

Oinarrizko Prog.

3. laborategia. Iterazioa eta Sekuentziak

Izena:___ Lucia eta Ainhoa__Data:_2023/09/26_____

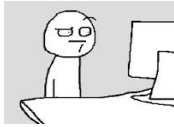
Laborategi honetan iterazioak eta sekuentziak landuko dira. Laguntza modura, hemen dituzue erabili ditzakegun iterazio-egiturak.

Aldagaien hasieraketa;
bitartean (baldintza) egin
 errepikatzeko eragiketak;
 aldagaien eguneraketa;
ambitartean;

egin I guztietarako 1 tik 30eraino
 errepikatzeko eragiketak;
amguztietarako;

Aldagaien hasieraketa;
errepikatu
 errepikatzeko eragiketak;
 aldagaien eguneraketa;
atera baldin (baldintza)
amaitu_errepikatu

Aldagaien hasieraketa;
errepikatu (baldintza) bete arte
 errepikatzeko eragiketak;
 aldagaien eguneraketa;
amaitu_errepikatu



1. Ariketa

Partxisean fitxa bat etxetik ateratzeko algoritmoa idatzi ezazue. Fitxa bat etxetik aterako da baldin eta dado bat botatzean 5 bat ateratzen dugun. Ez bada 5 bat lortzen berriro botako dugu dadoa, gehienez 4 aldiz bota arte. Algoritmoa garatzeko hurrengo azpiprogramak erabil daitezke: *fitxa_etxetik_atera()*, *dadoa_bota()* (*dadoa_bota()* azpiprogramak balio bat eskatuko dio erabiltzaileari).

1. Espezifikazioa:

Sarrera: balio bat (inoiz ez 4 aldi baino gehiago)

Aurre: balioak 0 eta 6 artean egongo dira

Irteera: mezu bat

Post: fitxa aterako da saiatu gaitezken 4 saiakeraren artean behin 5 bat lortuz gero. 4 saiakera agortuz gero mezu bat inprimatuko da “sentitzen dut, saiakerak agortu dituzu”

2. Proba kasuak

Sarrera: 5

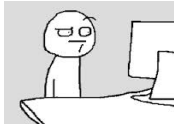
Irteera: fitxa aterako da

Sarrera: 2,6,1,2

Irteera: fitxak etxean jarraitzen du, eta mezua idatziko da.

3. Algoritmoa

1. Zenbakia, aldia: integer;
2. Aldia $\leftarrow 1$;
3. Errepikatu aldia = 4 || zenbakia = 5 bete arte
4. zenbakia \leftarrow dadoa_bota();
5. Aldia \leftarrow aldia + 1;
6. Amaitu_errepikapena;
7. Baldin (zenbakia = 5) orduan
8. fitxa_etxetik_atera();
9. Bestela
10. idatzi(“fitxak etxean jarraitzen du, eta mezua idatziko da”);
11. Amaitu_baldin.



2. Ariketa (ariketa hau ebatzita dago, ulertu baino ez duzue egin behar)

Programa honen helburua zenbaki baten biderketa taula lortzea da. Erabiltzaileari eskatutako N zenbaki jakin bat edukita ($0 < n \leq 10$) bere biderketa taula kalkulatzeko programa sortu

1. Espezifikazioa:

Sarrera: Zenbaki oso bat

Aurre: $0 < n$:balioa ≤ 10

Irteera: 10 zenbaki

Post: $0 < n$ -ren biderketa bakoitza ≤ 10

2. Proba kasuak

Sarrera: 4

Irteera: 4, 8, 12, 16, 20, 24, 28, 32, 36, 40

Sarrera: 6

Irteera: 6, 12, 18, 24, 30, 36, 42, 48, 54, 60

3. Algoritmoa:

emaitza, biderkatzailea, n: integer; -

hasiera

irakurri(n);

biderkatzailea \leftarrow 1; --*memoria erresebatzen*

emaitza \leftarrow 0;

errepikatu biderkatzailea = 11 **bete arte** --*bukatzeko baldintza*

emaitza \leftarrow n * biderkatzailea;

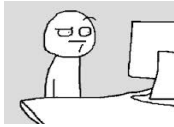
idatzi(emaitza);

biderkatzailea \leftarrow biderkatzailea + 1; --*aldagaiaren eguneraketa*

-- (*bukatzeko baldintza noizbait bete dadin*)

amaitu_errepikatu

amaiera



4. Simulazioa edo traza

Buelta kop* ¹	¿Sartu?	n	biderkatzailea	emaitza
hasieraketak	-	4	1	0
1	BAI	4	2	4
2	BAI	4	3	8
3	BAI	4	4	12
4	BAI	4	5	16
5	BAI	4	6	20
6	BAI	4	7	24
7	BAI	4	8	28
8	BAI	4	9	32
9	BAI	4	10	36
10	BAI	4	11	40
11	EZ			

Demagun “*” tekla izorratua dela. Biderketa bat gehiketen bitartez burutzea ere posible da hurrengo algoritmoak erakusten duenez.

1. Espezifikazioa:

Sarrera: Zenbaki oso bat

Aurre: $0 < n: \text{balioa} \leq 10$

Irteera: 10 zenbaki

Post: $0 < n\text{-ren biderkakizun bakoitza: balioa} \leq 100$

2. Proba kasuak

Sarrera: 4

Irteera: 4, 8, 12, 16, 20, 24, 28, 32, 36, 40

3. Algoritmoa:

kontadorea, gehiketa, n: integer; --*memoria erresebatzen*

hasiera

irakurri(n);

kontadorea $\leftarrow 0$;

gehiketa $\leftarrow 0$;

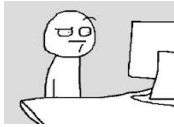
errepikatu kontadorea = 10 **bete arte** ---*bukatzeko baldintza*

gehiketa \square gehiketa + n;

idatzi(gehiketa);

kontadorea \square kontadorea + 1; ---*aldagaiaren eguneraketa*

¹ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula

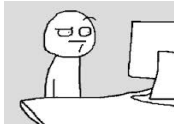


amaitu_errepikatu
amaiera

4. Simulazioa edo traza

Buelta kop* ²	¿Sartu?	n	kontadorea	gehiketa
hasieraketak	-	4	0	0
1	BAI	4	1	4
2	BAI	4	2	8
3	BAI	4	3	12
4	BAI	4	4	16
5	BAI	4	5	20
6	BAI	4	6	24
7	BAI	4	7	28
8	BAI	4	8	32
9	BAI	4	9	36
10	BAI	4	10	40
11	EZ			

² Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula



3. Ariketa

Demagun “*” izorratuta jarraitzen duela eta bi zenbakien arteko biderketa kalkulatzeko eskatzen digutela. Aldatu aurreko algoritmoa 2 zenbaki positibo emanda (biak ≤ 10) bien arteko biderketa kalkulatzeko.

1. Espezifikazioa:

Sarrera: 2 zenbaki oso

Aurre-baldintza: $0 < n$, eta biderkatzailea :balioa ≤ 10

Irteera: Zenbaki oso bat

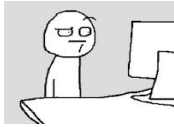
Post-baldintza: $0 < \text{emaitza:balioa} \leq 100 \mid \text{emaitza } n * \text{biderkatzailea}$
izango den

2. Proba kasuak

Sarrera	Irteera
n 5 eta biderkatzailea 1	5
n 5 eta biderkatzailea 4	20
n 10 eta biderkatzailea 10	100

3. Algoritmoa

1. Zenb1, zenb2, gehiketa: integer;
2. $\text{Gehiketa} \leftarrow 0$;
3. Idatzi(“sartu bi zenbaki”);
4. Irakurri(zenb1);
5. Irakurri (zenb2);
6. Errepikatu zenb2 = 0 bete arte
7. $\text{Gehiketa} \leftarrow \text{gehiketa} + \text{zenb1}$;
8. $\text{Zenb2} \leftarrow \text{zenb2} - 1$;
9. Amaitu_errepikapena;
10. Idatzi(gehiketa);



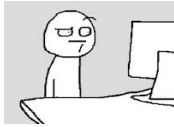
3. Simulazioa

Buelta kop* ³	¿Sartu?	Znb_1	Znb_2	Gehiketa
hasieraketak	-	5	4	0
1	bai	5	3	5
2	bai	5	2	10
3	bai	5	1	15
4	bai	5	0	20
5	ez			
Buelta kop* ⁴	¿Sartu?	Znb_1	Znb_2	Gehiketa
hasieraketak	-	5	1	0
1	bai	5	0	5
2	EZ			
Buelta kop* ⁵	¿Sartu?	Znb_1	Znb_2	Gehiketa
hasieraketak	-	10	10	0
1	bai	10	9	10
2	bai	10	8	20
3	bai	10	7	30
4	bai	10	6	40
5	BAI	10	5	50
6	BAI	10	4	60
7	BAI	10	3	70
8	BAI	10	2	80
9	BAI	10	1	90
10	BAI	10	0	100
11	EZ			

³ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula

⁴ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula

⁵ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula



4. Ariketa

Orain demagun “/” ere izorratu dela. Erabiltzaileari eskatu zenbaki oso bat, zatikizun deituko duguna ($0 \leq \text{zatikizun} \leq 100$) eta beste zenbaki oso bat zatitzailea deituko duguna ($0 < \text{zatitzailea} \leq 10$). Algoritmoak eskatutako balioen arteko zatiketaren emaitza osoa eta hondarra kalkulatu behar du. Horretarako ez dira “/” eta rem eragiketak erabiliko, zatiketa kenketak eginez burutu beharko da.

Zatiketa zatitzailea zatikizunari ken diezaiokegun aldien kopurua izango da, emaitza ez-negatiborik eman gabe. Adibidez, $6/2 \Rightarrow 6-2=4 \Rightarrow 4-2=2 \Rightarrow 2-2=0$, 3 kenketa burutu direnez, honek esan nahi du zatiketaren emaitza 3 izango dela, eta hondarra kasu honetan 0. Beste adibide honetan, $7/2 \Rightarrow 7-2=5 \Rightarrow 5-2=3 \Rightarrow 3-2=1$, 3 kenketa burutu direnez, honek esan nahi du zatiketaren emaitza 3 izango dela eta hondarra kasu honetan 1 izango da.

1. Espezifikazioa:

Sarrera: 2 zenbaki oso

Aurre: $0 \leq \text{zatikizun:balio1} \leq 100$

$0 < \text{zatitzaile:balio2} \leq 10$

Irteera: 2 zenbaki oso

Post: $0 \leq \text{zatiketa:zatikizun}$ eta zatitzaile arteko zatiketa osoaren emaitza

$0 \leq \text{hondarra: zatikizun}$ eta zatitzaile arteko zatiketa osoa egitearen hondarra $< \text{zatitzaile}$

2. Proba kasuak

Sarrera	Irteera
Zatikizun 6 zatitzaile 2	Zatiketa 3 eta hondarra 0
Zatikizun 21 zatitzaile 2	Zatiketa 10 eta hondarra 1

3. Algoritmoa:

zatikizun, zatitzaile, zatiketa, hondarra: integer;

hasiera

zatiketa $\leftarrow 0$;

hondarra $\leftarrow 0$;

Irakurri(zatikizun);

Irakurri(zatitzaile);

errepikatu zatikizuna $<$ zatitzailea **bete arte** ;

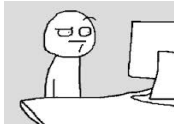
zatiketa \leftarrow zatiketa + 1;

hondarra \leftarrow zatikizun – zatitzaile;

zatikizun \leftarrow zatikizun – zatitzaile;

amaitu_errepikatu

idatzi(“zatiketaren emaitza” zatiketa);



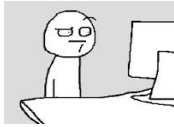
idatzi(“hondarra” hondarra “da”);
amaiera

4) Simulazioa edo traza

Buelta kop* ⁶	Sartzen da?	zatikizun	zatitzaile	zatiketa	hondarra
hasieraketak	-	21	2	0	0
1	BAI	19	2	1	19
2	BAI	17	2	2	17
3	BAI	15	2	3	15
4	BAI	13	2	4	13
5	BAI	11	2	5	11
6	BAI	9	2	6	9
7	BAI	7	2	7	7
8	BAI	5	2	8	5
9	BAI	3	2	9	3
10	BAI	1	2	10	1
11	EZ				
Buelta kop* ⁷	Sartzen da?	zatikizun	zatitzaile	zatiketa	hondarra
hasieraketak	-	6	2	0	0
1	BAI	4	2	1	4
2	BAI	2	2	2	2
3	BAI	0	2	3	0
4	EZ				

⁶ Buelta bakoitzean bueltaren bukaeran aldagaien balioak nola gelditzen diren errepresentatuko da

⁷ Buelta bakoitzean bueltaren bukaeran aldagaien balioak nola gelditzen diren errepresentatuko da



5. Ariketa

Izorratatuko guztia konpondu dugu jada!! Beraz orain eragiketa guztiak erabilgarri daude berriro. Erabiltzaileari eskatutako N zenbaki jakin bat edukita (≥ 1), n eta n^2 balioen artean dauden bere multiplo bikoiti guztiak kalkulatzen dituen programa sortu.

1. Espezifikazioa:

Sarrera: Zenbaki oso bat

Aurre: $1 \leq n$:balioa

Irteera: n eta n^2 balioen artean dauden n-ren multiplo bikoiti haina zenbaki

Post: $n \leq$ n-ren multiplo bikoitiak $\leq n^2$;

2. Proba kasuak

Sarrera: 2

Irteera: 2, 4

Sarrera: 7

Irteera: 14, 28, 42

3. Algoritmoa:

probatzeko_multiplo, n: integer; ---memoria alokatzen

hasiera

Irakurri(n);

probatzeko_multiplo \leftarrow n;

Errepikatu probatzeko_multiplo $\geq n^2$ **bete arte** ;

 Baldin (probatzeko_multiplo rem 2 = 0) orduan

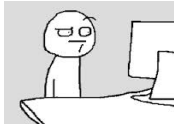
 Idatzi (probatzeko_multiplo)

 Amaitu_baldin;

 Probatzeko_multiplo \leftarrow Probatzeko_multiplo + n;

amaitu_errepikatu

amaiera



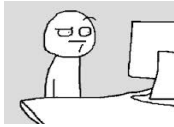
4. Simulazioa edo traza

Buelta kop* ⁸	¿Sartu?	n	probatzeko_multiplo	Pantaila
hasieraketak	-	7	7	-
1	BAI	7	14	14
2	BAI	7	21	14
3	BAI	7	28	14 28
4	BAI	7	35	14 28
5	BAI	7	42	14 28 42
6	BAI	7	49	14 28 42
7	EZ			

Buelta kop* ⁹	¿Sartu?	n	probatzeko_multiplo	Pantaila
hasieraketak	-	2	2	-
1	BAI	2	2	2
2	BAI	2	4	2 4
3	EZ			

⁸ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula

⁹ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula



6. Ariketa

Sei baino txikiagoa ala berdina den zenbaki oso bat emanda, kalkulatu bere faktoriala

1. Espezifikazioa:

Sarrera: Zenbaki oso bat

Aurre: $0 \leq \text{zenb:balioa} \leq 6$

Irteera: Zenbaki oso bat

Post: $0 < \text{faktoriala:balioa} \leq 720$ | $\text{emaitza} = \text{num} * (\text{num}-1)!$ eta
emaitza1 izango da num 0 denean.

2. Proba kasuak

Sarrera	Irteera
0!	1
4!	24

3. Algoritmoa:

faktorial, zenb, Zenb_aux: integer;

hasiera

faktorial \leftarrow 1;

zenb_aux \leftarrow 1;

Irakurri(zenb);

errepikatu faktorial > zenb **bete arte** ;

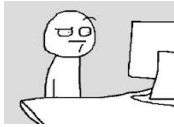
faktorial \leftarrow faktorial +1;

zenb_aux \leftarrow zenb_aux * faktorial;

amaitu_errepikatu;

idatzi(zenb_aux);

amaiera



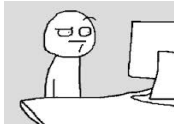
4. Simulazioa edo traza

Buelta kop* ¹⁰	¿Sartu?	zenb	faktorial	Zenb_aux	pantaila
hasieraketak	-	0	1	1	1
1	EZ				

Buelta kop* ¹¹	¿Sartu?	zenb	faktorial	Zenb_aux	pantaila
hasieraketak	-	4	1	1	1
1	BAI	4	2	2	2
2	BAI	4	3	6	6
3	BAI	4	4	24	24
4	EZ				

¹⁰ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula

¹¹ Buelta bakoitzean sartzean aldagaien balioak errepresentatzen duen taula

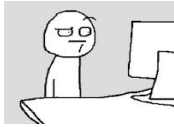


SEKUENTZIEKIN LAN EGITEN

Zenbaki sekuentziekin eragiketa batzuk

- 1) Memoria erreserbatu zenbaki sekuentzia batentzat
erreserbatu 10 zenbaki osorentzat
sekuentzia¹²: 10 integer;
- 2) Teklatutik sekuentzia osatuko duten zenbakiak takada batean irakurri eta sekuentzia aldagaian gorde
sekuentzia_irakurri(sekuentzia)
- 3) Gorde sekuentzian n balioa, egungo posizioan (hau da, aurrera_egin/atzera_egin-ez gelditu garen posizioan)
gorde(sekuentzia, n);
- 4) Sekuentziaren hasierara joan
hasieran_jarri(sekuentzia);
- 5) Sekuentziaren bukaerara joan
bukaeran_jarri(sekuentzia);
- 6) Aurrera egin sekuentziaren posizioetan
aurrera_egin(sekuentzia);
horrela sekuentzian zehar mugitu ahalko gara aurretik atzera
- 7) Atzera egin sekuentziaren posizioetan
atzera_egin(sekuentzia);
horrela sekuentzian zehar mugitu ahal gara atzetik aurrera
- 8) Lortu sekuentziaren elementu bat, hain zuzen ere egungo posizioan dagoena
egungo_elementua(sekuentzia);
- 9) Sekuentziatik kanpo gauden jakiteko
sekuentziatik_kanpo(sekuentzia); true sekuentziatik kanpo baldin bagaude ala false barne bagaude

¹²Kasu honetan aldagaiaren izen bezala **sekuentzia** hautatu dugu, baina **sek** edo **segida** edo hautatu dezakegun edozein izen esanguratsua izan zitekeen



7. Ariketa

Eragiketa hauekin sekuentzia baten zeharketa egingo dugu. Sekuentziaren elementu bakoitza pantailaratuz

Suposatu dezagun **sekuentzia_irakurri(sekuentzia)** algoritmo bat existitzen dela, eta erabili dezakegula. Algoritmo hau egikaritzuz, erabiltzaileari sekuentzia bat guztiz betetzeko aukera ematen diogu.

1. Espezifikazioa:

Sarrera: 10 zenbaki oso

Aurre: -

Irteera: 10 zenbaki oso

Post: pantailaratuko dira sekuentzian gordetako balioak

2. Proba kasuak

Sarrera	Irteera
12000,565,7,8,2,300,4,6,9,0	12000,565,7,8,2,300,4,6,9,0

3. Algoritmoa

sekuentzia1: 10 integer;---*alokatu memoria*

aux: integer;

hasiera

irakurri_sekuentzia(sekuentzia1);

hasieran_jarri(sekuentzia1);

errepikatu (sekuentziazatik_kanpo(sekuentzia1)) **bete arte**;

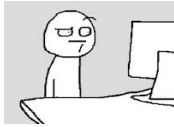
 aux ← **egungo_elementua**(sekuentzia1);

 idatzi(aux);

aurrera_egin(sekuentzia1);

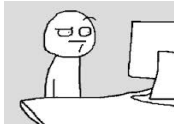
amaitu_errepikatu

amaiera



4. Simulazioa edo traza

Buelta kop	Sartu?		aux
Hasieraketak	-		-
1	BAI		1200
2	BAI		565
3	BAI		7
4	BAI		8
5	BAI		2
6	BAI		300
7	BAI		4
8	BAI		6
9	BAI		9
10	BAI		0



8. Ariketa

Orain, sekuentzia bat nola zeharkatzen den jakinda, idatzi algoritmo bat sekuentzia baten zenbaki guztien gehiketa kalkulatuko duena.

1. Espezifikazioa

Sarrera. 10 zenbaki oso

Aurre:-

Irteera: zenbaki oso bat

Post: pantailaratuko da sekuentziaren gehiketa.

2. Proba kasuak

Sarrera	Irteera
12000,565,7,8,2,300,4,6,9,0	12000,565,7,8,2,300,4,6,9,0

3. Algoritmoa

sekuentzia1: 10 integer;

gehiketa: integer;

hasiera

irakurri_sekuentzia(sekuentzia1);

hasieran_jarri(sekuentzia1);

gehiketa \leftarrow 0;

errepikatu (sekuentziatik_kanpo(sekuentzia1)) **bete arte**;

gehiketa \leftarrow gehiketa + **egungo_elementua**(sekuentzia1);

idatzi(gehiketa);

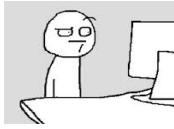
aurrera_egin(sekuentzia1);

amaitu_errepikatu

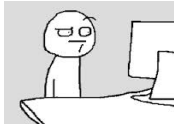
amaiera

4. Simulazioa edo traza

Buelta kop	Sartu?		aux
Hasieraketak	-		-
1	BAI		1200
2	BAI		1765
3	BAI		1772
4	BAI		1780
5	BAI		1782
6	BAI		2082
7	BAI		2086
8	BAI		2092



9	BAI		2101
10	BAI		2101
11	EZ		



9. Ariketa

Idatzi algoritmo bat sekuentzia batean 0z bukatzen diren zenbakiak pantailaratzen dituen.

1. Espezifikazioa

Sarrera. 10 zenbaki oso.

Aurre:-

Irteera:

Post: 0z bukatzen diren zenbaki guztiak pantailaratuko dira.

2. Proba kasuak

Sarrera	Irteera
12000,565,7,8,2,300,4,6,9,0	12000,565,7,8,2,300,4,6,9,0

3. Algoritmoa

sekuentzia1: 10 integer;

aux: integer;

hasiera

irakurri_sekuentzia(sekuentzia1);

hasieran_jarri(sekuentzia1);

errepikatu (sekuentziatik_kanpo(sekuentzia1)) **bete arte**;

 baldin (**egungo_elementua**(sekuentzia1) rem 10 = 0) orduan

 aux ← **egungo_elementua**(sekuentzia1);

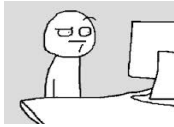
 idatzi(aux);

 amaitu_baldin;

aurrera_egin(sekuentzia1);

amaitu_errepikatu

amaiera



4. Simulazioa edo traza

Buelta kop	Sartu?	aux	pantaila
Hasieraketak	-	-	-
1	BAI	1200	1200
2	BAI	-	1200
3	BAI	-	1200
4	BAI	-	1200
5	BAI	-	1200
6	BAI	300	1200 300
7	BAI	-	1200 300
8	BAI	-	1200 300
9	BAI	-	1200 300
10	BAI	0	1200 300 0
11	EZ		