Azpiprogramen erabilera motibatzen

Algoritmoak eta programak ulergarriagoak izan daitezen azpiprogramak erabil daitezke. Ondoko orrietan erakutsi nahi dugu nola egin daitekeen programa bat bat ere azpiprogramarik erabili gabe, edo azpiprogramekin (funtzioekin edota prozedurekin).

Helburua hau da: algoritmo konplexuak oso erraz ulertzen diren algoritmo txikiekin osatzea.

Adibidez ariketa hauek egingo ditugu:

- *N* osoko zenbaki positibo bat emanda, zenbaki lehena denentz aztertuko duen algoritmoa espezifikatu eta idatzi. Lehen gisa hartuko dugu 1 zenbakia ere.
- Irakurri zenbaki oso positibo bat eta bera baino txikiagoak diren zenbaki lehenak idatzi.

Bukaeran ikusiko dugu zuzentzea ere, errazagoa egiten dela azpiprogramekin ondo antolatu den algoritmoa,

Lehena ote den aztertu (azpiprogramarik gabe)

Zehaztapena
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: "Lehena da" idatzi da N zenbaki lehena bada, bestela "Ez da lehena"

```
algoritmo Lehena_Da
hasiera
   Irakurri_Osokoa (N)
   Kontagailua := 0
   egin I guztietarako 1 tik N raino
      baldin  N mod I  = 0  orduan
         Kontagailua:= Kontagailua + 1
      ambaldin
   amguztietarako
   baldin  Kontagailua  <= 2  orduan
       Idatzi_Katea ("Lehena da")
   bestela
       Idatzi_Katea ("Ez da lehena")
   ambaldn
amaia.
```

Lehena ote den aztertu (Zenbat_Zatitzaile funtzioa erabiliz)

Aurreko ariketan zenbaki batek zenbat zatitzaile dituen ematen digun algoritmoa bere osotasunean erabili ahal izango bagenu? zelako pagotxa! Hau baino ez litzateke izango:

```
algoritmo Lehena_Da
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: "Lehena da" idatzi da N zenbaki lehena bada,
                bestela "Ez da lehena"

hasiera
   Irakurri_Osokoa (N)
   baldin  Zenbat_Zatitzaile(N) <= 2 orduan
      Idatzi_Katea ("Lehena da")
   bestela
      Idatzi_Katea ("Ez da lehena")
   ambaldin
amaia.
```

Hala ere, zatitzaileak kontatzeko algoritmoa egokitu beharko dugu horrela erabili ahal izateko. Algoritmoak ez du zertan irakurri behar zenbaki osoa, eta ez du idatzi behar zenbat diren zatitzaileak. Ez, modu bat asmatu behar da datu eta emaitzak algoritmoen artean elkar trukatzeko irakurri eta idazteko aginduak erabili beharrik gabe.

```
function Zenbat_zatitzaile (N: osokoa) itzuli osokoa
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: kontagailua = N zenbakiak zenbat zatitzaile dituen
hasiera
   Irakurri_Osokoa (N)
   Kontagailua := 0
   egin I guztietarako 1 tik N raino
      baldin  N mod I  = 0  orduan
         Kontagailua:= Kontagailua + 1
      ambaldin
   amguztietarako
   Idatzi_Osokoa( Kontagailua)
   itzuli( Kontagailua)
amaia
```

M baino txikiagoak diren zenbaki lehenak kalkulatu
(Lehena_Da funtzioa erabilita)

Beste aukera bat, `Lehena_Da` algoritmoa funtzio gisa definituta badugu:

```
algoritmo Lehenak
Aurrebaldintza: M (osokoa), M>0.
Postbalbintza: S (osoko-sekuentzia).
    Elementuak  zenbaki lehenak dira, M baino txikiagoak direnak
hasiera
   Irakurri_Osokoa (M)
   egin N guztietarako 1 tik M-1 raino
      baldin Lehena_Da (N) orduan
         Idatzi_Osokoa (N)
      ambaldin
   amguztietarako
amaia
```

Lehena_Da funtzioa:

```
funtzio Lehena_Da (N: datu Osokoa) itzuli Boolearra
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: itzuli da Egiazkoa N lehena bada,
               bestela Faltsua
hasiera
   Irakurri_Osokoa (N)I
   baldin  Zenbat_Zatitzaile(N) <= 2 orduan
      Idatzi_Katea ("Lehena_da")
      itzuli Egiazkoa
   bestela
      Idatzi_Katea ("Ez da lehena")
      itzuli Faltsua
   ambaldin
amaia

function Zenbat_zatitzaile (N: osokoa) itzuli osokoa
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: itzuli da N zenbakiak zenbat zatitzaile dituen
hasiera
   Irakurri_Osokoa (N)
   Kontagailua := 0
   egin I guztietarako 1 tik N raino
      baldin  N mod I  = 0  orduan
         Kontagailua:= Kontagailua + 1
      ambaldin
   amguztietarako
   Idatzi_Osokoa( Kontagailua)
   itzuli( Kontagailua)
amaia
```

M baino txikiagoak diren zenbaki lehenak kalkulatu
(azpiprogramarik gabe)

Aurreko ariketako algoritmoa erabiliz, *M* osoko positiboa baino txikiagoak diren zenbaki lehenak idatziko dituen algoritmoa espezifikatu eta idatzi.

```
algoritmo Lehenak
Aurrebaldintza: M (osokoa), M>0.
Postbalbintza: S (osoko-sekuentzia).
    Elementuak  zenbaki lehenak dira, M baino txikiagoak direnak
hasiera
   Irakurri_Osokoa (M)
   egin N guztietarako 1 tik M-1 raino
      Kontagailua := 0
      egin I guztietarako 1 tik N raino
         baldin  N mod I  = 0  orduan
            Kontagailua:= Kontagailua + 1
         ambaldin
      amguztietarako
      baldin  Kontagailua <= 2  orduan
         Idatzi_Osokoa (N)
      ambaldin
   amguztietarako
amaia
```

M baino txikiagoak diren zenbaki lehenak kalkulatu
(Lehena_Da prozedura erabilita)

Lehena_Da azpiprograma prozedura gisa ere definitu daiteke, horrela balio ezikagindu modura erabili beharko litzateke:

```
algoritmo Lehenak
Aurrebaldintza: M (osokoa), M>0.
Postbalbintza: S (osoko-sekuentzia).
    Elementuak  zenbaki lehenak dira, M baino txikiagoak direnak
hasiera
   Irakurri_Osokoa (M)
   egin N guztietarako 1 tik M-1 raino
      Lehena_Da (N,B)
      baldin  B  orduan
         Idatzi_Osokoa (N)
      ambaldin
   amguztietarako
amaia
```

Lehena_Da algoritmoa aldatu beharko genuke: A) Jaso behar duen datua ez da irakurriko sekuentzia batetik; aitzitik, erabili nahi izango denean adierazi beharko da, balioa zein den azalduz.. B) Itzuli behar duen emaitza ez da idatzi behar; *Lehena_Da* algoritmoa erabili ondoren nongo aldagaian utzi behar den zehaztu beharko da.

```
prozedura Lehena_Da ( N: datu Osokoa;
                      B: emaitza Boolearra)
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: B=Egiazkoa N lehena bada,
               bestela B=Faltsua
hasiera
   ~~Irakurri_Osokoa (N)~~I
   baldin  Zenbat_Zatitzaile(N) <= 2 orduan
      ~~Idatzi_Katea (“Lehena da”)~~
      B := Egiazkoa
   bestela
      ~~Idatzi_Katea (“Ez da lehena”)~~
      B := Faltsua
   ambaldin
amaia.

function Zenbat_zatitzaile (N: osokoa) itzuli osokoa
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: itzuli da N zenbakiak zenbat zatitzaile dituen
hasiera
   Kontagailua := 0
   egin I guztietarako 1 tik N raino
      baldin  N mod I  = 0  orduan
         Kontagailua:= Kontagailua + 1
      ambaldin
   amguztietarako
   itzuli( Kontagailua)
amaia
```

Lehena ote den aztertu
(Zenbat_Zatitzaile prozedura moduan erabiliz)

Edo beste aukera bat zatitzaileak kontatzeko algoritmoa, funtzio gisa ez, prozedura bezala erabilita:

```
algoritmo Lehena_Da
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: "Lehena da" idatzi da N zenbaki lehena bada,
                bestela "Ez da lehena"B (Boolearra).

hasiera
   Irakurri_Osokoa (N)
   Kontatu_Zatitzaileak( N, Zatitzaile_Kopurua)
   baldin  Zatitzaile_Kopurua <= 2  orduan
      Idatzi_Katea ("Lehena da")
   bestela
      Idatzi_Katea ("Ez da lehena")
   ambaldin
amaia.


algoritmo Kontatu_zatitzaileak (N: datu osokoa;
                                Zenbat: emaitza osokoa)
Aurrebaldintza: N (osokoa), N > 0.
Postbalbintza: itzuli da N zenbakiak zenbat zatitzaile dituen
hasiera
   Irakurri_Osokoa (N)
   Zenbat:= 0
   egin I guztietarako 1 tik N raino
      baldin  N mod I  = 0  orduan
         Zenbat:= Zenbat + 1
      ambaldin
   amguztietarako
   Idatzi_Osokoa( Zenbat)
   itzuli( Zenbat)
amaia
```

Azpiprogramen erabilera motibatzen

Zuzenketa errazagoa da azpiprogramak erabiliz

Ariketa osagarria:

Baina modu eraginkorrago batez defini daiteke `Lehena_Da` algoritmoa. Ez dira kontatu behar zatitzaile guztiak. Aski da aurkitzea zatitzaile bat; 1 edo zenbakia bera ez den zatitzailea:

```
algoritmo Lehena_Da
hasiera
   Irakurri_Osokoa (N)
   B := egiazkoa
   I :=  2
   bitartean  B  and I <= N/2  egin
      baldin  N mod I  = 0  orduan
         B := faltsua
      ambaldin
      I:= I + 1
   ambitartean
   Idatzi_Boolearra (B)
amaia
```

Errepikatzen segitzeko baldintzan, *(I<=N-1)* adierazpenaren ordez *(I <= N/2)* baldintza erabili da. Ordurako zatitzailerik aurkitu ez badugu hortik aurrera ere ez baita aurkituko.

Eta zer aldatu beharko zen **M baino txikiagoak diren zenbaki lehenak kalkulatzeko** gure orain arteko hiru aukera horietan? Zein bertsiotan egiten da errazago aldaketa?

Bertsio hobetua

M baino txikiagoak diren zenbaki lehenak kalkulatu

(azpiprogramarik gabe)

Bertsio trinkoan:

Zehaztapena

Aurrebaldintza: M (osokoa), M>0.

Postbalbintza: S (osoko-sekuentzia). Elementuak zenbaki lehenak dira eta M baino txikiagoak

```
algoritmo Lehenak
hasiera
   Irakurri_Osokoa (M)
   egin N guztietarako 1 tik M-1 raino
      Kontagailua := 0
      egin I guztietarako 1 tik N raino
         baldin N mod I = 0 orduan
            Kontagailua:= Kontagailua + 1
         ambaldin
      amguztietarako
      B := egiazkoa
      I := 2
      bitartean B and I <= N/2 egin
         baldin N mod I = 0 orduan
            B := faltsua
         ambaldin
         I:= I + 1
      ambitartean
      baldin B orduan
         Idatzi_Osokoa (N)
      ambaldin
   amguztietarako
amaia
```

Bertsio hobetua

M baino txikiagoak diren zenbaki lehenak kalkulatu

(Lehena_Da funtzio gisa definituta)

Lehena_Da funtzio gisa definituta badugu:

```
algoritmo Lehenak
hasiera
   Irakurri_Osokoa (M)
   egin N guztietarako 1 tik M-1 raino
      baldin Lehena_Da (N) orduan
         Idatzi_Osokoa (N)
      ambaldin
   amguztietarako
amaia
```

Lehena_Da funtzioa:

```
funtzio Lehena_Da (N: datu Osokoa) itzuli Boolearra
hasiera
   Kontagailua := 0
   egin  I guztietarako 2 tik N-1 raino
      baldin  N mod I  = 0  orduan
         Kontagailua:= Kontagailua + 1
      ambaldin
   amguztietarako
   baldin  Kontagailua >0  orduan
      B := faltsua
   bestela
      B := egiazkoa
   ambaldin
      B := egiazkoa
      I :=  2
      bitartean  B   and   I <= N/2  egin
         baldin  N mod I  = 0  orduan
            B := faltsua
         ambaldin
         I:= I + 1
      ambitartean
      itzuli B
amaia
```

Ikusi denez, bertsio trinkoan zailago izan daiteke egokitzapena. Ondo aztertu behar baitugu non egin behar den aldaketa. Bigarren bertsioan, berriz, argi dago Lehena_Da funtzioan bakarrik egin behar dela aldaketa, gainontzeko guztia berdin geratuko dela.

Azken oharra:

Hobekuntza gisa baldintza hau jarri dugu zatitzaileen bilaketan:

```
bitartean  B   and   I <= N/2  egin
```

Baina ez al dago lehenago bukatzerik?