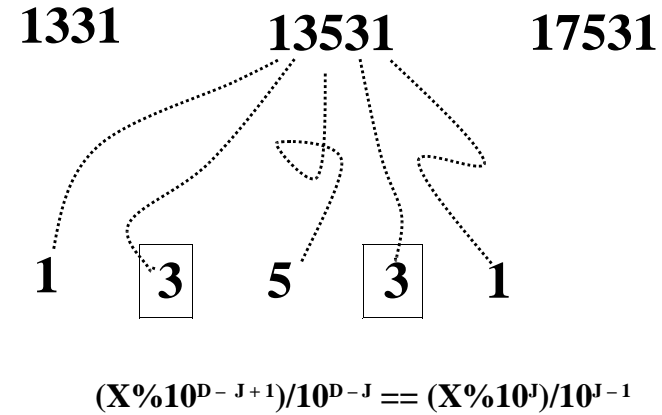


1.4 BEHARRA BERPOS azpiprograma



Programa deitzaileak kontrolatu behar ditu baldintzak:
 $!(Y < 0 \parallel (X == 0 \& \& Y == 0))$

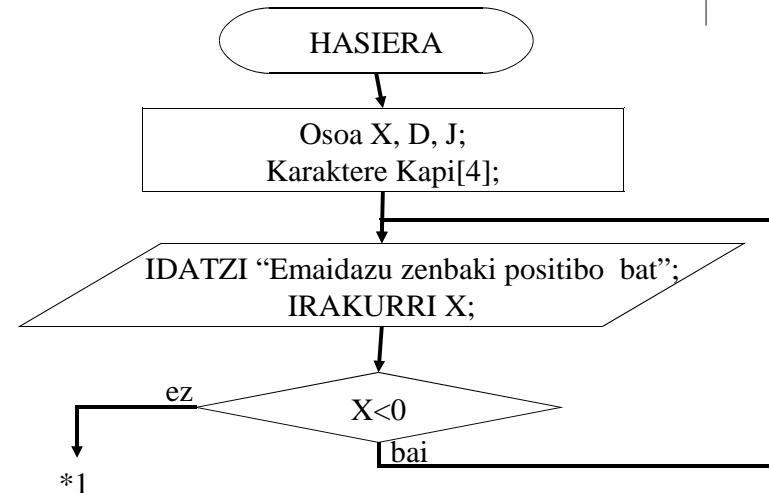
1.5 BEHARRA Kapikua



1.6 BEHARRA Kapikua. BERPOS azpiprogramaz

- Aurrebaldintza:
 - Zer da: **X: zenbakia**
 - Mota: **$X \in \mathbb{Z}$**
 - Baldintza indibidualak: **$X \geq 0$**
 - Erlazioak:
 - Ondorengo baldintza:
 - Zer da: **Kapi: mezu bat "BAI" ala "EZ" adieraziz**
 - Mota: **Kapi esaldia (karaktere Kapi[4])**
 - Baldintza indibidualak:
 - Erlazioak:
- Bada D, X-ren digitu kopurua ($10^D > X \geq 10^{D-1}$). J-k hartu ditzala jarraian $J = 1 \dots D$ balioak.
- J balio guztietan $(X \% 10^{D-J+1}) / 10^{D-J} == (X \% 10^J) / 10^{J-1} \rightarrow$ Kapi == "BAI"
 - Bestela \rightarrow Kapi == "EZ"

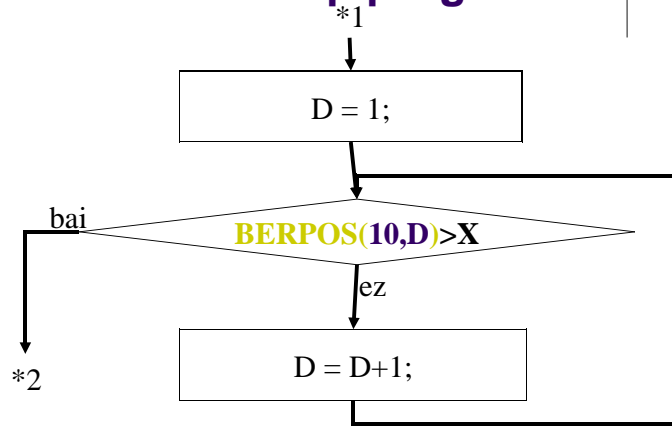
1.7 BEHARRA Kapikua. BERPOS azpiprogramaz



1.8 BEHARRA

Kapikua. BERPOS azpiprogramaz

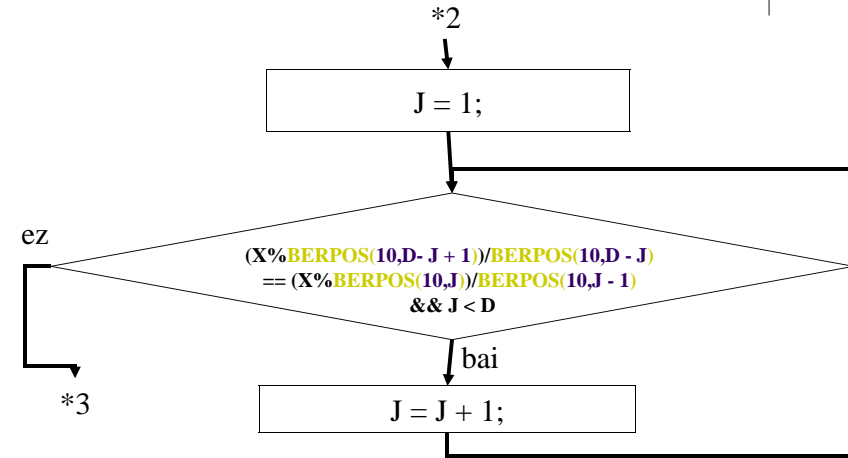
6
Programazio
modularra



1.9 BEHARRA

Kapikua. BERPOS azpiprogramaz

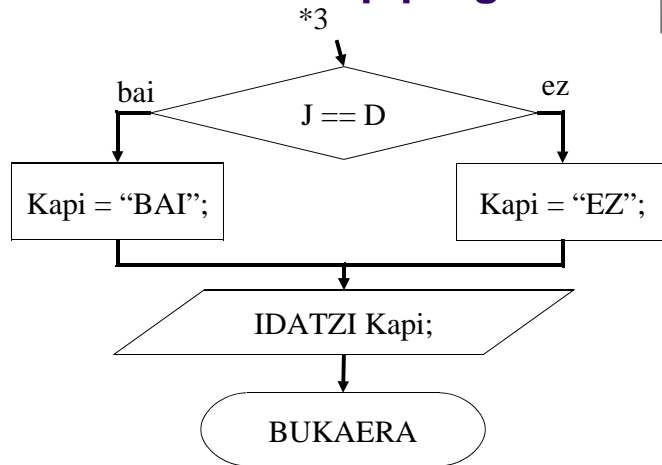
6
Programazio
modularra



1.10 BEHARRA

Kapikua. BERPOS azpiprogramaz

6
Programazio
modularra



2.1 ABANTAILAK

6
Programazio
modularra

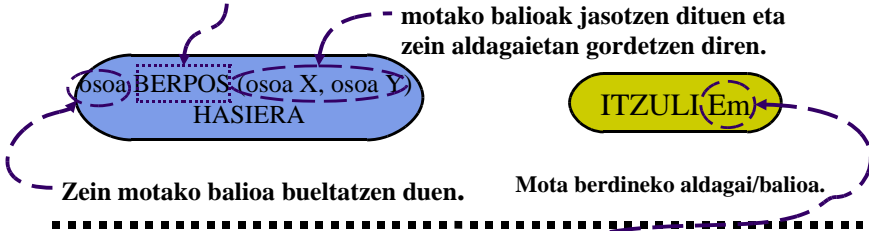
- Arazo konplexuak zatituz erraztu
 - Abstrakzioaz ulergarritasuna lortu
 - Zati bakoitza independente sortu eta probatu
- Azpiprogramak berrerabili
 - Programatzen denbora aurreztu
 - Akats gutxiago (proba gehiago)
 - Parametroen bitartez ebazpen partikularrak lortu
 - Berbidura → **BERPOS (?, 2)**
 - $(2^3)^4$ → **BERPOS(BERPOS (2,3), 4)**

3.1 ADIERAZPIDEA

Fluxu diagraman

Azpiprogramaren izena

PARAMETRO FORMALAK. Zein motako balioak jasotzen dituen eta zein aldagaietan gordetzen diren.



Zein motako balioa bueltatzen duen.

Mota berdineko aldagai/balioa.



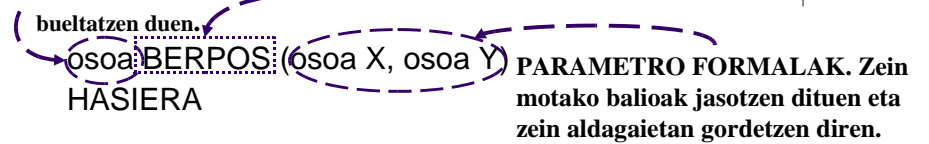
Azpiprogramari dei egitea

PARAMETRO ERREALAK. Prozesaketarako balio zehatzak. Parametro formalen motakoak.

3.2 ADIERAZPIDEA

Sasikodean

Zein motako balioa bueltatzen duen.



HASIERA

...

ITZULI (Em);

...

BUKAERA/

...



...

Mota berdineko aldagai/balioa.

Azpiprogramari dei egitea

PARAMETRO ERREALAK. Prozesaketarako balio zehatzak. Parametro formalen motakoak.

3.3 ADIERAZPIDEA

Kontutan izan

- Parametro formalak adieraztea erazagupen modukoa da
- Parametro formalak eta azpiprogramaren aldagaiak azpiprograman bakarrik existitzen dira
- Programa deitzailearen aldagaiak ez dira ikusten azpiprograman, beraz, programa deitzaileak eta azpiprogramak izen berdineko aldagaiak izan ditzakete
- Taulak ez dira parametro modura pasatzen, informazio trukaketan denbora eta memori asko erabiltzen delako. Taulari apuntatzen dion erakuslea pasa behar zaio azpiprogramari

4.1 ADIBIDEAK

Posizioa azpi-algoritmoa

- *Esaldi batetan non agertzen den karaktere zehatz bat*
- Zehaztapena:
 - Aurrebaldintza:
 - Zeintzuk dira:
 - erak: karaktere kateari erakuslea
 - kar: aurkitu beharreko karakterea
 - topea: gehienez zenbat posizioetan bilatu
 - Mota:
 - karakterea *erak;
 - karakterea kar;
 - osoa topea; topea ∈ Z
 - Baldintza indibidualak: topea ≥ 0
 - Erlazioak:

4.2 ADIBIDEAK

Posizioa azpi-algoritmoa

- Zehaztapena:

- Ondorengo baldintza:

- Zer da: **Em: zein posiziotan dagoen kar *erak apuntatzen duen karaktere katean**

- Mota: **Em ∈ Z**

- Baldintza indibidualak: **Em ≥ -1**

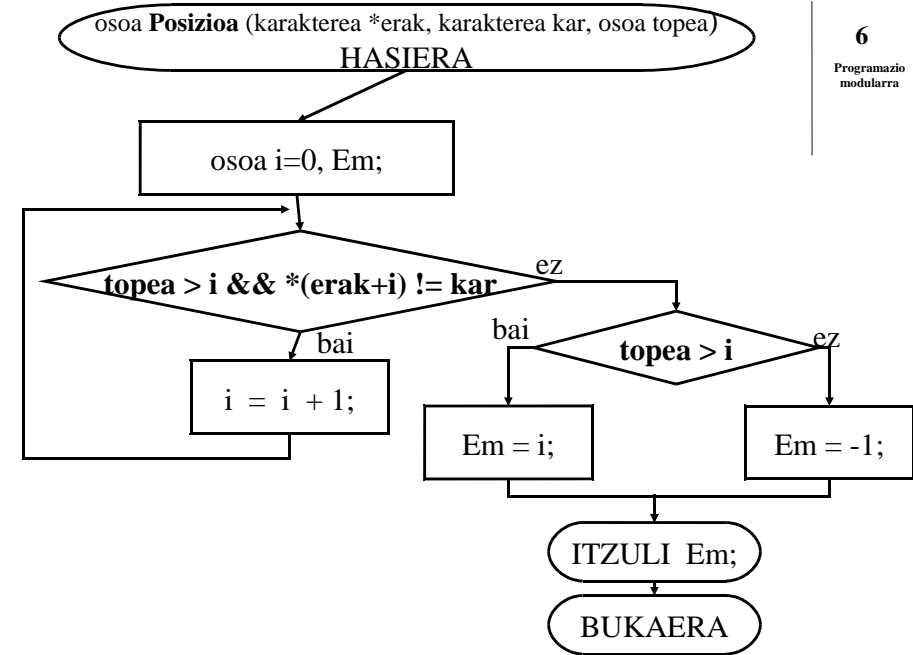
- Erlazioak:

- Existitzen bada i balioa betetzen duena $*(erak+i) == kar$ eta $i \geq 0$ eta $i < Topea$ eta beste k balio guztiak betetzen badute $0 \leq k < i$ eta $*(erak+k) \neq kar$ → Em == i

- Bestela → Em == -1

6

Programazio modularra



6

Programazio modularra

4.4 ADIBIDEAK

Kalkulagailua

- Bi zenbaki osoen kalkulagailu bat (batu, kendu eta biderkatu)

- Kalkulagailuaren eragiketa bakoitza azpi-problema bat

- Programa nagusia beste bat

6

Programazio modularra

4.5 ADIBIDEAK

Kalkulagailua. Algoritmo nagusia

hutsa Kalkulagailua ()

HASIERA

osoa Auk, X, Y, Z;
EGIN

EGIN

1 IDATZI ("1. Batu; 2. Kendu; 3. Biderkatu; 4. Irten");

2 IDATZI (" Zer egin nahi duzu?");

3 IRAKURRI Auk;

4 DENBITARTEAN (Auk < 1 || Auk > 4);

5 IDATZI ("Ze bi zenbaki osoekin?");

6 IRAKURRI X, Y;

BALDIN (Auk)

7 BADA 1:

Z = Batura (X, Y);

IDATZI Z;

IRTEN;

11 BADA 2:

Z = Kendura (X, Y);

IDATZI Z;

IRTEN;

15 BADA 3:

Z = Biderkadura (X, Y);

IDATZI Z;

IRTEN;

19 BESTELA:

IDATZI "Eskerrik asko kalkulagailua erabiltzeagatik";

BUK_BALDIN;

DENBITARTEAN (Auk != 4);

BUKAERA

4.6 ADIBIDEAK

Kalkulagailua. Batura algoritmoa

6
Programazio
modularra

1 osoa Batura (osoa A, osoa B)

HASIERA

osoa Em;

2 $Em = A + B$;

3 ITZULI Em;

BUKAERA

4.7 ADIBIDEAK

Kalkulagailua. Kendura algoritmoa

6
Programazio
modularra

1 osoa Kendura (osoa C, osoa D)
HASIERA

osoa Eaitza;

2 Eaitza = Batura (C, - D);

3 ITZULI Eaitza;

BUKAERA

4.8 ADIBIDEAK

Kalkulagailua. Biderkadura algoritmoa

6
Programazio
modularra

1 osoa Biderkadura (osoa F, osoa G)

HASIERA

2 Osoa $Em = 0$;

3 BADA ($F < 0$)

4 $F = -F$;

5 $G = -G$;

BUK_BADA

6 DENBITARTEAN ($F > 0$)

7 $Em = \text{Batura}(Em, G)$;

8 $F = F - 1$;

BUK_DENBITARTEAN

9 ITZULI Em;

BUKAERA

5.1 EXEKUZIOA

Kalkulagailua. Simulazio taula (i)

6
Programazio
modularra

- Erabiltzaileak -4, 1, 2, 3, 4, 5 eta 6 sakatzen du.

#	Auk	X	Y	Z	Batura A	Batura B	Batura Em	Pant.
1	?	?	?	?				1. ...
2	?	?	?	?				Zer ...
3	-4	?	?	?				
4	-4	?	?	?				
1	-4	?	?	?				1. ...
2	-4	?	?	?				Zer ...

5.2 EXEKUZIOA

Kalkulagailua. Simulazio taula (ii)

6
Programazio
modularra

#	Auk	X	Y	Z	Batura A	Batura B	Batura Em	Pant.
3	1	?	?	?				
4	1	?	?	?				
5	1	?	?	?				Ze bi.
6	1	2	3	?				
7	1	2	3	?				
8	1	2	3	?				
Batura 1	1	2	3	?	2	3	?	

5.3 EXEKUZIOA

Kalkulagailua. Simulazio taula (iii)

6
Programazio
modularra

#	Auk	X	Y	Z	Batura A	Batura B	Batura Em	Pant.
Batura 2	1	2	3	?	2	3	5	
Batura 3	1	2	3	?	2	3	5	
8	1	2	3	5				
9	1	2	3	5				5
10	1	2	3	5				
21	1	2	3	5				
1	1	2	3	5				1. ...

5.4 EXEKUZIOA

Kalkulagailua. Simulazio taula (iv)

6
Programazio
modularra

#	Auk	X	Y	Z	Batura A	Batura B	Batura Em	Pant.
2	1	2	3	5				Zer ...
3	4	2	3	5				
4	4	2	3	5				
5	4	2	3	5				Ze bi.
6	4	5	6	5				
7	4	5	6	5				
11	4	5	6	5				

5.5 EXEKUZIOA

Kalkulagailua. Simulazio taula (v)


6
Programazio
modularra

#	Auk	X	Y	Z	Batura A	Batura B	Batura Em	Pant.
15	4	5	6	5				
19	4	5	6	5				
20	4	5	6	5				Esk...
21	4	5	6	5				

5.6 EXEKUZIOA

Simulazio taulak

6
Programazio
modularra

- Abantaila: “Zati bakoitza independente sortu eta probatu”
 - Azpi-algoritmoa bere aldetik probatu. Itxuraz ondo badabil ...
- 
- ...algoritmo deitzaileen simulazio tauletan ez dugu azpi-algoritmoa aginduz-agindu exekutatu behar, ondo dabilela suposatu eta emaitza erabili.

5.7 EXEKUZIOA

Biderkadura. Simulazio taula I (i)

6
Programazio
modularra

- Kasu orokorra: Biderkadura $(-2, 3) == -6$

#	F	G	Em	Pant.
1	-2	3	?	
2	-2	3	0	
3	-2	3	0	
4	2	3	0	
5	2	-3	0	
6	2	-3	0	

5.8 EXEKUZIOA

Biderkadura. Simulazio taula I (ii)

6
Programazio
modularra

#	F	G	Em	Pant.
7	2	-3	-3	
8	1	-3	-3	
6	1	-3	-3	
7	1	-3	-6	
8	0	-3	-6	
6	0	-3	-6	
9	0	-3	-6	

5.9 EXEKUZIOA

Biderkadura. Simulazio taula II (i)

6
Programazio
modularra

- Kasu orokorra: Biderkadura $(-2, -3) == 6$

#	F	G	Em	Pant.
1	-2	-3	?	
2	-2	-3	0	
3	-2	-3	0	
4	2	-3	0	
5	2	3	0	
6	2	3	0	

5.10 EXEKUZIOA

Biderkadura. Simulazio taula II (ii)

6
Programazio
modularra

#	F	G	Em	Pant.
7	2	3	3	
8	1	3	3	
6	1	3	3	
7	1	3	6	
8	0	3	6	
6	0	3	6	
9	0	3	6	

5.11 EXEKUZIOA

Biderkadura. Simulazio taula III (i)

6
Programazio
modularra

- Kasu orokorra: Biderkadura $(2, -3) == -6$

#	F	G	Em	Pant.
1	2	-3	?	
2	2	-3	0	
3	2	-3	0	
6	2	-3	0	

5.12 EXEKUZIOA

Biderkadura. Simulazio taula III (ii)

6
Programazio
modularra

#	F	G	Em	Pant.
7	2	-3	-3	
8	1	-3	-3	
6	1	-3	-3	
7	1	-3	-6	
8	0	-3	-6	
6	0	-3	-6	
9	0	-3	-6	

5.13 EXEKUZIOA

Biderkadura. Simulazio taula IV (i)

6
Programazio
modularra

- Kasu orokorra: Biderkadura $(2, 3) == 6$

#	F	G	Em	Pant.
1	2	3	?	
2	2	3	0	
3	2	3	0	
6	2	3	0	

5.14 EXEKUZIOA

Biderkadura. Simulazio taula IV (ii)

6
Programazio
modularra

#	F	G	Em	Pant.
7	2	3	3	
8	1	3	3	
6	1	3	3	
7	1	3	6	
8	0	3	6	
6	0	3	6	
9	0	3	6	

5.15 EXEKUZIOA

Biderkadura. Simulazio taula V

6
Programazio
modularra

- Kasu berezia: Biderkadura $(0, 3) == 0$

#	F	G	Em	Pant.
1	0	3	?	
2	0	3	0	
3	0	3	0	
6	0	3	0	
9	0	3	0	

5.16 EXEKUZIOA

Biderkadura. Simulazio taula VI (i)

6
Programazio
modularra

- Kasu berezia: Biderkadura $(-2, 0) == 0$

#	F	G	Em	Pant.
1	-2	0	?	
2	-2	0	0	
3	-2	0	0	
4	2	0	0	
5	2	0	0	
6	2	0	0	

5.17 EXEKUZIOA

Biderkadura. Simulazio taula VI (ii)

6
Programazio
modularra

#	F	G	Em	Pant.
7	2	0	0	
8	1	0	0	
6	1	0	0	
7	1	0	0	
8	0	0	0	
6	0	0	0	
9	0	0	0	

5.18 EXEKUZIOA

Biderkadura. Simulazio taulak

- Kasu bereziak eta orokor batzuk probatuta ondo dabilela dirudi...
- ...suposatuz *Batura* azpialgoritmoa ondo dabilela
- *Biderkadura* azpialgoritmoa erabiltzen duten algoritmoek emaitza egokia bueltatzen duela suposatu dezakete