OP-4 Laborategiko informea

Izena: Eneko Samepdro, Gontzal Pujana eta Markel Arevalo **Data:** 2014-10-22

1-Ikasitako kontzeptuak eta lortutako gaitasunak

	Eneko	BAI	EZ
1	Baldintzak sortzen badakit.	Bai	
2	Sekuentzia bat zeharkatzeko iterazioa sortzen badakit.	Bai	
3	Sekuentziak tratatzeko eragiketak badakizkit eta ondo ulertzen ditut.	Bai	

4	Irakurri_sekuetzia(seku1);
	kokatu_hasieran(seku1);
	errepikatu atera sekuentziatik_kanpo(seku1);
	idatzi(egungo_elementua(seku1);
	amaitu_errepikatu;
	Badakit algoritmo hau simulatzen eta ulertzen dut. Zeozer kopondu behar al da? Konpondu ezazu
		aurrera_jo() funtzioa erabili behar da, hurrengo baliora joateko, horrela sekuentziatik kanpo egoteko, eta programa amaitzeko.	

5	BESTELAKOAK: Azaldu zeintzuk		

	Gontzal	BAI	EZ
1	Baldintzak sortzen badakit.	Bai	
2	Sekuentzia bat zeharkatzeko iterazioa sortzen badakit.	Bai	
3	Sekuentziak tratatzeko eragiketak badakizkit eta ondo ulertzen ditut.	Bai	

4	Irakurri_sekuetzia(seku1);
	kokatu_hasieran(seku1);
	errepikatu atera sekuentziatik_kanpo(seku1);
	idatzi(egungo_elementua(seku1);
	amaitu_errepikatu;
		Kode hori infinituki errepikatuko da, beraz hurrengo baliora eramaten gaituen	

	Badakit algoritmo hau simulatzen eta ulertzen dut. Zeozer kopondu behar al da? Konpondu ezazu	zerbait behar dugu.	
5	BESTELAKOAK: Azaldu zeintzuk		

	Markel	BAI	EZ
1	Baldintzak sortzen badakit.	Bai	
2	Sekuentzia bat zeharkatzeko iterazioa sortzen badakit.	Bai	
3	Sekuentziak tratatzeko eragiketak badakizkit eta ondo ulertzen ditut.	Bai	

4	Irakurri_sekuetzia(seku1);
	kokatu_hasieran(seku1);
	errepikatu atera sekuentziatik_kanpo(seku1);
	idatzi(egungo_elementua(seku1);
	amaitu_errepikatu;
	Badakit algoritmo hau simulatzen eta ulertzen dut. Zeozer kopondu behar al da? Konpondu ezazu
		Txarto dago. aurrera_jo() edo atzera_jo() funtzioak erabili behar dira errepikatu atera horretatik ateratzeko.	

5	BESTELAKOAK: Azaldu zeintzuk		

2- Laborategiko Soluzioa

1- Ariketa

10 zenbaki osoz osatutako sekuentzia bat edukita, sekuentzian dagoen lehenengo zenbaki bikoitia inprimatzeko programa idatz ezazue. Ez balego zenbaki bikoitirik mezu bat agertuko da horretaz informatzeko.

1. Espezifikazioa

Sarrera: 10 zenbaki osoko sekuentzia

Aurre: sekuentzia 10 zenbakiz osatuta dago, ez bat gutxiago, ez bat gehiago.

Irteera: zenbaki osoko bat ala mezu bat

Post: balioa hondarra 2=0 | balioa∈ hasierako sekuentzia ala “zenbaki bikoitirik ez dago” mezua.

2. Proba Kasuak

1,6,8,4,2,3,9,2,5,1 -> “6”

1,3,5,3,7,9,11,13,21,33 -> “Zenbaki bikoitirik ez dago”

3,5,11,69,5,11,1,7,65,10 -> “10”

3.Algoritmoa

```
sekuentzia: 10 integer;
BikoitiaDa: boolean;
BikoitiaDa <- false;
irakurri_sekuentzia(sekuentzia);
kokatu_hasieran(sekuentzia);

Errepikatu atera(sekuentziatik_kanpo(sekuentzia) = true) edo (BikoitiaDa = true)
        (egungo_elementua(sekuentzia) rem 2 = 0) egia balitz
                BikoitiaDa = true;
        amaitu;
        aurrera_jo(sekuentzia);

errepikatu_amaitu;
(BikoitiaDa = true) egia balitz
        Idatzi(“%egungo_elementua(sekuentzia) sekuentziako lehen zenbaki bikoitia da”);
bestela
        Idatzi(“Zenbaki bikoitirik ez dago”);
amaitu;
```

4.Simulazioa

1,3,5,3,7,9,11,13,21,33

egungo_elementua(sekuentzia)	BikoitiaDa
1	false
3	false
5	false
3	false
7	false
9	false
11	false
13	false
21	false
33	false

1,3,5,4,7,9,12,13,21,102

egungo_elementua(sekuentzia)	BikoitiaDa
1	false
3	false
5	false
4	true

2- Ariketa

10 zenbaki osoz osatutako sekuentzia bat edukita, sekuentziako posizio bikoitietan dauden elementuak goranzko ordena jarraitzen duten ala ez esango digun algoritmoa idatz ezazue.

1. Espezifikazioa

Sarrera: 10 zenbaki osoko sekuentzia

Aurre: sekuentzia 10 zenbakiz osatuta dago, ez bat gutxiago, ez bat gehiago.

Irteera: Mezu bat

Post: “posizio bikoitietan dauden elementuak goranzko ordena jarraitzen dute” |balio1 eta balio2 posizioen hondarra 2=0 eta balio1 <| balio2 eta balioa1 eta balio2 ∈ hasierako sekuentzia

2. Proba kasuak

1 ,2 ,5, 7, 8, 9, 10 ,12, 25, 67 -> Bai, goranzko ordena jarraitzen du

1 ,2 ,5, 7, 8, 9, 10 ,12, 25, 3 -> Ez, ez du goranzko ordena jarraitzen

3.Algoritmoa

```
sekuentzia: 10 integer;
zen1, zen2: integer;

ordenan: boolean;
ordenan <- true;

irakurri_sekuentzia(sekuentzia);
kokatu_hasieran(sekuentzia);
aurrera_jo(sekuentzia);

Errepikatu atera (sekuentziatik_kanpo(sekuentzia) = true) edo (ordenan = false)
        zen1 = egungo_elementua(sekuentzia);
        aurrera_jo(sekuentzia);
        aurrera_jo(sekuentzia);
        zen2 = egungo_elementua(sekuentzia);
        (zen1 > zen2) egia balitz
                ordenan <- false;
        amaitu;
errepikatu_amaitu;
```

(ordenan = true) egia balitz
    Idatzi(“Bai, goranzko ordena jarraitzen du”);
bestela
    Idatzi(“Ez, ez du goranzko ordena jarraitzen”);
amaitu;

4.Algoritmoa

2,3,6,9,4,5,8,11,14,25

egungo_elementua(sekuentzia)	**ordenan**
3	true
9	true
5	false

11,2,15,3,62,5,98,9,10,11

egungo_elementua(sekuentzia)	**ordenan**
2	true
3	true
5	true
9	true
10	true

3- Ariketa

10 zenbaki osoz osatutako sekuentzia bat edukita, sekuentzian dauden elementuen artean, sekuentziako azken elementuagatik zatigarria direnak inprimatzeko algoritmoa idatz ezazu.

1. Espezifikazioa

Sarrera: 10 zenbaki osoko sekuentzia

Aurre: sekuentzia 10 zenbakiz osatuta dago, ez bat gutxiago, ez bat gehiago.

Irteera: Hainbat zenbaki osoko ala mezu bat.

Post: balioa|balioa $\in$ hasierako sekuentzia eta balioaren hondarra azkeneko balioa=0.

2. Proba kasuak

1 ,2 ,5, 7, 8, 9, 10 ,12, 25, 1 -> 1 ,2 ,5, 7, 8, 9, 10 ,12, 25

1 ,2 ,5, 7, 8, 9, 10 ,12, 25, 30 -> Ez dago azkeneko zenbakiagatik zatigarria den zenbakirik.

3. Algoritmoa

```
sekuentzia: 10 integer;
zen1, zen2, kont: integer;
zatigarriDa: boolean;
zatigarriDa <- false;

kokatu_bukaeran(sekuentzia);
zen1 <- egungo_elementua(sekuentzia);
kokatu_hasieran(sekuentzia);

Errepikatu atera (sekuentziatik_kanpo(sekuentzia) = true) edo (kont = 9)
        zen2 <- egungo_elementua(sekuentzia);
        (zen2 rem zen1 = 0) egia balitz
                Idatzi(%zen2);
                zatigarriDa <- true;
        amaitu;
        aurrea_jo(sekuentzia);
        kont <- kont +1;
amaitu_errepikatu;
(zatigarriDa = false) egia balitz
        Idatzi("Ez dago azkeneko zenbakiagatik zatigarria den zenbakirik.");
amaitu;
```

4 Proba kasuak

Sarrera: **2,3,6,44,10,25,26,97,3,1**

zen1	**zen2**	**kont**	**zatigarriDa**
1	2	1	true
1	3	2	true
1	6	3	true
1	44	4	true
1	10	5	true
1	25	6	true
1	26	7	true
1	97	8	true
1	3	9	true

Irteera: **2,3,6,44,10,25,26,97,3,1**

Sarrera: **2,3,6,44,10,25,26,97,3,3**

zen1	**zen2**	**kont**	**zatigarriDa**
3	2	1	false
3	3	2	true
3	6	3	true
3	44	4	true
3	10	5	true
3	25	6	true
3	26	7	true
3	97	8	true
3	3	9	true

Irteera: **3,6,26,3**

4-Ariketa

Eskatu erabiltzaileari zenbaki osoko bat (balio >0) eta irudika ezazu hurrengo grafikoa algoritmo bidez:

1.Proba Kasuak

N = 3	N = 5
*00	*0000
**0	**000
***	***00
	****0
	*****

2.Algoritmoa

```
zen1, aux, kont, kon1: integer;
Irakurri(zen1);
aux <- 0;
Errepikatu atera (aux > zen1)
        aux <- aux + 1;
        kont1 <- 0;
        kont2 <- 0;

        Errepikatu atera (kont1 = aux)
                Idatzi("*");
                kont1 <- kont1 + 1;
        errepikatu_amaitu;
        Errepikatu atera (kont2 >= (zen1 - kont1) )
                Idatzi("0");
                kont2 <- kont2 + 1;
        errepikatu_amaitu;
        lerro_saltoa();
errepikatu_amaitu;
```

3.Simulazioa

zen1 sarrera: **2**

zen1	**aux**	**kont1**	**kont2**
2	0	0	kk
2	1	0	0
2	1	1	0
2	1	1	1
2	2	0	0
2	2	1	0
2	2	2	0

Irteera:

***0**

**

6-Ariketa

Erabiltzaileari >0 den zenbaki oso bat eskatuko dion programa bat idatziko dugu. Programa horren helburua, erabiltzaileak sartutako zenbakiak zenbat zifra dituen asmatzea izando da.

1. Espezifikazioa

Sarrera : *zenbaki bat*

Aurre: *zenbaki oso bat >0*

Irteera: *zenbaki oso bat*

Post: *pantailatik >=1 den balio bat inprimatuko da. Balio horrek sarrerako zenbakiaren zifra kopurua izando da.*

2. Proba kasuak

36 -> 2

1025 -> 4

295456 -> 6

3. Algoritmoa

```
zen1, kont: integer;
kont <- 1;
Irakurri(zen1);
aux: boolean;
aux <- false;

Errepikatu atera ( aux = true)
        zen1 = zen1 / 10;
        (zen1 = 0) egia balitz
                aux = true;
        amaitu;
        kont <- kont +1;
amaitu_errepikatu
Idatzi(kont);
```

4. Simulazioa

Sarrera: **3**

zen1	**kont**	**aux**
3	1	false
0	1	true

Irteera: **1**

Sarrera: **1025**

zen1	**kont**	**aux**
1025	1	false
102	2	false
10	3	false
1	4	false
0	4	true

Irteera: **4**

7-Ariketa

Eskatu erabiltzaileari zenbaki binartar bat sartzeko (zenbaki oso bat balitz gordeko dena), 1etik hasiko dena eta zenbakia hamartarrera bihurtu.

1. Espezifikazioa

Sarrera : zenbaki bat
Aurre: 1z hasten den zenbaki bitar bat (hau da 0z eta 1z soilik osatua), gehienez 10 zifrakoa

Irteera: zenbaki oso bat
Post: Sarrerako zenbaki bitarraren balio hamartarra. Adibidea

$10010= 1*2^4+0*2^3+0*2^2+1*2^1+0*2^0= 18$
ADAz $oinarria^{exp}$ -> oinarria**exp
Cz $oinarria^{exp}$ -> pow(oianrria,exp). Gehitu goikaldean #include <math.h> liburuategia
Perl-ez $oinarria^{exp}$ -> oinarria**exp
Python-ez $oinarria^{exp}$ -> oinarria**exp

2. Proba kasuak

$111 = 1*2^2 + 1*2^1 + 1*2^0 = 7$
$1001010 = 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 = 74$

3. Algoritmoa

```
sekuentzia: 10 integer;
irakurri_sekuentzia();
kokatu_bukaeran(sekuentzia);
akumuladorea, kont: integer;
kont <- 0;

Errepikatu atera (sekuentziatik_kanpo(sekuentzia) = true)

        akumuladorea <- (egungo_elementua(sekuentzia)) * (2**kont);
        kont <- kont +1;
        atzera_jo(sekuentzia);

errepikatu_amaitu;
```

Idatzi(akumuladorea);

4. Simulazioa

Sarrera: **1001010**

akumuladorea	kont	egungo_elementua(sekuentzia
0	0	0
2	1	1
0	2	0
10	3	1
0	4	0
0	5	0
74	6	1

Irteera: **74**