

Programazioa: Algoritmoak eta M Lengoaia

iker¹ and jesus.ibanez¹

¹University of the Basque Country

August 21, 2019

Egileak:

Iker Azpeitia

Jesús Ibáñez

Baliabideak:

-Baliabideak:

<http://euskaljakintza.com/baliabideak/>

<http://www.ehu.eus/seg/>

-Euskara => X hiztegiak:

<https://eu.glosbe.com/>

-Euskarazko ikasmaterialgintza sustatzeko 2017ko deialdia

<https://www.ehu.eus/eu/web/euskara/2017-10-09-deialdia>

Eskuliburua:

Atalen Aurkibidea:

1. Sarrera (Eskuliburu hau nola erabili)
2. Informatika (azalpen orokorrak: HW, SW, informazio-lanketa= datu sarrera+prozesaketa+datu irteera)
3. M lengoiarako Garapen-Ingurune Bateratuak: Matlab eta Octave.
4. Oinarrizko prozesaketa M lengoian: (Aldagaiak, datu-bakarrak, eragile sinpleak)
5. Prozesaketa aurreratua M lengoian: (Datu egiturak, taulak, karaktere taulak, barne funtzioak).
6. Datu sarrera-irteera M lengoian: (testu-datu trukaketa, grafikoak, fitxategiak)
7. M lengoiako paketeak eta izengabeko funtzioak.
8. Algoritmoak. (Zehaztapena, Fluxu Diagramak eta Simulazio Taulak).
9. Kontrol egiturak.
10. Azpiprogramak M lengoian. (agindu-multzoak, programatzaile-funtzioak)
11. Algoritmoak M lengoiara itzuli.

12. Adibideak eta ariketak: (Zenbakizko Kalkulua, Datu ugarien kudeaketa, Simulazioak, Jolasak,)

1. SARRERA

.....

ARIKETAK eta ADIBIDEAK tartekatzen dira .

.....

2. INFORMATIKA

Harluxet (<http://www1.euskadi.net/harluxet/>) hiztegi entziklopedikoaren arabera, *Informatika* (<http://www1.euskadi.net/harluxet/hiztegia.asp?sarrera=informatika>) da “Informazioaren tratamendu automatikoaren zientzia. [...] Zientzia honen funtsa datuak bildu, sarrera-unitateen bidez ordenadorera sartu, bertako unitate zentrallean prozesatu ondoren informazio erabilgarria sortu eta irteera-unitateen bidez ateratzeari datza. Prozesu honetan bi kontzeptu nagusi bereiz daitezke: hardwarea, hots, prozesamendu automatikoan erabiltzen diren tresnen multzoa, eta softwarea ordenadoreak funtzionatzeko eta problema konkreituak ebazteko behar diren programen multzoa.[...]”.

Berez, *Informatika* hitza “informazioa” eta “automatikoa” hitzak elkartetik eratorri da. Ordenadoreak (ordenagailu edo konputagailu ere izendatzen da) datuen prozesaketa automatikoa bideratzen du gailua fisiko (hardware) eta programen (software) bitartez. Baina, zer da Informazioa? “0” si datua da, errepresentazio bat den heinean. Gizakiok datuak interpretatu eta ondorioak ateratzen ditugunean Informazioa bihurtu dugu datua. Alegia, Informazioa gizakiak datuen interpretazioa izango litzateke. Adibidez, “0” datua ikasle baten nota baldin bada informazio garrantzitsua bilakatzen da ikaslearentzat.

Ordenagailuek datuak prozesatzen dituzte hauek interpretatu gabe. Gizakiak dira datuak (sarrerako zein irteerakoak) interpretatzen dituztenak. Gizaki eta ordenagailu binomio horri *Sistema Informatikoa* deritzaio.

Datuen prozesaketa hiru urrats hauetan banatzen da:

| *Sarrera* => *Prozesaketa* => *Irteera*

Elkarlotutako hiru maila desberdinetan ikus ditzakegu urrats hauek. Maila logikoan, gizakiok problemaren hasierako eta emaitzen informazioa erlazionatuz problemak zehazten ditugu. Software mailan, programa baten datuak eskatu, aginduak jarraituz tarteko emaitzak kalkulatu eta azken emaitza bueltatzen du. Hardware mailan, gailu fisikoek datuak ordenagailura sartzen dituzte, irteera gailuek emaitzak erabiltzaileari aurkeztu, eta Prozesaketarako Unitate Zentralak (PUZ) kalkuluak egiten ditu.

Ordenagailuek prozesaketa automatikoa era egokian egiten dute ezaugarri hauek betetzen dituztelako:

- Fidagarriak: errore gutxi dituzte. Hardwarea matxuratzen denean aldatu egiten dute, programek erroreak baldin badituzte programatzailearen akatsa izaten da.
- Azkarrak: kalkulu ugari denbora gutxian. Ordenagailuen kalkulo-potentzia FLOP (Floating Point Operations per Second) neurrian adierazten da. Hau da, segundoko zenbat eragiketa egikari ditzakeen. Gaur egungo ordenagailuen potentzia TeraFLOPetan adierazten da, 10 ber 9 eragiketa segundoko alegia.

Gizakiak Problema Zehazten du	Sarrera	Prosezaketa	Irteera
Programak (SW) datuak moldatzen ditu aginduak jarraituz	Problemaren informazioa	Hasierako informazio eta emaitzen arteko erlazioak	Emaitza
Gailuek (HW) azkar eta zehatz egikaritzen dute programa	Datuak hartu	Kalkuluak egin programaren aginduak jarraituz	Datuak bueltatu
	Sarrerarako gailuak	PUZ: prozesakeraturako Unitate Zentrala	Irteerarako gailuak

Table 1: Datu prozesaketarako urratsak

- Zehatzak: zenbakiak nahi adina zehaztasunarekin adierazi ditzakegu. Adibidez, M lengoaian erabili dezakegun zenbaki positibo handiena $1.7977e+308$ da, eta txikiena $2.2251e-308$.
- Errentagarriak: teknologia urtetik urtera potentzia handitzen doa eta prezio mantendu ala gutxitzen.
- Moldagarria: alde batetik, ordenagailuek gailu berriak txertatzea ahalbidetzen dute, bai potentzia zein egin beharreko berriak onartzeko. Bestetik, ordenagailuak helburu orokorrekoak dira, beraz egin beharreko zehatz batera bideratzeko programa zehatza egikaritu behar dute. Honen abantaila da ordenagailu bakarrarekin egin beharreko asko gauzatu ditzakegula, programen bitartez.

Beraz, programatzaileak problema ebatzi behar du. Ezin dugu programatu alde aurretik ebazten ez daki-guna. Problema bat nola ebazten den jakinda, ordenagailuari jarraitu beharreko pausuak adierazi beharko dizkiogu. Alegia, sarrerako datuak nola prozesatu behar diren emaitzak lortzeko. Programatzeak denbora eskatzen du, horregatik kontutan hartu beharko dugu ea komeni den programa bat sortzea. Problema horrelako ezaugarriak dituztenean programa bat sortzea komeni dezake:

- Datu asko ditugunean edo ohiko ataza errepikakorrek egin behar ditugunean. Gizakiok aspertu egiten gera, ordenagailuak inoiz ez.
- Kalkulu konplexuak direnean. Ordenagailuen abiaduraz eta zehaztasunez egingo dituzte, errorerik gabe.
- Datuak konpartitu behar direnean bikoizketak ekiditeko.

ARIKETAK:

Ondorengo ariketen emaitza adierazi:

$2+2 =$

hamar aldiz $2+\dots+2 =$

ehun aldiz $2+\dots+2 =$

$2*2 =$

hamar aldiz $2*\dots*2 =$

ehun aldiz $2*\dots*2 =$

$2 \text{ ber } 2 =$

hamar aldiz $2 \text{ ber } 2 \text{ ber } 2 \dots \text{ ber } 2 =$

ehun aldiz $2 \text{ ber } 2 \text{ ber } 2 \dots \text{ ber } 2 =$

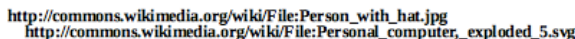
Programak *programazio lengoaien* bitartez idazten dira, eta programen diseinua *algoritmoen* bitartez. Harluxet hiztegi-entziklopedikoaren arabera, algoritmoa da:

Problema baten ebazpenerako eman behar diren urratsen deskribapen formalak.
Programazio-lengoaia baten bidez, algoritmoa ordenadore batek egikari dezakeen programa bihurtzeko daiteke

Euskalterm terminologia banku publikoaren arabera, programazio-lengoaia da:

Lengoaia artifiziala, bereziki eta espresuki ordenagailu-programak adierazteko sortua.

Esan bezala, ordenagailuak oso egokiak dira problema konplexuak zuzen eta azkar ebatzteko. Horretarako,



A diagram of a computer system with numbered components. 1. A small grey box (modem or power supply) with a cable to the tower. 2. A tower unit with a floppy disk drive. 3. A CD-ROM drive. 4. A monitor. 5. A keyboard. 6. A mouse. 7. A tower unit with a floppy disk drive. 8. A tower unit with a floppy disk drive.

Ebazpen fasean problema zehazki adierazten da sarrera eta emaitzen erlazioak adieraziz modu matematikologikoan. Algoritmoak Zehaztapenean adierazitako erlazioak lortzeko bidean egin beharreko prozesaketa adieraziko du. Algoritmoa eskuz eginiko programaren diseinua da. Ondo dagoen ikusteko simulazio taulak

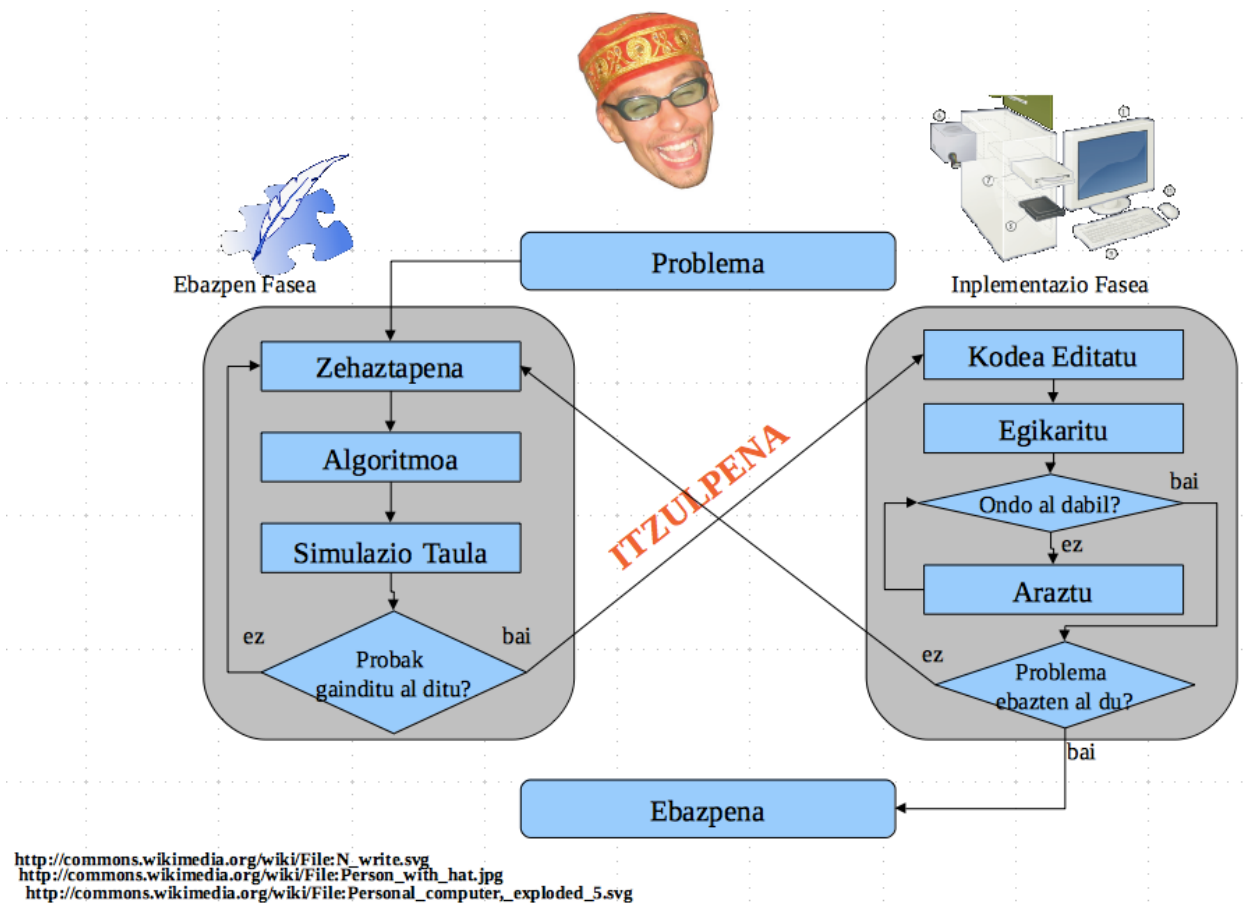


Figure 2: Ebazpen eta implementazio faseak.

beteko dira, non kasu bereziak probatuko diren. Probak ondo baldin badaude orduan algoritmoa programazio lengoaiara itzuliko da, era honetan programa exekutagarria sortuz. Behin programa idatzita daukagula, berriro ere probak egingo dira, baina kasu honetan, arazketa ordenagailuak ematen dituen tresna automatikoen bitartez egingo da.

2.1-IDEI NAGUSIAK

- Informazioa gizakiok egiten dugun datuen interpretazioa da. Batez ere emaitzen interpretazioa baina sarrerako datuei lotuta.
- Datuen prozesaketa automatikoari *Informatika* deritzo.
- Informatikak (ordenagailu+programak) datuen prozesaketa azkarra, zehatza eta merkea ahalbidetzen ditu.
- Datuen prozesaketaren pausuak hauek dira: datu sarrera, datuak moldatu eragiketen bitartez, emaitzen irteera

3-PROBLEMAK EZAGUTU ETA ULERTU

Informatikan, beste arlo askotan bezala, garrantzitsuena problema nola ebazten den jakitea da. Hau da, problemaren sarrerako datuak nola prozesatu behar diren emaitza lortzeko. Adibidez, bi zenbakien batez bestea nola ebazten da? ala, labirinto baten irteera nola aurketzen da? ala lur azpiko tenperaturen laginak izanda, zein da geotermia instalazio eraginkorra?

Problema motak askotarikoak dira, matematikoak, logikoak, jolas grafikoak eta abar. Hauek ebazten dituen programa bat ezin izango dugu idatzi problema nola ebazten den ez badakigu. Lehenengo pausua problema definitzen dituen sarrera eta irteerako datuak identifikatzea da.

3.1-ZEHAZTAPENA

Problema bat ondo definitzeko beharrezkoak diren datuak identifikatu behar ditugu, eta beste guztiak alde batera laga. Hau lortzeko aldera ebatzi nahi ditugun problemen *Zehaztapena* egiten dugu. Zehaztapena da problema baten sarrera datuak eta emaitzak (irteerako datuak) definitzea. Nolabait, problemaren **Aurrebaldintza** (sarrerako datuek bete beharrekoa) eta **Ondorengo-baldintza** (emaitzek bete beharrekoa) zehazten dugu. Datuak definituta izango ditugu lau galdera hauei erantzuna ematen badiegu:

- Zeintzuk dira datu horiek? Datuak izendatu momentu bakoitzean zein datuaz hitz egiten ari garen jakiteko. Adibidez, **kontagailua** izendatu dezakegu datu bat **topea** beste bat.
- Zein motakoak dira? Datuak mota desberdinekoak izan daitezke: zenbakiak (errealak ala osoak), letrak, taulak eta abar. Adibidez, **zenbaki osoa** izatea.
- Zer murriztapen betetzen dituzte? Hau da, datu mota batekoa izanda, edozein balio hartu dezake ala mugatuta al ditu balizko balioak? Baldintzen bitartez zein balio diren balizkoak adieraziko da. Adibidez, balio oso positiboak onartzen direla adierazteko **kontagailua** > 0 ezarri dezakegu.
- Zer erlazio dute datuak beraien artean? Datu mota eta balizko balioak zehaztu ondoren gerta daiteke datuek beraien arteko murriztapenak ezartzea. Adibidez, datu bat bestea baino txikiagoa izatea era honetan adierazi dezakegu, **kontagailua** $< \text{topea}$.

Aurrekoak adierazteko lengoia logiko-matematikoa erabiliko dugu; unibertsala, ulergarria eta unibokoa den heinean. Hala ere, ezinbestean, gure hitzez adieraziko ditugu erlazio eta murriztapenak. Hortaz, datu motak multzo matematikoen ikurren bitartez adieraziko ditugu. Laburbilduz hauek dira:

- Zenbaki oso: $[?]$ **Z**
- Zenbaki erreal: $[?]$ **R**
- Karakterea: $[?]$ **karakterea**
- Balio logikoa (bai/egi ez/gezur): $[?]$ **logikoa**
- Multzo esplizitu bat: $[?]$ **{15, 30, 45, 60}** Adibide honek adierazten du datuak 4 balio hauetako bat hartu dezakeela.

Datu mota sinple hauek tauletan elkartu ahala dira. Dimentsio anitzetako taulak ian daitezke: bat (bektore), bi (matrize), hiru (kuboa) eta abar. Horiek datuaren izenaren ondoren dimentsio eta dimentsio bakoitzaren tamaina zehaztuko ditugu. Adibidez, **Tenperaturak(3,4) [?] R** adieraziko du Tenperaturak izeneko datuak 3 errenkada eta 4 zutabe dituela eta gelaxka bakoitzean zenbaki erreal bat egongo dela. Tauletako gelaxka guztiak mota berdinekoak dira, taulak homogeenak dira alegia.

Datuak izendatzeko programazio lengoaietan jarraitzen diren erregelak hartuko ditugu kontutan. Era honetan, programa sortzerakoan aldagaien izenak mantendu ahal izango ditugu. Erregelak honakoak dira:

1. letra alfabetikoz hasten dira
2. ondoren letra alfabetiko zein karaktere hauek izan dezakete ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '-')

Adibidez, **Kont5** izen zuzena da. Kontutan hartu letra txikiak eta larriak ez direla letra bera. Beraz, **kont5**, nahiz eta letra desberdin bakarra izan, beste datu bat izendatzen du.

Ezagutzen dugun lengoaia matematikoa erabiliko dugu: eragiketak (x , $-$, batukaria, \dots) eta erlazioak ($<$, \leq , \dots). Logika formalaren zenbait ikur ere erabiliko ditugu baldintzak adierazteko. Asko laburtuz,

A eta B edozein adierazpen logiko izanda, hau da, egia ala gezurra bueltatzen dituzten adierazpen erlazioen alak (adibidez, $\text{kont} < 6$), Fig. 3n agertzen diren eragiketa logikoak erabiliko ditugu.

$\neg A$: Ukapena. Egia da baldin eta A gezurra bada.
$A \vee B$: Edo. Egia da baldin eta A egia bada edo B egia bada.
$A \wedge B$: Eta. Egia da baldin eta A eta B aldi berean egia baldin badira.
$A \Rightarrow B$: Ondorioa. Egia da baldin eta A gezurra bada edo B egia bada.

Figure 3: Logika formalaren eragiketak

3.2-ADIBIDEAK

Problema batzuen zehaztapenak ikusiko ditugu guztiz ulertzeko.

PROBLEMA: Bi zenbakien batez bestea

“Bi zenbakien batez bestea” izan daiteke problemaren enuntziatua. Enuntziatua laburra izan da, informazioa eman gabe laga dezakete. Nolako zenbakiak dira? osoak, errealak edozein? **Zehaztapenean erabakiak hartu beharko ditugu problema argia eta zehatza izan dadin.** Adibide honetan zenbakiak errealak izatea erabakiko dugu, eta errealen artean denak, murriztapenik gabe. Horrelako erabaki bat hartzen dugunean justifikatua izan beharko du. Zenbaki errealen multzoak zenbaki osoak ere hartzen ditu barne. Beraz, soilik zenbaki osoen batez bestea kalkulatzeari baino orokorragoa izango da. Gainera, behin problema bat ebatziko badugu, hobe da ahalik eta orokorrena izatea. Bestela pentsa dezagun muturreko kasu hau. Demagun, zenbaki erreal hau besterik ez dugula onartzen: $4'5$. Horrelako bi zenbakien batez bestea beti $4'5$ izango da. Hau ebatzteko programarik ez dugu behar eta gainera beste zenbaki konbinazio infinituen batez bestea ebatzi gabe lagako genuke.

Zehaztapena Fig. 4n erakusten dena izan daiteke.

	AURREBALDINTZA (sarrerako datuak)	ONDORENGO BALDINTZA (emaitzak)
ZEINTZUK DIRA:	A: lehenengo zenbakia B: bigarren zenbakia	C: A eta B arteko batez bestea.
ZEIN MOTAKOAK:	$A \in \mathbb{R} \wedge B \in \mathbb{R}$	$C \in \mathbb{R}$
BALDINTZA INDIBIDUALAK:		
BALDINTZA ERLAZIONALAK:		$C = (A+B)/2$

Figure 4: Bi zenbakien batez bestea problemaren zehaztapena

Esan den bezala, sarrerako datuetan edozein balio erreal onargarria da, horregatik Murriztapenik ez dugu. Emaitzan ezta ere. Konturatu zaitezte sarrerako datuen murriztapenak behartu egiten direla, aldiz, emaitzen murriztapenak berez betetzen dira. Kasu honetan, murriztapenik gabeko bi zenbaki errealen batez bestea baita ere edozein balio erreal izan daiteke. Propietate matematikoki bat da.

Sarrerako datuek ez dute erlazorik beraien artean. Honek ezan nahi du, A eta B ez direla beraien artean baldintzatzen. **Emaitzetan, ordea, beti egongo dira erlazonatuta emaitzak eta sarrerako datuak.**

Erlazio horrek adierazten digu zer lortu nahi den, eta soluzioaren bidean jartzen gaitu. Sarrerako daturen bat emaitzekin erlazionatuta ez balego, beharrezkoa ez den datu bat dela nabarmentzen du. Beraz, ez litzateke sarrerako datua izango.

PROBLEMA: Zenbaki baten faktoriala.

Zenbaki bat emanda bere faktoriala kalkulatu nahi dugu. Adibidez, erabiltzaileak 5 balioa ematen badu 120 emaitza nahi dugu, zeren $5! = 120$. Zehaztapen bat egiten dugunean ahal den kasu posible guztiak kontutan hartzea komeni da. **Zenbat eta denbora gehiago inbertitu zehaztapenean kasu posible guztiak pentsatzen are eta denbora gehiago aurreztuko dugu, zeren eta ustekabe gutxiagoei aurre egin beharko diegu programa egiterakoan.** Ustekabe hauek zehaztapena berregitea suposa dezake gure lana lehen pausotik hasiz. Beraz, sarrerako datu posible guztiak kontutan hartu beharko dugu. Baina, infinitu balio posible dira! Bai baina, multzokatu ditzakegu. Zenbaki bakoitien faktoriala kalkulatu al daiteke? Bai, 5 balioaren faktoriala 120 da. Bikoitiek arazorik ba al dugu? Ez, 4 balioaren faktoriala 24 da. Orokortuz, positiboaren faktoriala kalkulatu al daiteke? Bai, ez dago arazorik. Eta negatiboaren faktoriala kalkulatu ahal da? ummmh! Ez dakit! -4 balioaren faktoriala existitze al da? Eta zenbaki errealeen faktoriala existitzen al da nahiz eta positiboak izan? 5.5 balioaren faktoriala zein da?

Matematiketan (eta beste arlo batzuetan) alde zuretik finkatutako eragiketen definizioa aztertzea komeni zaigu. Definizioak zehazten duena jarraituz ez dago eztabaidarik. Ez dugu erabaki beharrik faktoriala nola ulertzen dugun baizik eta definizioak zehazten duena onartu eta kito. Faktoriala Christian Kramp (1760ko uztailako 8an jaioa, 1826ko maiatzaren 13an zendua) matematikari frantsesak asmatu zuen. Definizioa Wikipedian aurki dezakegu <https://eu.wikipedia.org/wiki/Faktorial>. Uste genuen bezala faktoriala era honetan definitzen da $n! = n \times (n - 1) \times \dots \times 3 \times 2 \times 1$. Horregatik, 5 faktoriala 120 da, zeren $5 \times 4 \times 3 \times 2 \times 1 = 120$. Gainera, definizioak zehazten du n zenbaki natural (baita zenbaki arruntak bezala ezagunak) izan behar duela. Hortaz, zenbaki errealeen faktoriala ez da existitzen, ezin da 5.5 zenbakiaren faktoriala kalkulatu ($5.5! = 4.5 \times 3.5 \times 2.5 \times 1.5 = 324,84375$ kalkulatu daiteke baina ez da faktoriala, beste zeozer da. “Faktoerrela” asmatu dugu!). Orduan, definizioa aztertuz argitu dugu faktoriala zenbaki oso eta positiboena dela soilik. Zero balioa eragiketa matematikoetan nahiko berezia izaten da eta beti kontutan hartzea komeni da. 0 balioaren faktoriala zein da 1 ala 0? Pentsatu.

Kasu orokor eta berezietan pentsatu oztean zehaztapena idatzi dezakegu (ikus Fig. 5).

	AURREBALDINTZA (sarrerako datuak)	ONDORENGO BALDINTZA (emaitzak)
ZEINTZUK DIRA:	n : zenbatgarren faktoriala	em : n -ren faktoriala
ZEIN MOTAKOAK:	$n \in \mathbb{Z}$	$em \in \mathbb{Z}$
BALDINTZA INDIBIDUALAK:	$n \geq 0$	$em \geq 1$
BALDINTZA ERLAZIONALAK:		$em = n!$ non: $n = 0 \Rightarrow n! = 1$ $n > 0 \Rightarrow n! = n * (n-1)!$

Figure 5: Faktoriala problemaren zehaztapena

Zehaztapen honek lehen pentsatu ditugun kasu guztiak kontutan hartzen ditu. n ezin da erreala izan, ezta negatiboa ere. Emaitza sarrerako faktoriala izango da. $n!$ eragiketa matematiko ezaguna da baina zehaztu egingo dugu guztiz argitzeko. n zero denean faktoriala 1 da. n zero baino handiagoa denean emaitza n bider $(n-1)$ balioaren faktoriala izango da. Hizkuntzalaritzan definitzen ari garen kontzeptua ezin da definizioan ageri (adibidez, “euria da lainoetatik erortzen den euria”), baina matematiketan bai: funtzio errekurtsiboa deritzo. Baina zuzena izan dadin emaitza ematen duen oinarritzko kasu batetara ailegatu behar da. Kasu honetan egoki definituta dago. Ikusi dezagun. Demagun n -k 5 balioa duela. Definizio errekurtsiboaren

arabera bere faktoriala da $5 \times (5-1)!$. Azken emaitza lortzeko $4!$ kalkulatu beharko da: $4 \times (4-1)!$ izanik. Hau da errekursioa, behin eta berriro gauza bera egitea. Errepikapen hau noiz bait bukatu beharko litzateke emaitza bat lortzeko. Gure kasuan, balioa unitate batean txikiagotzen doa. 4, 3, 2, 1 eta azkenean 0 izango da. Hau da oinarrizko kasua, $0!$ da 1. Beraz, $5!$ da $5 \times 4 \times 3 \times 2 \times 1 \times 1$. Emaitza 120.

PROBLEMA: Instalazio geotermiko eraginkorrena.

Problemak formula matematiko bat baino askoz konplikatugoak izan daitezke. Adibidez, problema honetan instalazio geotermiko bertikal baten instalazio eraginkorrena kalkulatu nahi dugu. Hau da, beherantz lur azpiko berotasuna hobekien harrapatzen duen hodian instalazioa. Hodiek “U” forma izango dute derrigorrez (ikus 3. irudia). Hodiaren luzera ez da murriztapen bat adibide honetan. Beste kasu betetan izan ahal izango litzatekeen arren. Adibidez, kostu ekonomikoa kontutan hartuko bagenu.

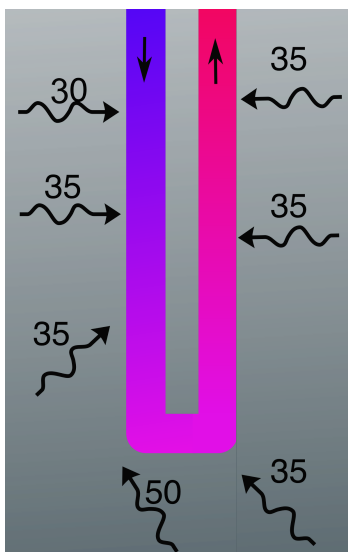


Figure 6: Geotermia hodia lur azpiko berotasuna harrapatzen [Irudia:Geotermia].

Benetakoa ez den berotasun harrapaketa modelo hau erabiliko dugu: hodia pasatzen den tokiko berotasun guztia hartzen du hodi barneko likidoak. Termodinamikak ez du modelo hau onartzen beste bero-tasun galerarik ez duelako. Baina gure oraingo helburua zehaztapenak ulertzea da, eta ez formula zuzenak erabiltzea.

Problemaren ebazpena instalazio eraginkorrena definitzen duten datuak izango dira. Demagun lur azpiko tenperaturen laginak ditugula metroz metro, 10 metro gure etxetik aurrera, eta beste hamar metro lur behera (ikus 4. taula). 10 metro horizontal horietan hodiaren sarrerako zuloa eta irteerako zuloa non kokatzen diren jakin behar dugu. Horretaz gain, beherantz hodia zenbat metro lurperatu behar den. Beraz soluzioa hiru datu hauek definitzen dute: sarrerako zuloaren kokapena, irteerako zuloaren kokapena, eta hodiaren sakonera. Beti ere kontutan hartuta 10 metro aurrera eta 10 metro beherako murriztapen hori dugula.

Problema honen zehaztapena 4. irudian erakusten dena izan daiteke.

Zehaztapenak dioenaren arabera, problemaren datu sarrera 10×10 balio erreal dituen taula bat da. Hamar metroko luzeran eta hamar metroko sakoneran hartutako laginen balioa izango dira. Balio erreal guztiak

behera	1	2	3	4	5	6	7	8	9	10
1	30	20	45	20	35	45	20	15	10	35
2	35	35	40	25	35	40	25	45	20	35
3	35	35	35	25	20	35	25	40	25	20
4	50	35	30	45	25	30	45	35	25	25
5	45	20	30	45	30	45	15	30	45	50
6	40	25	30	35	30	35	45	20	30	45
7	35	25	25	25	25	25	40	25	30	35
8	30	45	30	45	40	25	35	25	25	25
9	30	35	30	35	35	25	30	45	25	25
10	25	25	25	25	30	45	40	40	45	45

Table 2: Temperatura laginak, 10 metro aurrera eta 10 behera.

	AURREBALDINTZA (sarrerako datuak)	ONDORENGO BALDINTZA (emaitzak)
ZEINTZUK DIRA:	Laginak: 10x10 taula laginekin	SarZuloa: sarrerako zuloa zein metroan. IrtZuloa: irteerako zuloa zein metroan. Sakonera: behera zenbat metro.
ZEIN MOTAKOAK:	$Laginak(10,10) \in \mathbb{R}$	$SarZuloa \in \mathbb{Z} \wedge IrtZuloa \in \mathbb{Z} \wedge Sakonera \in \mathbb{Z}$
BALDINTZA INDIBIDUALAK:		$SarZuloa > 0 \wedge SarZuloa < 11$ $\dots \wedge IrtZuloa > 0 \wedge IrtZuloa < 11$ $\dots \wedge Sakonera > 0 \wedge Sakonera < 11$
BALDINTZA ERLAZIONALAK:		$SarZuloa < IrtZuloa$ $\sum_{i=1}^{Sakonera} (Laginak(i, SarZuloa) + Laginak(i, IrtZuloa)) +$ $\sum_{j=(SarZuloa+1)}^{IrtZuloa-1} (Laginak(Sakonera, j) + Laginak(Sakonera, j))$ batura maximoa da beste edozein Sakonera, SarZuloa eta IrtZuloa konbinazioarekiko

Figure 7: “U” forma duen geotermia instalazioaren zehaztapena

onartzen dira, baldintza indibidual eta erlazionalik ez dagoenez. Hau da, temperatura negatibo bat (lurra izoztuta dagoenean) balizko balio bat da. Ondorengo baldintzak zehazten duenez emaitza hiru datuez osatua dago: non zulatu hodia sartzeko, nondik atera hodia eta zein sakonera arte sartu hodia. Hiru balio hauek 1 eta 10 arteko balio osoak izan behar dute. Baina ez du balio edozein balio konbinazioa, ematen den formula maximizatzen duten balioak baizik. Gainera, bete baldintza bat bete behar dute, sarrerako zuloaren kokapena irteerakoa baino txikiagoa izan behar du.

PROBLEMA: Labirinto baten irteera

Kasu honetan, sarrera labirintoaren mapa bat izango da eta emaitza mapa bera irteerako bidea markatuta duela. Labirinto mota asko dagoenez murriztapen hauek jarriko dizkiegu: bide guztiek unitate bateko zabalera dute, labirintoak sarrera bakarra izango du goi-ekzker iskinean, irteera bakarra izango du behe-eskuin iskinean, ez du begiztarik (biribilgunerik).

Gu gizakiok mundu fisikoa modelizatu egiten dugu problemez errazago pentsatzeko. Labirinto fisikoa koloretzatutako mapa batean bihurtzen dugu: zuria bidea, beltza pareta. Ondoren margoekin bide egokia bilatzen hasten gara: orlegia irteerako bidea, gorria bide itxia. Era berean, **ordenagailuek mundu fisikoa (ala teorikoa) modelizatuta jaso behar dute, ondoren prozesatu eta soluzioa bilatzeko**. Labirintoaren irudi eskaneatuta eman ahal diogu ordenagailuari baina orduan soluzioa saildu egiten da irudi prozesaketarako programa bat beharko genukeelako. Hobe da, labirintoa zenbaki-taula modura kodetuta ematen

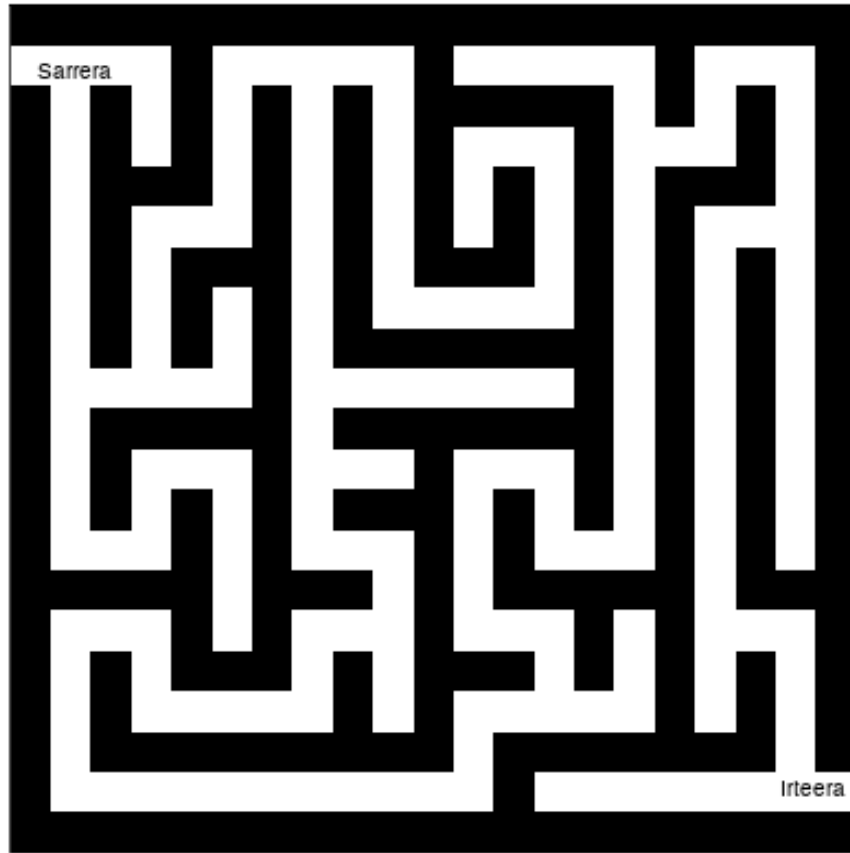


Figure 8: Labirintoaren mapa

badiogu ordenagailuari. Hau izan daiteke kodeketa: esploratu gabeko bidea 1 balioa, pareta 2 balioa, bide itxia 3 balioa, bide egokia 4 balioa.

Zehaztapena Fig. 10 irudian erakusten da.

3.3-ARIKETAK:

Egin ondorengo ariketen enuntziatua.

MATEMATIKA:

-Zenbaki bat bestearen zatitzailea al da?

-ZKH

-MKT

-N-garren Fibonacci zenbakia

-Zenbaki lehena al da?

JOLASAK: kontutan izan zehaztapenak ez duela zehazten jokoaren dinamika baizik eta sarrera eta irteera.

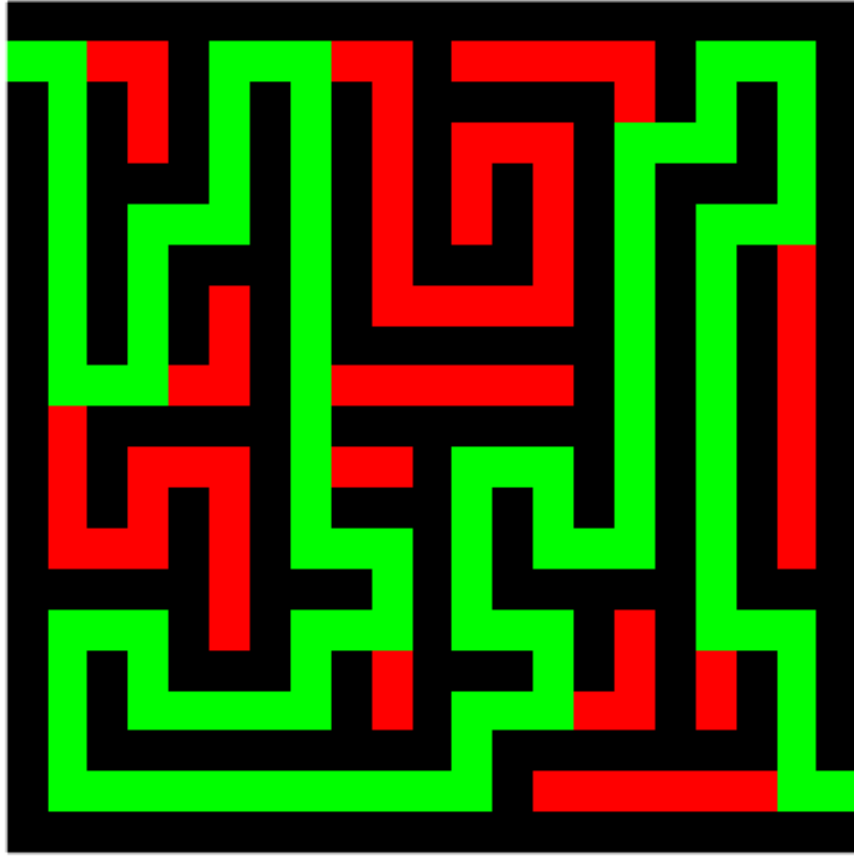


Figure 9: Labirintoa bide zuzena markatuta duela

	AURREBALDINTZA (sarrerako datuak)	ONDORENGO BALDINTZA (emaitzak)
ZEINTZUK DIRA:	labirintoa: $n \times n$ dimentsioko taula	labirintoa: $n \times n$ dimentsioko taula suzeneko bidea markatuta duela
ZEIN MOTAKOAK:	$labirintoa(n, n) \in \mathbb{Z}$	$labirintoa(n, n) \in \mathbb{Z}$
BALDINTZA INDIBIDUALAK:	$labirintoa(n, n) \in \{1, 2\}$	$labirintoa(n, n) \in \{1, 2, 3, 4\}$
BALDINTZA ERLAZIONALAK:		Labirintoak sarrera irterarekin lotzen duen bide bat izango du 4 balioaz markatuta.

Figure 10: Labirintoaren zehaztapena

-Harri-orri-ar ordenagailuaren kontra (<https://eu.wikipedia.org/wiki/Harri-orri-ar>).

-Ez kutuko zenbaki bat asmatu. Ordenagailuak 1 eta 99 arteko zenbakia ausaz aukeratuko du. Erabiltzaileak ze zenbaki den asmatu behar du. Ordenagailuak.

ENERGIA BERRIZTAGARRIEN TESTUINGURUAN:

-????

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	2
2	1	2	1	2	1	2	1	2	1	2	2	2	2	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1	2	1	2	1	1	1	2	1	1	1	2	1	2
2	1	2	2	2	1	2	1	2	1	2	1	2	1	2	1	2	2	2	1	2
2	1	2	1	1	1	2	1	2	1	2	1	2	1	2	1	2	1	1	1	2
2	1	2	1	2	2	2	1	2	1	2	2	2	1	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1	2	1	1	1	1	1	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1	2	2	2	2	2	2	2	1	2	1	2	1	2
2	1	1	1	1	1	2	1	1	1	1	1	1	1	2	1	2	1	2	1	2
2	1	2	2	2	2	2	1	2	2	2	2	2	2	2	1	2	1	2	1	2
2	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1	2	2	2	1	2	1	2	1	2	1	2	1	2
2	1	1	1	2	1	2	1	1	1	2	1	2	1	1	1	2	1	2	1	2
2	2	2	2	2	1	2	2	2	1	2	1	2	2	2	2	2	1	2	2	2
2	1	1	1	2	1	2	1	1	1	2	1	1	1	2	1	2	1	1	1	2
2	1	2	1	2	2	2	1	2	1	2	2	2	1	2	1	2	1	2	1	2
2	1	2	1	1	1	1	1	2	1	2	1	1	1	1	1	2	1	2	1	2
2	1	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	1	2
2	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Table 3: Labirintoaren kodeketa: 1 bidea, 2 pareta

3.4-IDEI NAGUSIAK:

-Zehaztapenak problema guztiz definitzen du sarrera eta irteerako datuak nolakoak izan behar duten adierazita.

-Enuntziatuak argiak ez badira gure kabuz erabaki beharko ditugu hainbat gauza problema definitu aldera.

-Kasu orokor eta berezietan pentsatzea komeni izaten da.

-Ahal bada, lengoaia logiko-matematikoa erabiliko da, bestela azalpenak gure hitzez emango ditugu.

4- ALGORITMOAK

Harluxet hiztegi-entziklopedikoaren [Harluxet] arabera, algoritmoa da:

“Problema baten ebazpenerako eman behar diren urratsen deskribapen formala.
Programazio-lengoaia baten bidez, algoritmoa ordenadore batek egikari dezakeen programa bihurtzeko daiteke.”

Algoritmoa programaren diseinua da, ondoren programa bihurtzen dena. Ordenadoreak, gizakiok ez bezala, ez dituzte aginduak interpretatzen, zehatz-mehatz jarraitzen dituzte. Horregatik algoritmoek (eta ondoren programek) ezaugarri hauek izan behar dituzte:

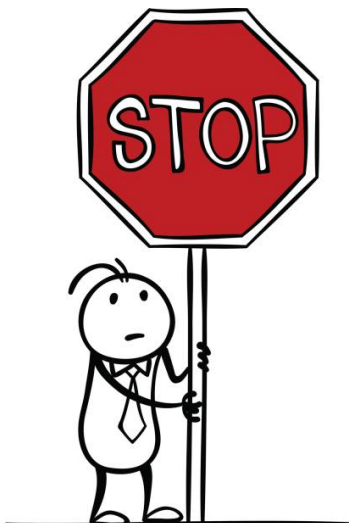


Figure 11: Ez irakurri gehiago!!!



Figure 12: Ondorengo testuek lanketa behar dute

- Zehatzak izan behar dute
- Pausoen arteko ordena azaldu behar dute
- Sarrerako datu berdinekin, bi aldiz algoritmoak dioenari jarraituz, emaitza berdina izan behar du
- Beti amaitu behar dute
- Programazio lengoaiarekiko independentea izan behar dute

Algoritmoak zehatzak izan daitezzen **programazio inperatiboa** erabiliko dugu, hau da, **agindu multzo ezagun eta mugatu bat** erabiliko dugu. **Aginduak datuak ordenagailura sartu, prozesatu eta ateratzeko balio dute.** Agindu hauek ordenagailuarentzat ulergarriak dira eta algoritmoak adierazten duen ordenan egikariturako ditu. Era honetan, beti sarrerako balio berdinekin emaitza berdina lortuko da. Baina, zenbait problema ezin dira ebatzi agindu kopuru zehatz bat egikarituta. Adibidez, erabiltzaileak ematen duen 0 balioaz bukatzen den zenbaki multzo baten batura kalkulatu nahi badugu, zenbat aldiz eskatu behar du algoritmoak zenbaki berri bat? Hamar zenbaki izango balira, hamar balio eskatu eta batzearekin nahiko izango litzateke. Tamalez, aldeztu aurretik zenbat zenbaki izango diren ezin dezakegu jakin. Horrelako problema ebazteko **programazio egituratua** erabiliko dugu. **Kontrol egitura berezi batzuk** erabiliko ditugu zeintzuk baldintzen arabera aginduen exekuzio lineala alda dezaketen. **Egitura errepikakorrak (agindu multzo bat behin eta berriro egikaritzen dutenak) eta baldintzapekoak (agindu multzo bat baldintza baten arabera egikariturako dena).** Adibidez, datu berri bat eskatu eta batu sarrerako datua zero den arte. Konturatu gaituz kontrol egiturak erabiliz algoritmoak asko laburtu

daitezkeela. Ehun zenbaki batu behar baldin baditugu, ehun aldiz balioa eskatu eta batu beharrean, kontrol egitura errepikakor baten bitartez errepikatu dezakegu balio bat eskatu eta batu. Kontagailu baten bitartez zenbat zenbaki batu diren kontrolatu daiteke (ehun kontatu arte). Algoritmoa motzago eta ulergarriagoa izango da.

Hala eta guztiz ere, algoritmoak asko luzatu daitezke. Horregatik, **programazio modularra** ere erabiliko dugu. Programazio mota honetan moduluak sortu eta erabiltzen dira. Moduluak algoritmo batek erabiltzen dituen beste algoritmoak dira. Aldez aurretik egindako algoritmoak berrerabili daitezke beste algoritmoetan. Hauei modulu, azpialgoritmo edo azpiprogramak deitzen diegu. Abantailak asko dira:

- Problema handi bat problema txikiagotan zatituz errazago ebatzi daiteke
- Algoritmoak laburragoak eta ulergarriagoak
- Behin egin eta askotan erabiliz denbora aurreztu eta akats gutxiago sortzen dira

4.1-Programazio inperatiboa

Programazioan, matematiketan ez bezala, eragin bat duten egin beharreko bezala, eragin bat duten egin beharreko aginduak ematen ematen dira. Matematiketan $X = 5$ jartzen denean egi bat esaten ari gara: “X aldagaiak 5 balioa du”. X aldagaiak balio hori du. Aldiz, programazioan $X = 5$ jartzen dugunean, ordenagailuari agintzen diogu X aldagaian 5 balioa gordetzeko. Esleipen agindua da, X aldagaiaren balioa aldatzen du.

DATUAK

Problemak ebazteko datuak prozesatu behar dira. Sarrerako datuetatik abiatuta, tarteko datuak kalkulatu, azkenik emaitza lortu arte. Orokorrean algoritmoek, programazio lengoaietatik eratorritako datu mota hauekin egiten dute lan:

- zenbakiak: zenbaki osoak (adibidez 8) eta errealak (adibidez 4’5).
- karaktereak: ASCII taulan (<https://eu.wikipedia.org/wiki/ASCII>) aurkitzen diren ikurrak dira. Adibidez, letra alfabetikoak (‘a’, ‘b’, ‘c’,...) eta ordenagailuko teklatuan aurki ditzakegunak.
- boolear edo logikoa: egia (bai, betetzen da) eta gezurra (ez, ez da betetzen) balio bitarra. Baldintza eta konparaketen emaitza izaten da.

Datu mota hauei sinpleak deritze. Sintaxi edo idazteko era zehatza jarraitu behar da. Zenbaki osoak eskuz idazten ditugun bezala idazten dira (4). Zenbaki errealak koma (behean ala goian) izan beharrean puntua dute. Hau da, 4’5 idatzi beharrean 4.5 idazten da. Karaktereak kakotxen artean idazten dira: kakotx kakotx sinplearekin ‘a’ edo bikoitzarekin “a”. M lengoaian kakotx sinplea sinplea erabiltzen da. Balio boolearrak, aldiz, ingelezko *true* eta *false* hitzez adierazten dira.

Datuak Datuak sinpleak izan daitezke (balio bakarra) ala datu egitura konposatuak konposatuak (balio asko). Ohikoen datu egitura konposatua taula da: dimentsio dimentsio batekoa, bikoa ala nahi haina dimentsioetakoa. Taulak homogeneoak homogeneoak izaten dira. Alegia, datu guztiak mota berdinekoak izan behar dute. dute. Zenbakiez osatuta badago, sektorea edo matrizea deitzen zaio. Karakterez Karakterez osatuta badago hitza edo esaldia da. Programazio lengoaietan esaldiak esaldiak izendatzeko *string* ingelesezko izena erabiltzen da. Hauek Hauek karaktere solteak bezala adierazten dira, kakotxen artean, adibidez, adibidez, ‘kaixo, zer moduz zaude?’.

MATLAB, MATrix LABoratory esaldiaren laburpena da. M lengoia matrizeekin lan egiteko pentsatuta dago. Matrizea (ala taula) oinarritzko datu mota du eta eragozpenik gabe eta era eraginkorrean erabiltzen ditu. M lengoian taulak kortxete artean adierazten dira.

ADIBIDEA

[10 20 30] hiru zenbaki osoen bektorea da. Zehazki hamar, hogei eta hogeita hamar balioez osatuta dago. Gai bakoitza beteetatik hutsunez edo komaz desberdintzen da. [10,20 30] bektore berdina da.

‘;’ ikurrak errenkadak desberdintzen ditu. Hortaz, [1 2 3; 4 5 6] bi dimentsioko taula adierazten du. Bi errenkada eta hiru zutabe dituen.

ALDAGAIK

Programazio lengoaietan datuak (balioak) aldagaietan gordetzen dira prozesaketa egin ahal izateko. Aldagaiak izendatu egiten dira erregela hauei jarraituz:

1. letra alfabetikoz hasten dira
2. ondoren bai letra alfabetiko zein karaktere hauek izan dezakete (‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’, ‘.’)

ADIBIDEA

kont, *Z4* eta *X* aldagaien izenak izan daitezke.

Aldagaiak momentu bakoitzean balio bakarra gordetzen dute. Baina balio hori aldatu daiteke esleipena eragiketaren bitartez. ‘=’ ikurrak esleipena adierazten du eta bere eragina da bere ezker aldean dagoen aldagaiak eskuin aldeko balioa hartzen duela. Era honetan: $\langle \text{ALDAGAIARENIZENA} \rangle = \langle \text{BALIOA} \rangle$. Balioa sinplea izan daiteke ala konposatua. Datu egitura baldin bada eta datu bakarra aldatu nahi badugu, zein posizioan kokatu nahi dugun adierazi beharko dugu. Adibidez, bektore bat bada (dimentsio bakarreko taula) era honetan egingo dugu: $\langle \text{ALDAGAIARENIZENA} \rangle (\langle \text{POSIZIOA} \rangle) = \langle \text{BALIOA} \rangle$. Parentesiak adierazten du bektoreko zein posiziotan gorde behar den balio sinplea. Aldiz, bi dimentsioko taulabatetan, $\langle \text{ALDAGAIARENIZENA} \rangle (\langle \text{ERRENKADAZENBAKIA} \rangle, \langle \text{ZUTABEZENBAKIA} \rangle) = \langle \text{BALIOA} \rangle$ jarri beharko dugu. Bai posizioak zein errenkada eta zutabeak 1 zenbakitik hasten dira kontatzen. Datuak atzitzeko ere era horretan egiten da.

ADIBIDEA

kont = 5 bost balioa esleitzen dio *kont* aldagaia.

nireBektorea = [1 2 3 4] lau gaien bektorea gordetzen da *nireBektorea* aldagaian.

nireBektorea (2) = 22 *nireBektorea* aldagaian gordeta dagoen bektorea aldatu da, [1 22 3 4] da orain.

nireTaula = [10 20 30; 40 50 60] bi dimentsioko taula gordetzen da *nireTaula* aldagaian.

nireTaula (1, 3) = 33 lehen errenkadan hirugarren zutabea 33 balioa gorde da. Taula da [10 20 33; 40 50 60]

nireTaula (1, 1) = *nireTaula* (2,3) Taula da [60 20 33; 40 50 60]

ERAGILEAK

Prozesaketa egin ahal izateko eragileek datuak moldatu egiten dituzte. Ohiko eragile matematikoak programazio lengoaietan hauek dira:

- batuketa: +
- kenketa: -
- biderketa: *
- zatiketa: /
- berreketa: ^

Eragiketa matematikoak zenbakiekin lan egiten dute (edo zenbakiak gordetzen dituzten aldagaiekin) eta emaitza zenbaki bat da baita ere. Eragile erlazionalek balio boolearra bueltatzen dute:

- handiago: >
- txikiago: <
- berdina: ==
- desberdina: ~=
- handiago edo berdina: >=
- txikiago edo berdina: <=

Lokarri logikoak ere erabili daitezke:

- edo: ||
- eta: &&
- ukapena: ~

Aurreko eragile guztiak bitarrak dira eta eragingaien artean kokatzen dira. Emaitzak esleipenaren bitartez aldagaietan gordetzea komeni da.

ARIKETA

```
>> Batura = 5+3
```

```
>> BatezBestea = Batura / 2
```

```
>> HandiagoaAlDa = 5 > BatezBestea
```

Aurreko hiru agindu hauek jarraian goitik behera egikaritzuz, zer balio izango dute aldagaiek?

M lengoaiaren eragingaiak taulak izan daitezke.

BARNE FUNTZIOAK

Programazio lengoaiak eragiketak konplexuagoak ere eskaintzen dituzte baina hauek ez dira eragile bitarrak izaten bizik eta barne funtzioak. Hau da, programazio lengoaiak aldezturik aurretik programaturako programak lengoaiaren zati bezala eskaintzen direnak. Barne funtzioak programazio lengoaiaren guztietan ageri dira baina zeintzuk eta hauen izenak desberdinak izan ohi dira. Kontutan izan behar da funtzioak egin behar berezia dutela eta programazio lengoaiaren ikuspegia islatzen dutela. Adibidez, M lengoaiaren matrizeekin lan egiteko espezializatuta dagoenez erro karratua kalkulatzeko barne funtzioa, ***sqrt***(***<DATUA>***), matrizeekin lan egin dezake.

ADIBIDEA

sqrt ([4 9 16]) bueltatuko du [2 3 4]. Hau da, emandako bektorearen gai bakoitzaren erro karratua. Sarrera eta emaitzen arteko lotura posizioa da: lehenengo balioaren erro karratua lehen posizioan dagoen emaitza eta abar.

Erro karratua nola kalkulatu duen ez dakigu (eta ez zaigu axola), ezagutu beharrekoa da zer sarrerako argumentu eman behar zaion eta zer emaitza bueltatuko digun. Barne funtzio bat egikaritu ahal izateko bere izena jakin behar dugu eta bere sarrerako argumentuak zeintzuk diren eta zein ordenean eman behar zaizkion. Bueltatzen duen emaitza esleipenaren bitartez aldagai batean gorde dezakegu ondoren kalkulu gehiago egiteko. Barne funtzioak paketeetan multzokatuta izaten dira duten helburuaren arabera. Adibidez, bi dimentsioko grafikoak marrazteko paketea, karaktere kateekin lan egitekoa eta abar.

IRADOKIZUNA

MATLAB eta Octave GIBetako agindu leihuan ***help*** agindua egikaritzuz pakete guztiak zerrendatzen ditu. ***help*** ***<PAKETEARENIZENA>*** agindua pakete horretako funtzio guztiak zerrendatzen ditu. ***help*** ***<FUNTZIOARENIZENA>*** agindua egikaritzuz barne funtzio horren sintaxia eta adibideak jasoko dituzu. ***doc*** agindua ***help*** agindua bezalakoa da baina informazio zabalagoa ematen du.

PROZESAKETA

Orain arte ikusitakoarekin datuak prozesatu ditzakegu MATLAB edo OCTAVEko agindu leihuetan. Adibidez, demagun 100, 88 eta 65 zenbakien batez bestea kalkulatu nahi dugula. Hiru zenbaki horiek batu eta batura zati hiru egitearekin nahiko genuke. Ondorengo idatzi dezakegu agindu leihuan:

KODEA:

```
>> batura = 100 + 88 + 65
```

```
>> bb = batura / 3
```

Bi agindu hauekin *bb* aldagaian batez bestea kalkulatu genuke. Kalkulagailu baten modura erabiltzen ari gera GIBa. Prozesaketa sinplea denean (kalkulu eta datu gutxi) horrela egin dezakegu. Baina normalean prozesaketa konplexuagoa izaten da eta “kalkulagailuan” egin beharreko lanak zaildu egiten dira. Adibidez, demagun hilabete bateko euri jasan batez besteak kalkulatu nahi ditugula. Orain ez dira hiru datu baizik eta hogeita hamar. Hogeita hamar datu hauek banaka kopiatzea ez da oso eraginkorra (imajinatu urte osoko datuak direla). Seguruenik datuak fitxategi batean izango ditugu. Hauek agindu baten bitartez taula batean kopiatu eta ondoren prozesatuko genuke. Adibidez era honetan:

KODEA:

```
>> euria = load ('euria.dat', '-ascii')
```

```
>> batura = sum (euria)
```

```
>> bb = batura / length(euria)
```

load, *sum* eta *length* M lengoaiaren barne funtzioak dira. *load* funtzioak *euria.dat* fitxategiko hileko euri jasan datuak irakurri eta, esleipenaren bitartez, *euria* aldagaian gordetzen ditu. *sum* funtzioak aldiz bektoreko datu guztiak batzen ditu. Azkenik *length* funtzioak bektorearen luzera ematen du. Era honetan, hilabete osoko euri jasan batez bestea kalkulatzeko dugu.

Demagun orain asteburuetako batez bestea kalkulatu nahi dugula. Prozesaketa konplexuagoa da, datu batzuk baztertu behar dira eta beste batzuk kontutan hartu. Prozesaketa baldintzatuta egongo da, batu soilik 6., 7., 13, 14, ... eguneko datuak. Gainera zenbat diren jakin beharko dugu batez bestearen zatiketa egiteko. Era honetako prozesaketa baldintzatu edo konplexua dugunean programa bat idatzi behar dugu prozesaketa kontrolatzen duten kontrol egitura bereziak erabiliz.

Demagun baita asteko egunez eguneko batez besteak nahi digula ere. Gure kabuz kalkulatu dezakegu GIBa kalkulagailu baten modura erabilita, baina denbora asko beharko dugu. Gainera geroz eta konplexuagoa izan daiteke kalkulatu beharrekoa. Adibidez urte osoko datuen batez bestea astelehenetik ostirala arteko egunen batez bestea eta asteburuen batez bestea (larunbat eta igandeak elkarrekin). Gogoratu aurreko atalean esandakoa, datu asko edo egin eragiketa konplexuak direnean programa bat egitea komeni dela.

IDEI NAGUSIAK

- Datuak sinpleak (zenbakiak eta boolearrak) aldagaietan gordetzen dira esleipenaren bitartez (=).
- M lengoaiaren matrizeak (taulak) era naturalean erabiltzen dira datu bezala.
- Eragile matematiko (+, -, /, *), erlazional (<, >, ==, <=, >=) eta logiko (&&, ||) sinpleak bi eragingai hartzen dituzte. Eragile negatiboa (-) eta ukapen logikoa (~) izan ezik.
- Eragile konplexuagoak programazio lengoaiak eskaintzen dituen funtzioak (azpi-programak) izaten dira.
- M lengoaiak barne funtzioak paketeetan multzokatzen ditu.
- GIBa kalkulagailu baten modura erabili dezakegu zenbait kalkulu egiteko. Datu asko, eragiketa konplexu edo prozesaketa baldintzatua dugunean programa bat egitea komeni da.

mmmm

5-SIMULAZIO TAULAK

ldsjfdfsldjfdsl

6- OINARRIZKO PROZESAKETA M LENGOAIA (GIBa KALKULAGAILU MODURA ERABILI)

M lengoaia Mathworks enpresak (<https://es.mathworks.com/>) asmatutako programazio lengoaia da. M lengoaian idatzitako programak ez dira zuzunean exekutagarriak, IDE (Integrated Development Environment edo Garapenerako Ingurune Bateratura) batek egikaritzen ditu. Hau da, interpretatuak dira: aginduak banaka banaka irakurri eta exekutatzen dira. MATLAB (https://es.mathworks.com/products.html?s_tid=gn_ps) edo debaldeko alternatiba Octave GNU (<https://www.gnu.org/software/octave/>) erabili daiteke. Bi garapen inguruneen erabilpena eta zati nagusiak berdinak dira:

1. Fitxategi kudeaketa. Programa-fitxategiak menu aukeren bitartez kudeatzen ohi dira: ireki, itxi, berria, etab.
2. Editorea. Programa berri bat idazteko erabilgarriak diren laguntzak: bilatu eta aldatu, sintaxi-koloreztaketa automatikoa, etab.
3. Agindu leihoa. Erabiltzaileak zuzenean aginduak helarazi ahal dizkio inguruneari exekutatu ditzan.
4. Lan-gunea. Exekuzioan erabilgarri dauden datuak gordetzen ditu.

DATUAK

Problemak ebazteko datuak prozesatu behar dira. Sarrerako datuetatik abiatuta, tarteko datuak kalkulatu, azkenik emaitza lortu arte. Orokorrean programazio lengoaiak datu mota hauekin egiten dute lan:

- zenbakiak: zenbaki osoak (adibidez 8) eta errealeak (adibidez 4'5).
- karaktereak: ASCII taulan (<https://eu.wikipedia.org/wiki/ASCII>) aurkitzen diren ikurrak dira. Adibidez, letra alfabetikoak ('a' adibidez) eta ordenagailuko teklatuan aurki ditzakegunak.
- boolear edo logikoa: egia (bai, betetzen da) eta gezurra (ez, ez da betetzen) balio bitarra. Baldintza eta konparaketan emaitza izaten da.

Hauei datu sinpleak deritze eta sintaxi (idazteko era) bat jarraitu behar da programazio lengoaiak datu ulertu dezaten. Lengoaia bakoitzean idazteko era bat egon daiteke baina ohikoena da zenbakiak eskuz idazten ditugun bezala idaztea salbuespen batekin. Zenbaki errealeak koma (behean ala goian) izan beharrean puntua dute. Hau da, 4'5 idatzi beharrean **4.5** idazten da. Karaktereak kakotxen artean idazten dira: kakotx sinplearekin 'a' edo bikoitzarekin "a". M lengoian kakotx sinplea erabiltzen da. Balio boolearrak, aldiz, ingelezko **true** eta **false** hitzez adierazten dira.

Datuak sinpleak izan daitezke (balio bakarra) ala datu egitura konposatuak (balio asko). Ohikoen datu egitura konposatua taula da: dimentsio batekoa, bikoa ala nahi haina dimentsioetako. Taulak homogeneoak izaten dira. Alegia, datu guztiak mota berdinekoak izan behar dute. Zenbakiez osatuta badago, bektorea edo matrizea deitzen zaio. Karakterez osatuta badago hitza edo esaldia da. Programazio lengoaietan esaldiak izendatzeko **string** ingelesezko izena erabiltzen da. Hauek karaktere solteak bezala adierazten dira, kakotxen artean, adibidez, 'kaixo, zer moduz zaude?'.

MATLAB, MATrix LABoratory esaldiaren laburpena da. M lengoaia matrizeekin lan egiteko pentsatuta dago. Matrizea (ala taula) oinarritzko datu mota du eta eragozpenik gabe eta era eraginkorrean erabiltzen ditu. M lengoian taulak kortxete artean adierazten dira.

ADIBIDEA

[10 20 30] hiru zenbaki osoen bektorea da. Zehazki hamar, hogei eta hogeita hamar balioez osatuta dago. Gai bakoitza beteetatik hutsunez edo komaz desberdintzen da. [10,20 30] bektore berdina da.

‘;’ ikurrak errenkadak desberdintzen ditu. Hortaz, [1 2 3; 4 5 6] bi dimentsioko taula adierazten du. Bi errenkada eta hiru zutabe dituen.

ALDAGAIK

Programazio lengoaietan datuak (balioak) aldagaietan gordetzen dira prozesaketa egin ahal izateko. Aldagaiak izendatu egiten dira erregela hauei jarraituz:

1. letra alfabetikoz hasten dira
2. ondoren bai letra alfabetiko zein karaktere hauek izan dezakete (‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’, ‘-’)

ADIBIDEA

kont, *Z4* eta *X* aldagaien izenak izan daitezke.

Aldagaiak momentu bakoitzean balio bakarra gordetzen dute. Baina balio hori aldatu daiteke esleipena eragiketaren bitartez. ‘=’ ikurrak esleipena adierazten du eta bere eragina da bere ezkerrean dagoen aldagaiak eskuin aldeko balioa hartzen duela. Era honetan: $\langle \text{ALDAGAIARENIZENA} \rangle = \langle \text{BALIOA} \rangle$. Balioa sinplea izan daiteke ala konposatua. Datu egitura baldin bada eta datu bakarra aldatu nahi badugu, zein posizioan kokatu nahi dugun adierazi beharko dugu. Adibidez, bektore bat bada (dimentsio bakarreko taula) era honetan egingo dugu: $\langle \text{ALDAGAIARENIZENA} \rangle (\langle \text{POSIZIOA} \rangle) = \langle \text{BALIOA} \rangle$. Parentesiak adierazten du bektoreko zein posiziotan gorde behar den balio sinplea. Aldiz, bi dimentsioko taulabatetan, $\langle \text{ALDAGAIARENIZENA} \rangle (\langle \text{ERRENKADAZENBAKIA} \rangle, \langle \text{ZUTABEZENBAKIA} \rangle) = \langle \text{BALIOA} \rangle$ jarri beharko dugu. Bai posizioak zein errenkada eta zutabeak 1 zenbakitik hasten dira kontatzen. Datuak atzitzeko ere era horretan egiten da.

ADIBIDEA

kont = 5 bost balioa esleitzen dio *kont* aldagaiari.

nireBektorea = [1 2 3 4] lau gaien bektorea gordetzen da *nireBektorea* aldagaian.

nireBektorea (2) = 22 *nireBektorea* aldagaian gordeta dagoen bektorea aldatu da, [1 22 3 4] da orain.

nireTaula = [10 20 30; 40 50 60] bi dimentsioko taula gordetzen da *nireTaula* aldagaian.

nireTaula (1, 3) = 33 lehen errenkadan hirugarren zutabea 33 balioa gorde da. Taula da [10 20 33; 40 50 60]

nireTaula (1, 1) = *nireTaula* (2,3) Taula da [60 20 33; 40 50 60]

ERAGILEAK

Prozesaketa egin ahal izateko eragileek datuak moldatu egiten dituzte. Ohiko eragile matematikoak programazio lengoaietan hauek dira:

- batuketa: +
- kenketa: -
- biderketa: *
- zatiketa: /
- berreketa: ^

Eragiketa matematikoak zenbakiekin lan egiten dute (edo zenbakiak gordetzen dituzten aldagaiekin) eta emaitza zenbaki bat da baita ere. Eragile erlazionalek balio boolearra bueltatzen dute:

- handiago: >
- txikiago: <
- berdina: ==
- desberdina: ~=
- handiago edo berdina: >=
- txikiago edo berdina: <=

Lokarri logikoak ere erabili daitezke:

- edo: ||
- eta: &&
- ukapena: ~

Aurreko eragile guztiak bitarrak dira eta eragingaien artean kokatzen dira. Emaitzak esleipenaren bitartez aldagaietan gordetzea komeni da.

ARIKETA

```
>> Batura = 5+3
>> BatezBestea = Batura / 2
>> HandiagoaAlDa = 5 > BatezBestea
Aurreko hiru agindu hauek jarraian goitik behera egikaritzuz, zer balio izango dute aldagaiek?
```

M lengoaian eragingaiak taulak izan daitezke.

BARNE FUNTZIOAK

Programazio lengoaiak eragiketak konplexuagoak ere eskaintzen dituzte baina hauek ez dira eragile bitarrak izaten bizik eta barne funtzioak. Hau da, programazio lengoaiak aldeztu aurretik programatutako programak lengoaiaren zati bezala eskaintzen direnak. Barne funtzioak programazio lengoaiaren guztietan ageri dira baina zeintzuk eta hauen izenak desberdinak izan ohi dira. Kontutan izan behar da funtzioak egin behar berezia dutela eta programazio lengoaiaren ikuspegia islatzen dutela. Adibidez, M lengoaiaren matrizeekin lan egiteko espezializatuta dagoenez erro karratua kalkulatzeko barne funtzioa, ***sqr***(***<DATUA>***), matrizeekin lan egin dezake.

ADIBIDEA

```
sqr ([4 9 16]) bueltatuko du [2 3 4]. Hau da, emandako bektorearen gai bakoitzaren erro karratua. Sarrera eta emaitzen arteko lotura posizioa da: lehenengo balioaren erro karratua lehen posizioan dagoen emaitza eta abar.
```

Erro karratua nola kalkulatu duen ez dakigu (eta ez zaigu axola), ezagutu beharrekoa da zer sarrerako argumentu eman behar zaion eta zer emaitza bueltatuko digun. Barne funtzio bat egikaritu ahal izateko bere izena jakin behar dugu eta bere sarrerako argumentuak zeintzuk diren eta zein ordenean eman behar zaizkion. Bueltatzen duen emaitza esleipenaren bitartez aldagai batean gorde dezakegu ondoren kalkulu gehiago egiteko. Barne funtzioak paketeetan multzokatuta izaten dira duten helburuaren arabera. Adibidez, bi dimentsioko grafikoak marrazteko paketea, karaktere kateekin lan egitekoa eta abar.

IRADOKIZUNA

MATLAB eta Octave GIBetako agindu leihuan ***help*** agindua egikaritzuz pakete guztiak zerrendatzen ditu. ***help <PAKETEARENIZENA>*** aginduak pakete horretako funtzio guztiak zerrendatzen ditu. ***help <FUNTZIOARENIZENA>*** agindua egikaritzuz barne funtzio horren sintaxia eta adibideak jasoko dituzu. ***doc*** agindua ***help*** agindua bezalakoa da baina informazio zabalagoa ematen du.

PROZESAKETA

Orain arte ikusitakoarekin datuak prozesatu ditzakegu MATLAB edo OCTAVEko agindu leihuetan. Adibidez, demagun 100, 88 eta 65 zenbakiak batez bestea kalkulatu nahi dugula. Hiru zenbaki horiek batu eta batura zati hiru egitearekin nahiko genuke. Ondorengo idatzi dezakegu agindu leihuan:

KODEA:

```
>> batura = 100 + 88 + 65
>> bb = batura / 3
```

Bi agindu hauekin *bb* aldagaian batez bestea kalkulatu genuke. Kalkulagailu baten modura erabiltzen ari gera GIBa. Prozesaketa sinplea denean (kalkulu eta datu gutxi) horrela egin dezakegu. Baina normalean prozesaketa konplexuagoa izaten da eta “kalkulagailuan” egin beharreko lanak zaildu egiten dira. Adibidez, demagun hilabete bateko euri jasan batez bestea kalkulatu nahi ditugula. Orain ez dira hiru datu baizik eta hogeita hamar. Hogeita hamar datu hauek banaka kopiatzea ez da oso eraginkorra (imaginatu urte osoko datuak direla). Seguruenik datuak fitxategi batean izango ditugu. Hauek agindu baten bitartez taula batean kopiatu eta ondoren prozesatuko genuke. Adibidez era honetan:

KODEA:

```
>> euria = load ('euria.dat', '-ascii')
>> batura = sum (euria)
>> bb = batura / length(euria)
```

load, *sum* eta *length* M lengoaiaren barne funtzioak dira. *load* funtzioak *euria.dat* fitxategiko hileko euri jasan datuak irakurri eta, esleipenaren bitartez, *euria* aldagaian gordetzen ditu. *sum* funtzioak aldiz bektoreko datu guztiak batzen ditu. Azkenik *length* funtzioak bektorearen luzera ematen du. Era honetan, hilabete osoko euri jasan batez bestea kalkulatzeko dugu.

Demagun orain asteburueta batez bestea kalkulatu nahi dugula. Prozesaketa konplexuagoa da, datu batzuk baztertu behar dira eta beste batzuk kontutan hartu. Prozesaketa baldintzatuta egongo da, batu soilik 6., 7., 13, 14, ... eguneko datuak. Gainera zenbat diren jakin beharko dugu batez bestearen zatiketa egiteko. Era honetako prozesaketa baldintzatu edo konplexua dugunean programa bat idatzi behar dugu prozesaketa kontrolatzen duten kontrol egitura bereziak erabiliz.

Demagun baita asteko egunez eguneko batez bestea nahi dugula ere. Gure kabuz kalkulatu dezakegu GIBa kalkulagailu baten modura erabilita, baina denbora asko beharko dugu. Gainera geroz eta konplexuagoa izan daiteke kalkulatu beharrekoa. Adibidez urte osoko datuen batez bestea astelehenetik ostirala arteko egunen batez bestea eta asteburuen batez bestea (larunbat eta igandeak elkarrekin). Gogoratu aurreko atalean esandakoa, datu asko edo egin eragiketa konplexuak direnean programa bat egitea komeni dela.

IDEI NAGUSIAK

- Datuak sinpleak (zenbakiak eta boolearrak) aldagaietan gordetzen dira esleipenaren bitartez (=).
- M lengoian matrizeak (taulak) era naturalean erabiltzen dira datu bezala.
- Eragile matematiko (+, -, /, *), erlazional (<, >, ==, <=, >=) eta logiko (&&, ||) sinpleak bi eragingai hartzen dituzte. Eragile negatiboa (-) eta ukapen logikoa (~) izan ezik.
- Eragile konplexuagoak programazio lengoaiak eskaintzen dituen funtzioak (azpi-programak) izaten dira.
- M lengoaiak barne funtzioak paketeetan multzokatzen ditu.
- GIBa kalkulagailu baten modura erabili dezakegu zenbait kalkulu egiteko. Datu asko, eragiketa konplexu edo prozesaketa baldintzatua dugunean programa bat egitea komeni da.

Bibliografia:

-Mijangos

-Ángel Franco Garcia, “MATLAB para el Grado en Ingeniería de Energías Renovables”, 2016. <http://www.sc.ehu.es/sbweb/fisica3/intro.html>

Erreferentziak:

[Harluxet] <http://www1.euskadi.net/harluxet/>

[R2]

ATRIBUTION:

Besteen irudiak

[Irudia:Geotermia] Eraldatutako irudia. Irudi Originala https://commons.wikimedia.org/wiki/File%3AVertical_heatpump_collector.svg

By K bentekik (Own work) [GFDL (<http://www.gnu.org/copyleft/fdl.html>) or CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0>)], via Wikimedia Commons