

DOCUMENTAȚIE

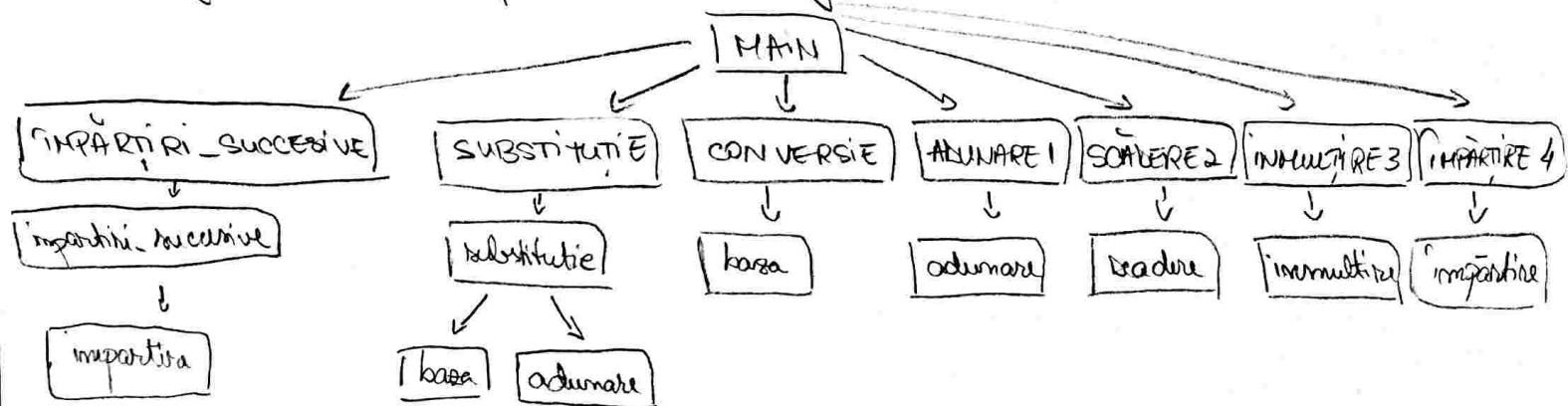
VINCZE Iulia-Paula
Grupa 214

* Enunț *

Creați o aplicație care realizează următoarele:

- conversie prin împărțiri succesive (din bază mai mare într-o bază mai mică)
- conversie prin substituție (din bază mai mică într-una mai mare)
- conversie utilizând o bază intermediară (bază 10)
- conversii rapide din bază 2 în bază 4, 8 sau 16 și respectiv invers
- adunarea a două numere date într-o bază oarecare
- scăderea a două numere date într-o bază oarecare
- înmulțirea unui nr. cu o cifră (care se citește de la tastatură) într-o bază oarecare
- împărțirea unui nr. la o cifră (care se citește de la tastatură) într-o bază oarecare

* Diagrama de apel a subalgoritmilor



* Tipuri de date folosite

INT → este un tip de date folosit pentru a reprezenta numere
întregi

CHAR → este un tip de date folosit pentru a reprezenta un caracter/
șiruri de caractere

→ un caracter este o unitate de folosire a informațiilor echivalente
cu o literă sau simbol alfabetic și este memorat prin codul
ASCII asociat caracterului

STACK → o stivă este o structură conceptuală constând dintr-un
set de elemente omogene și se bazează pe principiul ultimul
întrat, primul iese (LIFO)

→ este un tip de date abstracte folosit în mod obișnuit, cu două
operațiuni majore, și anume PUSH (adaugă element pe stivă) și
POP (scade un element din vârful stivei)

* Algoritmi principali

1. Pt. void baza : parametri : numrar, baza1, baza2

$nr10 \leftarrow 0, p \leftarrow 0, contor \leftarrow 1$

cât timp $contor \leq$ lungimea numărului

$c1 \leftarrow$ caracterul din număr de pe poziția lungimea - p

dacă $c1$ este literă

$cifra \leftarrow$ valoarea literii din baza 10

altfel

$cifra \leftarrow$ cifra din $c1$ // $c1$ fiind caracter, $cifra$ fiind int

$nr10 \leftarrow nr10 + cifra \cdot baza^p$

$contor \leftarrow contor + 1; p \leftarrow p + 1$

cât timp $nr10 \neq 0$

dacă $nr10 \% baza2 < 10$

facem push pe stivă la rest

altfel

facem push pe stivă la litera corespunzătoare restului
 $nr10 \leftarrow nr10 / baza2$

Date : numrar = numărul introdus care va
fi convertit
 $baza1$ = baza surse, $baza2$ = baza destinație
 $nr10$ = numărul în baza 10
 p = puterea a lui 10
 $contor$ = variabilă ghidătoare pt. a lua
caracter cu caracter până a modifica
numărul inițial

Precondiții :

$baza1, baza2$ să fie nr. naturale
numărul să fie natural

Postcondiții

rezultatul se află pe stivă, deci pt. a folosi
nr. trebuie extras de pe stivă

• Pt. void adunare, \rightarrow parametri: $nr1, nr2, baza, t, p$

- dacă $p \leq \text{lungimea lui } nr1$ sau $p \leq \text{lungimea lui } nr2$
 $c1 \leftarrow$ caracterul din $nr1$ de p poziția lungimii - p
 $c2 \leftarrow$ caracterul din $nr2$ de p poziția lungimii - p

- dacă $c1$ este litera
 $cifra1 \leftarrow$ valoarea literei în cifre
altfel
 $cifra1 \leftarrow$ cifra din $c1$

- dacă $c2$ este litera
 $cifra2 \leftarrow$ valoarea literei în variabilă int
altfel
 $cifra2 \leftarrow$ cifra din $c2$

$p \leftarrow p + 1$

- dacă $(cifra1 + cifra2 + t) \% baza < 10$
 facem push pe stivă la rest
altfel
 facem push pe stivă la litera corespunzătoare restului

$adunare(nr1, nr2, baza, (cifra1 + cifra2 + t) / baza, p)$ //apelăm funcția

altfel
 dacă $t \neq 0$
 facem push pe stivă la rest

Date: $nr1$ = primul nr care va fi folosit la adunare (în baza introdusă)
 $nr2$ = al doilea nr. care va fi folosit la adunare (în baza introdusă)
 $baza$ = baza în care s-au introdus numerele și în care va fi realizată suma
 t = transportul, în cazul în care suma a două cifre depășește baza introdusă
 p = variabilă ajutătoare pt. a lua caracter cu caracter fără a modifica nr. inițiale

$*(cifra1 + cifra2 + t) \% baza$ = suma cifrelor direct în baza introdusă

$*(cifra1 + cifra2 + t) / baza$ = transportul în cazul în care suma cifrelor depășește baza

Precondiții: $nr1, nr2$ să fie nr. naturale, $baza$ să fie nr. natural
la apelarea funcției $\left. \begin{array}{l} t_0 = 0 \\ p = 1 \end{array} \right\}$

Postcondiții: rezultatul (suma) în baza dată se află pe stivă, deci pentru a vizualiza suma trebuie extras numărul de pe stivă

Pt. void readur, - parametri: $m1, m2, baza, t, p$

dacă $p \leq$ lungimea lui $m2$

$c1 \leftarrow$ caracterul din $m1$ de pe poziția lungimii $-p$

$c2 \leftarrow$ caracterul din $m2$ de pe poziția lungimii $-p$

dacă $c1$ este litera

$cifra1 \leftarrow$ valoarea literei în număr

altfel

$cifra1 \leftarrow$ cifra din $c1$

dacă $c2$ este litera

$cifra2 \leftarrow$ valoarea literei în cifru înt

altfel

$cifra2 \leftarrow$ cifra din $c2$

$p \leftarrow p+1$

dacă $cifra1 \geq cifra2$

dacă $cifra1 - cifra2 - t < 10$

facem push pe stivă la dif cifrelor

altfel

facem push pe stivă la litera corespunzătoare diferenței

$scadur(m1, m2, baza, 0, p)$ // apelăm funcția fără transport

altfel

dacă $(baza + cifra1) - cifra2 - t < 10$

facem push pe stivă la dif cifrelor

altfel

facem push pe stivă la litera corespunzătoare diferenței

$scadur(m1, m2, baza, 1, p)$ // apelăm funcția cu transport (decarcă ne-am îngrumădat)

altfel

dacă $p \leq$ lungimea lui $m1$

introduc restul de cifre, ținând cont de transport

Date $m1 =$ descăzut, $m2 =$ mărșator

$baza =$ baza în care au fost introduse datele și în care se realizează diferența

$t =$ transportul (în cazul în care ne impurăm din mărșator)

$p =$ variabilă ajutătoare pt. a accesa caracter cu caracter fără a modifica numerele initiale

• Pt. void Inmultire, - parametri: $nr1, nr2, baza, t, p$

dacă $p \leq$ lungimea lui $nr1$

$c1 \leftarrow$ caracterul din $nr1$ de p poziția lungimii - p

dacă $c1$ este literă

$cifra1 \leftarrow$ valoarea literei în număr

altfel

$cifra1 \leftarrow$ cifra din $c1$

$p \leftarrow p + 1$

dacă $(cifra1 \cdot nr2 + t) \% baza < 10$

facem push pe stivă la rest

altfel

facem push pe stivă la litera corespunzătoare restului

$Inmultire(nr1, nr2, baza, (cifra1 \cdot nr2 + t) / baza, p)$

altfel

dacă $p >$ lungimea lui $nr1$ și $t \neq 0$

introduc transportul rămas după înmulțirea cifrelor lui $nr1$

Răspundere: $nr1, nr2, baza$ = numerele introduse

$nr2 = 0$ cifră

$t_0 = 0, p_0 = 1$

Postcondiție: rezultatul se află pe stivă, doar pentru a putea vizualiza produsul trebuie extras de pe stivă

Date: $nr1$ = primul număr care va fi înmulțit (în baza introdusă)

$nr2$ = cifra cu care înmulțim (în baza introdusă)

$baza$ = baza în care au fost introduse numerele și în care se calculează produsul

t = transportul, în cazul în care produsul depășește baza

p = variabila ajutătoare pt. a lua caracter cu caracter fără a modifica numărul inițial

• Pt. void Impartire - parametri : $m1, m2, baza, t, p, rest$

dacă $p < \text{lungimea lui } m1$

$c1 \leftarrow$ caracterul din $m1$ de poziția p

dacă $c1$ este literă

$cifra1 \leftarrow$ valoarea literei în cifră

altfel

$cifra1 \leftarrow$ cifra din $c1$

$p \leftarrow p + 1$

$\text{Impartire}(m1, m2, baza, (t * baza + cifra1) \% m2, p, rest)$

dacă $(t * baza + cifra1) / m2 < 10$

dacă $(t * baza + cifra1) / m2 \neq 0$

facem push pe stivă la $c1$

altfel

facem push pe stivă la litera compunătoare a celui

altfel

dacă $p = \text{lungimea lui } m1$ și $t \neq 0$

$rest = t + '0'$ // păstrează restul

Date : $m1 = \text{divizandul}$, $m2 = \text{divizor}$, $rest = \text{restul împărțirii}$

$baza =$ baza în care au fost introduse numerele în care se realizează împărțirea

$t =$ transportul, în cazul în care diferența nr. din timpul împărțirii este $\neq 0$

$p =$ variabila ajutoare

* $(t * baza + cifra1) / m2 =$ câtul direct în baza introdusă

* $(t * baza + cifra1) \% m2 =$ transportul (restul transportabil)

Precondiții : $m1, m2, baza = \text{nr. naturale}$

$m1 \geq m2$

pentru apelare $\begin{cases} t_0 = 0 \\ p_0 = 1 \end{cases}$

$m2 = 0$ cifră

Postcondiții : rezultatul se află pe stivă, deci pt. a vizualiza câtul, trebuie extras de pe stivă
restul împărțirii se transmite prin adresă, deci este salvat în variabila "rest"

• pt. void impartiri-succesive, folosesc funcția impartire(...)

cât timp $nr \neq 0$
il impart la baza2
salvez câtul
facem push pe stiva rezultat la rest

afișez rezultatul (adică modul nr.)

Date suplimentare: rezultat = 0 stivă locală

Precondiții: $nr1, baza1, baza2 = nr. naturale$
 $baza1 > baza2$

Postcondiții: -

• pt. void substitutie, folosesc funcțiile baza(...) și adunare(...)

cât timp $p \leq \text{lungimea numărului}$
realizez înmulțirea în baza 10 ($a_k \cdot baza1^k \quad \forall k \in [0, \text{lungimea nr.}]$)
* fac conversie a înmulțirii în baza destinație
salvez modul nr.
adun la nr1 produsul obținut
pun noua sumă în nr1
 $p < p+1$
afișez numărul

Precondiții: $nr, baza1, baza2 = nr. naturale$
 $baza1 < baza2$

Postcondiții: -

Obs: ceea ce e marcat cu * se putea realiza și cu funcția de înmulțire dintre cifra curentă și puterea bazei

* Cazuri de testare

• Pt. ÎMPĂRTIRI SUCCESIVE

Date in = 16, 12, 2 \Rightarrow Date out = Numărul este 10010

Date in = 2, 10010, 16 \Rightarrow Date out = Eroare! Căsa destinatie trebuie să fie mai mică decât casa sursă!

• Pt. SUBSTITUȚIE

Date in = 2, 10010, 16 \Rightarrow Date out = Numărul este 12

Date in = 16, 12, 2 \Rightarrow Date out = Eroare! Casa destinatie trebuie să fie mai mică decât casa sursă!

• Pt. BAZĂ INTERMEDIARĂ

Date in = 4, 12321, 13 \Rightarrow Date out = Numărul este 270

• Pt. ADUNARE

Date in = 10, 4, 6 \Rightarrow Date out = Suma este 10

• Pt. SCĂDERE

Date in = 10, 10, 3 \Rightarrow Date out = Diferența este 4

Date in = 10, 3, 9 \Rightarrow Date out = Eroare! Scăzătorul trebuie să fie mai mare decât scăzătorul!

• Pt. ÎNMULȚIRE

Date in = 10, 20, 5 \Rightarrow Date out = Produsul este 100

Date in = 10, 13, 56 \Rightarrow Date out = Eroare! Al doilea număr trebuie să fie doar o cifră!

• Pt. ÎMPĂRTIRE

Date in = 10, 5, 2 \Rightarrow Cățul este 2 rest 1

Date in = 10, 3, 5 \Rightarrow Eroare! Împărțitul trebuie să fie mai mare decât împărțitorul

Date in = 10, 45, 15 \Rightarrow Eroare! Al doilea număr trebuie să fie doar o cifră!