4. b)-alg1

Ideea algoritmului “alg1” este de a returna catul si restul impartirii a doua numere naturale nenule, de aici vine si linia de cod ce interschimba cele doua numere daca numarul pe care il impart este mai mic decat numarul la care impart. Algoritmul contorizeaza in variabila “c” catul impartirii, aceasta se incrementeaza la fiecare scadere, impartirea fiind o scadere repetata. Variabila “d” retine restul impartirii.

Initializari

PAS 6

PAS 5

PAS 4

PAS 3

PAS 2

PAS 1

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | d |
| 17 | 3 | 0 | 17 |
|  |  | 1 | 14 |
|  |  | 2 | 11 |
|  |  | 3 | 8 |
|  |  | 4 | 5 |
|  |  | 5 | 2 |
|  |  |  | STOP: d>b(F) |

17:3 = 5 rest 2

C = 5

D = 2

Insa algoritmul nu returneaza intodeauna un rezultat correct, acest lucru se poate observa daca intre datele de intrare se afla urmatoarea relatie: a = k \* b sau b = k \* a sau a=b unde k este un numar natural nenul. Aceasta relatie se refera la faptul unul dintre numere este un multiplu pentru celalalt numar, adica impartirea celor doua numere ar fi exacta. Pentru perechi de numere ca: (8,4), (10,5), (2,12), (6,48)

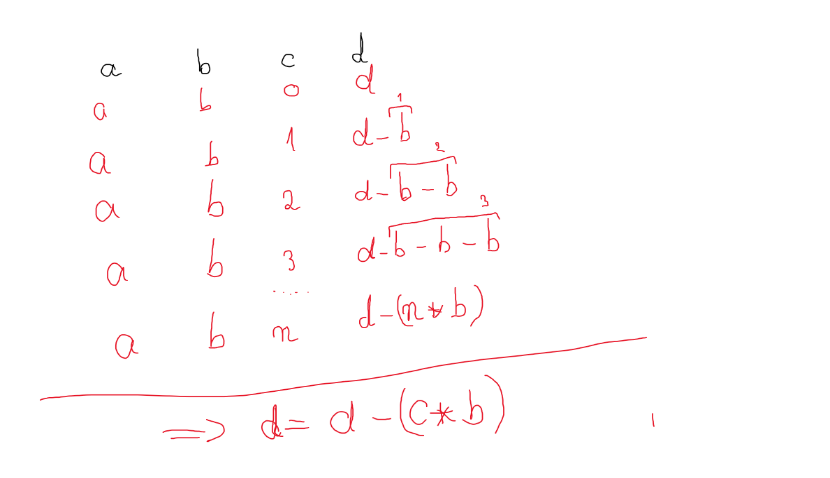
Sau orice pereche de forma (a,a), (b,b), ( (b/a)\*a, b ), ( a, (a/b)\*b ) pentru care impartirea lui “a” la “b” sau impartirea lui “b” la “a” este exacta.

Algoritmul ar fi returnat intodeauna rezultatul corect daca semnul **“>”** din linia de cod **“while d > b do”** ar fi fost schimbat in **“>=”**. In cazul semnului **“>”** algoritmul se va opri chiar daca mai trebuia efectuat inca un pas.

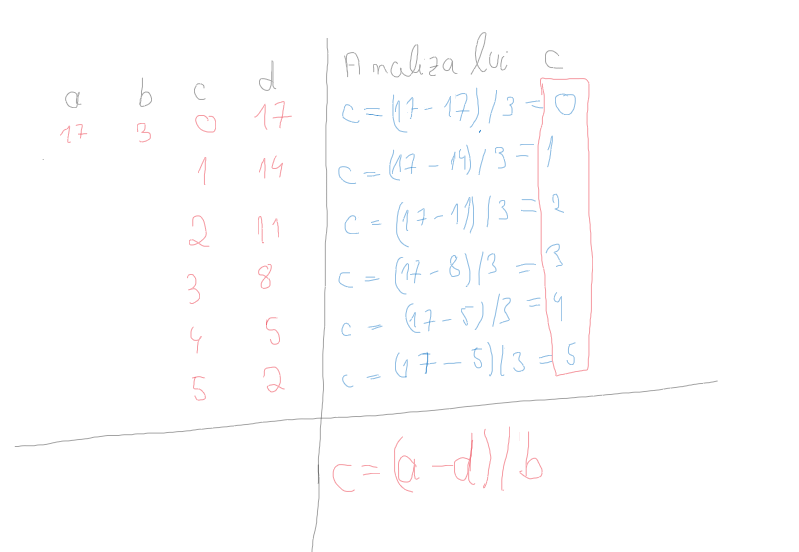
**[IDENTIFICAREA INVARIANTULUI]**

Pentru variabila “d” am ajuns la concluzia ca pentru orice valoare a lui “c” care respecta conditia c\*b < a(desi ca ar fi fost corect “c\*b<=a”, dar voi analiza cazul algoritmului pentru a arata ca e gresit) aceasta are valoarea: **d = a-c\*b.**

Atasez mai jos analiza pentru starea algorimului.



In cazul variabilei “c”, scopul acesteia este sa contorizeze de cate ori “intra” un numar in altul, pornind de aici putem observa urmatorul tabel:

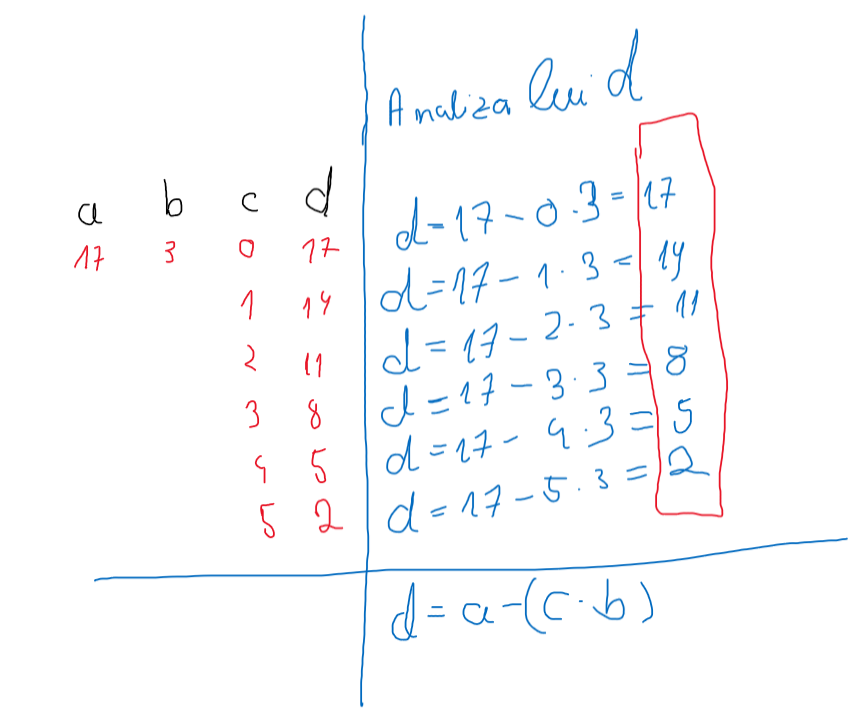


Invariantul pentru variabila “c” este: **c = (a – d) / b**

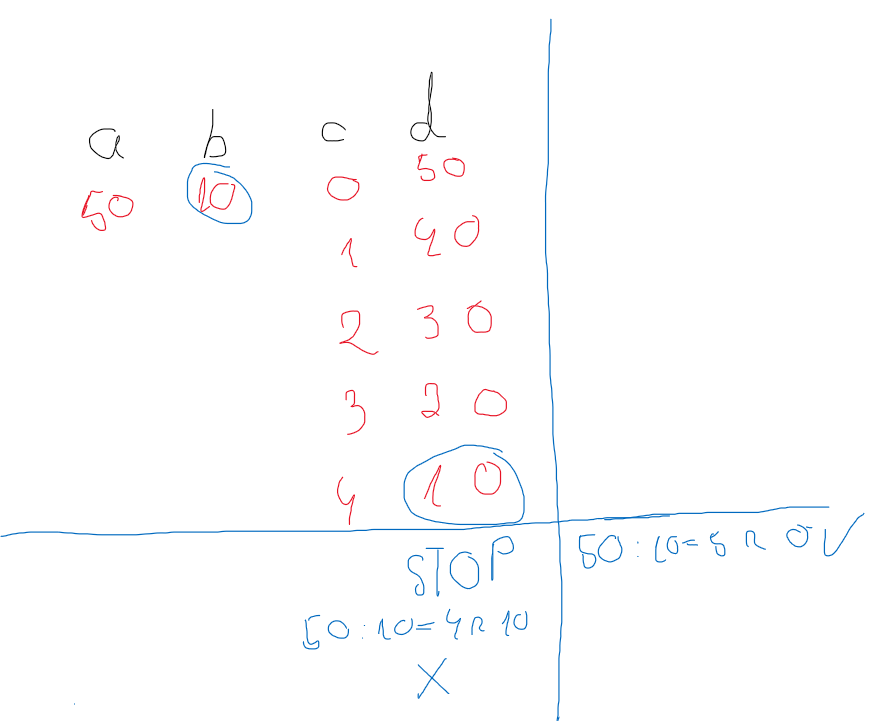
Invariantul1: I : { **d = a-c\*b}**

(i) Acesta este corect inainte de intrarea in ciclu deoarece variabila “c” va fi 0, iar produsul este 0 si ne mai ramane doar variabila “d”.

(ii)Este corect pe durata ciclului “while”, aceasta proprietate se conserva pe parcursul ciclului ”while”.



(iii) La iesirea din ciclu, invariantul se pastreaza intradevar. Variabila “d” nu mai este strict mai mare decat variabila “b”, iar conditia e falsa. Variabila “d” implica postconditia, adica aceasta retine restul impartirii numerelor 17 si 3.

(iii) OBS: Daca luam acelasi invariant pentru numerele 50 si 10 vom observa ca nu mai obtinem rezultatul correct:

**Impartirea lui 50 la 10 este o impartire exacta deci restul este 0**

**Daca aplicam invariantul pentru “d”, acesta va fi d = 50 – 4\*10 = 10 (rest)**

**Acest lucru nu este corect.**

Invariantul2: I : { **c = (a – d) / b}**

(i)Este correct inainte de intrarea in ciclu, deoarece (a-d) va fi 0, iar rezultatul impartirii va fi 0, deci c va fi 0.

(ii)Este correct pe parcursul ciclului dupa cum se poate vedea in starea algoritmului:

A white paper with writing on it

Description automatically generated with low confidence

(iii) La iesirea din ciclu, invariantul se pastreaza intradevar. Variabila “c” retine catul impartirii lui 17 la 3. Variabila “d” implica postconditia.

(iii) OBS: Daca luam acelasi invariant pentru numerele 50 si 10 vom observa ca nu mai obtinem rezultatul corect.

Diagram

Description automatically generated

**Impartirea lui 50 la 10 are catul 5**

**Daca aplicam invariantul pentru “c”, acesta va fi d = (50 – 40) / 4 = 10 (cat)**

**Acest lucru nu este corect.**

**[FINITUDINE]**

**Caz1: Algoritmul se termina dupa (a/b)–1 pasi in cazul in care intre datele de intrare se afla urmatoarea relatie: a = k \* b sau b = k \* a sau a=b unde k este un numar natural nenul. Aceasta relatie se refera la faptul unul dintre numere este un multiplu pentru celalalt numar, adica impartirea celor doua numere ar fi exacta.**

**Caz2:Algoritmul se termina dupa (a/b) pasi in cazul in care ambele numere sunt impare sau pare sau de paritati diferite**

**Caz1: = (a/b) - 1**

**Caz2: = (a/b)**

**[CONCLUZIE]**

**Algoritmul NU este corect**