

Probleme 2:

Algorithm-time esc:

Debut

counter $\leftarrow 0$

size(m)

fillMat(m, mat, 0, 0)

genResult(mat, m, "result.txt", 0, counter)

showFile("result.txt", counter)

Fin

TDONT:

m: boolean 19×19 entries

TDQ:

counter: entier

showFile, size, fillMat, genResult: procedure

m: entier

Module Saisir:

procedure Size (@m: entier)

Debut

Si non ($4 < m < 19$) alors

Écrire ("entier")

Fin

size(m)

Fin

Module remplissage:

procedure fillMat(n : entier, @ mat: m, i: entier, j: entier)

Debut

Si $i < n$ et $j < n$ alors

num \leftarrow alea(2,99)

columnIndex $\leftarrow j$

LineIndex $\leftarrow i$

Si $j = n - 1$ alors

columnIndex $\leftarrow 0$

LineIndex $\leftarrow i + 1$

Si non
columnIndex $\leftarrow i + 1$

Fin Si

Si isPrime(num, 2) alors

mat[i, j] \leftarrow num

fillMat(n , mat, lineIndex, columnIndex)

Si non
fillMat(n , mat, i, j)

Fin Si

Fin

TDDL:

LineIndex, columnIndex: entier

num: entier

isPrime: fonction

Module est premier !

fonction isprime (n: entier, i: entier): boolean:

Debut

Si $n > 1$ alors

Si $i \leq n \text{ div } 2$ alors

Si $n \bmod i \neq 0$ alors

retourner isprime (n, i+1)

Sinon

retourner faux

fin Si

Sinon

retourner vrai

fin Si

Sinon

retourner faux

Fin Si

Fin

Module est Resultat !



procedure genResult (m : m, m: entier, src: chaîne, i: entier, @counter: entier):

Debut

Si $i < m$ alors

ouvrir (src, myFile, "a")

Si Long (isSorted (m, m, i, false)) > 0 alors

ecrivre_l (myFile, "<" + concat (i) + ">" + isSorted (m, m, i, false) + concat (m[i, m-1]))

counter ← counter + 1

Fin Si

Si Long (isSorted (m, m, i, true)) > 0 alors

ecrivre_l (myFile, ">" + concat (i) + ">" + isSorted (m, m, i, true) + concat (m [1, i]))

counter ← counter + 1

Fin Si

fermer (myFile)

genResult (m, m, src, i + 1, counter)

Fermer (myFile)

Fin

TADOL:

myFile: fichiers texte

isSorted: fonction



Module isSorted

fonction isSorted(mat: m, m: entier, L: entier, istime: boolean): chaîne de caractères ;

Valeur

$I \leftarrow 0$

checkTime \leftarrow Vrai

checkDec \leftarrow Vrai

theString \leftarrow ""

tant que $I < m-1$ faire

Si istime alors

First \leftarrow mat[L, I]

Then \leftarrow mat[L, I+1]

Sinon

First \leftarrow mat[I, L]

Then \leftarrow mat[I+1, L]

Fin Si

Si First > Then alors

checkDec \leftarrow faux

Sinon

checkTime \leftarrow faux

Fin Si

Si checkTime ou checkDec alors

theString \leftarrow theString + concat(First) + " "

Sinon

theString \leftarrow ""

Fin Si

$T \leftarrow F + 1$

Fin lorsque
retourner theString

Fin

TDOL:

check Iner, check Dec: boolean

First, Then: entier

theString: chaîne

δ : entier

Module affichage fichiers

Procédure ShowFile (sx: chaîne, size: entier):

Début

ouvrir (sx, myFile, "r")

pour i de 1 à size faire

écrire (lire ml (myFile))

fin pour

Fin

TD02:

myFile: fichierTexte

i: entier

