

1 Adierazpenak ebaluatze ordena

a) $((\text{Not Eguakitsua}) \text{ or } \text{Euritsua})$

b) $((x > 4'0) \text{ eta } (y > 0,0))$

c) $-4'0 * (A^{**}2)$

d) $((\text{abs}(1+A)) \div B)$

e) $(A/B)^{*}C$

f) $A/(B^{*}C) \rightarrow$ aukera bakarra da

g) $(-4)^{*}(A^{**}(5+1)) \rightarrow$ aukera bakarra da

h) $(-4)^{*}(A^{**}(5+1))$

i) $(A/(B^{*}C)) \rightarrow$ ez dago beste aukerarik

j) $(A^{*}B/C)$

k) $((\text{abs}(x-y^{**}2)) > (2.0^{*}x/0'001)) \rightarrow$ Egia edo faltsua den esangia dugunaz kodearra izango da.

l) $((A/B)^{*}C) \rightarrow$ ez dago zalantzarik

m) $A+(B^{*}C)$

2. Algoritmoak diseinatu

2.2 triangeluaren azalera kalkulatu

Behatapena:

Aurre-baldintza: $F = \langle a, b \rangle$;

non;

- a den triangeluaren oinarria.
- b den triangeluaren altuera.
- a, b Erreal motakoak diren.
- $a, b > 0.0$

Post-baldintza: $G = \langle d \rangle$,

non d baita a oinarriko eta b altuerako triangeluaren azalera.

Proba-lasvak

Ez dago lasu orolorra besterik.

Algoritmoaren diseinua

Algoritmo Triangelu_Azalera - Idazleak
Oinarria Erreal, altuera Erreal

Hasiera:

Intrahiri_Erreal(F, oinarria)

Intrahiri_Erreal(F, altuera)

Azalera := oinarria * altuera / 2.0

La eragileak eragin eta emaitza azalera

Idatzi_Erreal(G, azalera)

Amaiera.

Korpo-errealia

$F = \langle a, b \rangle$

Oinarria
altuera
azalera

Korpo-errealia

$G = \langle d \rangle$

lehenengo N zenbaki arruntan batuz

Zehaztapena

Aurrebaldintza:

$$F = \langle a \rangle, \text{ non}$$

$$a = 0 \text{ edo}$$

$$a \geq 0$$

Post baldintza

$$G = \langle z \rangle, \text{ non}$$

$$z = 0 \text{ edo}$$

z bita lehenengo a zenbaki arruntan batuz

Proba klasua

$$a = 0 \Rightarrow G = \langle 0 \rangle$$

$$a = 1 \Rightarrow G = \langle 1 \rangle$$

$$a > 1 \Rightarrow G = \sum_{j=0}^a j$$

Algoritmoa goititik behara batuz.

Algoritmoa lehenengoetik buka, Batura 0 edo

Hasiere: irakurri osokoa (F, N)

Batura := N

Bitartean $N > 0$ egia

$$N := N - 1$$

Batura := ~~N~~ Batura + N

Amaituta

Algoritmoa: behetila gorra

Hasiere:

Idetzi osatzen N

Batua: 20

egin ~~II~~ ~~III~~ guztietarako n , tiki n reko

Batua := Batua + ~~III~~

an guztietarako

Idetzi osatzen $(G, Batua)$

amiciere

Algoritmoa: progresio aritmetikaren batua erabiltz

Hasiere:

Idetzi osatzen (F, N)

Idetzi osatzen $(G, (N+1) * N / 2)$

amiciere.

* erue!

erabiltzen osatzen!!!

Algoritmo

algoritmo A_letra_Kontab

Kont: Integer; -- 'A' letra zenbat aldiz agertu den.
Hasiera Kar: character; -- Onko karaktere

Kont := 0;

Irakurri_Karakterea (Kar)

Bitartean Kar /= 'a' egin

Baldin K = A orduan

Kont := Kont + 1

amabidatu

Irakurri_Karakterea (Kar);

amabidatu

Idatzi_Kontua (Kont)

osoa

amabidatu

Proba egitea ✓

Kodeketa (Programaren kodeketa ADA)

with irakurri_Karakterea, idatzi-osoa

procedure A_letra_Kontab is

Kont: Integer; -- 'A' letra zenbat aldiz agertu den

Kar: character; -- onko karaktere, azken irakurketa

egin

Kont := 0;

Irakurri_Karakterea (Kar);

while Kar /= 'a' loop

if Kar = 'A' then

Kont := Kont + 1;

end if;

Irakurri_Karakterea (Kar)

end loop

Idatzi-osoa (Kont);

end A_letra_Kontab;

Sellventziako elementuen tratemendua

Hasiara

Inklusio_karakterak (A)

Bitarteko karakterak ez badira azken orduan

inklusio horrengatik karakterak

ambitutea

anaiara

Ariketak

~~P~~ Puntuak azaltzen diren karakter-sellventziak emanda, kontatu
zenbat 'A' letra dauden

Espazitillazioa

Aurrebaldintza:

Sarrera. Estandarra: $\langle k_1, k_2, k_3, \dots, k_n \rangle$, non

- k_1, \dots, k_n : karakter

- $n \geq 1$

- $k_n \neq ' '$

Postbaldintza

Interna. estandarra: $\langle c \rangle$, non

c : integer

c izango baita sarrera. Estandarrak 'A' letra
kopurua

Prob. Usara

$S.E = \langle ' ' \rangle \rightsquigarrow T.E = \langle 0 \rangle$

$S.E = n > 0$ eta 'A' letarik ez dago $\rightsquigarrow T.E = \langle 0 \rangle$
 $n > 0$ eta 'A' letra egarri da

Algoritmos

Algoritmo: Secuencia Aritmética:

n, B_{inicio} integer;

Hesiera:

Inkurre: $O_{\text{sec}} = n$

$B_{\text{inicio}} := 0$ Kant

~~$B_{\text{inicio}} \leq N$ e egu~~

~~$B_{\text{inicio}} := B_{\text{inicio}} + 1$~~

$B_{\text{inicio}} := B_{\text{inicio}} + Kant$

$B_{\text{inicio}} := 0$

~~B_{inicio}~~

egun I guztietarako otile Wra

$B_{\text{inicio}} := B_{\text{inicio}} + 1$

an guztietarako

$I_{\text{dat}} := \text{osaketa}(B_{\text{inicio}})$

~~Hesiera~~

Begin

$B_{\text{inicio}} := \text{zabaki-osaketa} = 0$

$I_{\text{dat}} := \text{osaketa} : N$

While $N > 0$ loop;

$B_{\text{inicio}} := B_{\text{inicio}} + N;$

$N := N - 1;$

end loop;

$I_{\text{dat}} := \text{osaketa}(B_{\text{inicio}})$

end.

(15)

Lehenengo N zenbaki arrunten batura,

Ehaztapena:

Aurre-baldintza: $F \leq N$

Sarrera: N zenbaki ^{osoa} arrunta, non,
 $[1, \infty)$ dauden.

Postbaldintza:

$F \leq P$? zenbaki ^{osoa} arrunta izango da non,
1etik N ra dauden zenbaki arrunt guztiak batura
izango den.

Proba kasua

$N \rightarrow [1, \infty)$ denak ez dago proba kasu berezirik.

Algoritmoa

Algoritmoa: Segida-Aritmetikoa

N, P : integer;

Hasiera:

Iskorrri-arrunta: N

~~Baldin $n > 0$ egin~~

$P := (1+N) * N / 2$;

^{arruntak}
Idatzi-^{osoa} arrunta: P

Adea

with iskorrri-arrunta, idatzi-arrunta
procedure Segida-Aritmetikoa is

N, P : Integer;

begin

Iskorrri-Osokoa(n)

Baldin $n > 0$

A eta B-ren harteko zenbakirik txilliena
handiena eta ondoren txilliena.

B: hasierako

Idatzi lehenengo

1.2 Proba Kasuen diseinua

Hasiera: $A > B$ edo $A < B$

Buklata:

- $A > B$ denean, A hasierako A izango da eta B hasierako B, beraz A B idatzi.

- $A < B$ denean, A hasierako B izango da eta B hasierako A, beraz idatzi A B.

1.3 Algoritmoaren diseinua

Hasiera:

Irakurri - osolua: A

Irakurri - osolua: B

Baldin $A \geq B$ orduan,

Idatzi - osolua A

Idatzi - osolua B

Bestela

Trukatu (A, B)

Idatzi - osolua A

Idatzi - osolua B

am baldin

Auciera.

3. Lehenengo N zenbaki arrunten batura

$$N=6$$

$$\text{Batura} = 1+2+3+4+5+6 = 21$$

3.1 Zehaztapena

Aurrebaldintza

Sarrera: N zenbaki arrunta

Murriztapena: $[-\infty, 0]$ ez N berrua.

Postbaldintza:

Inteera: egin 1etik N ra
 dauken zenbaki arrunt gutxi batura. $(0, N) \rightarrow \sum_{j=1}^N j = 1+2+3+\dots+N$

3.2 Proba hasuen diseinua.

$N=1 \rightarrow$ idatzillo dugu 1

$N=2 \rightarrow$ idatzillo dugu 3

$$N \geq 1 \rightarrow \sum_{j=1}^N j$$

3.3 Algoritmoaren diseinua

Irakurri zenbaki arrunta N

Baturari 0 balioa eman

Kontagailuan 1 balioa eman

Bitartean Kontagailua $\neq N$

Baturari Kontagailua gehitu
 eta Kontagailuari gehitu bet
 am baldin

Idatzi osoko batura

Amaiera.

*

*

Baldin Elementu = Biletzella orden

Arkkituu: = Egiatzela

ambaldin

Posizio a: = Posizio + 1

Inkurrir - Osoc (elementu)

ambiterreen

Idetzi Baldin arkkituu z Egiatzela orden

Idetzi - osoc (posizio + 1)

bestela Idetzi - osoc (0)

ambaldin

Amerera

2. Elementuen baten bilaketa selbentzako ez-ordenatuen.

1. Zehaztapena

Aurrerabaldintza:

Sarrera-Estenderra $= \langle b, a_1, a_2, \dots, a_n \rangle$ non

• b eta a_i guztiek Integer

• $n \geq 1$ eta $a_n = 0$, eta a_1, \dots, a_{n-1} guztiek $\neq 0$

Postbaldintza:

Interea-Estandarra $= \langle p \rangle$, non

• p : Integer

• $p \geq 0$

• $1 \leq p < n$ eta $a_p = b$, eta a_1, \dots, a_{p-1} guztiek $\neq b$
(p den b -ren lehenengo agerperaren posizioa)

• $p = 0$, eta a_1, \dots, a_{n-1} guztiek $\neq b$

($p = 0$ den, selbentzaren b elementu agertzen ez delako)

Proba kasuak

Enuntziatzen agertzen diren kasuak.

$S_E = \langle 5, 0 \rangle \rightsquigarrow I_E = \langle 0 \rangle$
bilatu
nahidugua

Algoritmoa

Algoritmo: Elementu-baten bilaketa selbentzako.

Bilatzakoa: Elementua; Posizioa: Integer;

• Aurkitua: Boolean;

Hasiera: ...

Inkorrri-osloa: (Bilatzakoa)

Aurkitua: faltsua

Posizioa: 1

Inkorrri-osloa (Elementua

Bilatzakoa Elementua $\neq 0$ eta aurkitua = faltsua egia

*

Algoritmo Lehene - De

Hasiara

Irakurri-osoa(N)

bedin Zerkat-Zatizale(N) ≤ 2 orduan

Idatzi-kteta ("lehene de")

bestela,

Idatzi-kteta ("ez da lehene")

ambaldin.

Amaia

Funtzio Zerkat-Zatizale(N, osoa) itzuli oso

Hasiara

Kontagailua := 0

egin... Amino

bedin $N \bmod I = 0$ orduan

Kontagailua := Kontagailua + 1

ambaldin

amagaitzerako

Itzuli(Kontagailua)

Amaia.

~~Zatitzzeile Hopscotch~~

~~Algoritmus~~ ~~Zatitzzeile-Hopscotch~~ (

Algoritmus ~~lehene de~~

Hosszra

Inkorrni - asoc (F, N)

Zenbat - Zatitzzeile - kalkulatu (N, zenbat)

balidin zenbat ≤ 2 ordura

Idatziz - Uteke (G, lehene de)

B-stel

Idatziz - Uteke (G, & de lehene)

Prozedura zenbat - Zatitzzeile - kalkulatu (N: datu asoc;

Hosszra

Z: ~~datu~~ emaitza asoc)

Kontagailua := 0

egin I guztietarako ~~te~~ teile N-raia

balidin N mod I = 0 ordura

Kont: Kont + 1

amabaldin

amagutietarak

Z = Kont

onura

Aden idatzetako formula

Procedure idatz-prozedura (atet - formula) is

erazagupen-etak

begin

sententzia - form.

Azpiprogramak

Oinerrizko programazioa

B: mota:

Funtzioak: Emaitza bat, eta bakoitza, kalkulatzeko erabiltzen da, Azpiprogramak hainbat izango dituzte berak erabiliko dugu.

Adb

Pred erabiltzen da zenbaki oso baten aurreko kalkulatzeko.

$$N := \underbrace{\text{Pred}(8)}_{\substack{\text{zenbaki bat} \\ \text{de kasu honetan, 7}}} - 2$$

↑ parametroa

Funtzioa Izene (parametroa) itzuli mota

Parametro formula \rightarrow funtzioaren definizioa ezartzen diren eta mota definitu behar da.

Parametro erreala funtzioari beste programak bidezko deitzen diren erakusten da.

Alderian:

$$M := 2 \text{ zenb. 1}$$

$$N := 2 \text{ zenb. 2}$$

$$R := M \bmod N$$

bitartean $R \neq 0$ agian

$$M := N$$

$$N := R$$

$$R := M \bmod N$$

amaitzeko

itzuli N

amaitze

Zarb 1 etc Zarb 2 parametris formidli dire etc koner leuen
blize ez de abdetullo turizien zehar.

Prozedurek eliztar bat egitar duzu, irakurri, idetzi etc.
egitella. Gainera, koner bidet bako bat koner gahingo lortu
dituzgu. gaurdelle

Abb

Algoritmo Zurionet-Idetzi (Zurione-Uop : Idetzi Oso)

hasiera

egin kont gaurdelle 1 tik Zurione-Uop reiko

Idetzi-Uadltera (' ')

angardtelle

amaitz

Bi parametro erabili dituzte, baina gainera parametro horiek
amaitz gisa erabili ditzazke. Baina parametro horiek derrigorrez
alderantziz berak dire, bestelakoa ezinezkoa baretz amaitz
hori gaurdelle

Sarrera parametro bako bako \neq erabili ditzake.

Azpiprogramen beharrez

Bi modutan

Formulagor

Algoritmo gehitu

-- Aurrebaldintza: $kont = a$

Demagun kont alderantziz c balioa duela)

-- Postbaldintza: $kont = a + 1$

Informulagor

Algoritmo gehitu

-- Aurrebaldintza

-- Postbaldintza: $kont = a$ ^{hasiera} ~~er~~
balioari $+1$ egin

Funzioen eta prozeduren erabilera.

1. Aztertu F funtzioari eginiko deien zuzentasuna:

```
funtzio F (A, B : in Integer) return Boolean is
-- Aurre: A>0, B>0
-- Post: Emaitza = Egiazkoa baldin eta soilik baldin A > B2
Lag: Integer;
begin
    Lag := B**2;
    if A>Lag then
        return True;
    else
        return False;
    end if;
end F;
```

Deiak programa nagusitik:

```
...
Z, T: Integer;
...
Z:=10;
T:=3;
...
1) if F(10,3) then
2) F(Z+5,3);
3) if not F(Z,Z) then
4) while F(A,T) loop
5) while F(T,15)>1 loop
6) if F(A,B,Bool) then
7) if F(Lag,B) then
```

Handwritten notes:
if true then, else
if not false then
while loop
while loop
if true, then
if false, then
if true, then
if false, then

2. Aztertu P prozedurari eginiko deien zuzentasuna:

```
procedure P (A, B : in Integer; C: out Integer) is
-- Aurre: A>0, B>0
-- Post: C = A*B
begin
    C := A**2*B;
end P;
```

Deiak programa nagusitik:

```
...
Z, T, X: Integer;
...
Z:=10;
T:=3;
-- X hasieratu gabe
1) P(10,3,X);
2) P(X,T,Z);
3) P(T,T,X);
4) if P(T,Z,X)>200 then
5) P(X,T,10);
```

3. Aztertu P prozedurari eginiko deien zuzentasuna:

```
procedure P (D: in Integer; H: in out Integer) is
-- Aurre: H=a
-- Post: H = D**a
begin
    H := D**2*H;
end P;
```

Deiak programa nagusitik:

```
...
Z, T, X: Integer;
...
Z:=10;
T:=3;
-- X hasieratu gabe
1) P(10,X);
2) P(Z,10);
3) P(Z,T);
4) if P(T,Z)>200 then
```

