

\* DA Projet

Init

Obtenir a, c, m, X0

↓ a, c, m, X0

↓ isHulDobell

↓ isValide

if (isValide)

do while(nbStations ≤ sMax)

Traitement

nbStations++

↓ couts, nbStations

RechercheStationIdéale

↓ stationIdéale

else

sortir "La suite n'est pas valide"

Avant chaque traitement, X0 doit être réinitialisé  
 Pour chaque cas, on compare avec la même suite  
 pseudo-aléatoire.

\* Init

i = 0

valPrec = 0

do while (i < 5)

valPrec += (1,8^i\*EXP(-1,8))/FACT(i)

tabPoissonCumul[i] = valPrec

i++



François Donnay

Clément Vandendaelen

```
0-----0 ↓ a, c, m, X0
| isHullDobell |
0-----0 ↓ isValide
* isHullDobell

cond1 =
0-----0 ↓ a, b
| PGCD |
0-----0 ↓ PGCD==1

cond2 = (a - 1) % (c - 1) == 0

if (m % 4 == 0)
cond3 = ((a - 1) % 4 == 0);
else
cond3 = true

return (cond1 & cond 2 & cond3)

0-----0 ↓ a, b
| PGCD |
0-----0 ↓ PGCD
* PGCD
do while (b ≠ 0)
r = a % b
a = b
b = r

return a

0-----0
| GénérerNombreAléatoire |
0-----0
* GénérerNombreAléatoire
x1 = (a*x0+c)%m
x0=x1
```



Traitement

```

* Traitement
nbClientsOrdinairesDansFile = 0
nbClientsPrioritairesDansFile = 0

temps = 1
coutClientOrdinaire = 0
coutClientPrioritaire = 0
coutStationOccupée = 0
coutChangementTypeClient = 0

do while (temps ≤ tempsSim)
    Générer arrivées
    nbClientsOrdinairesDansFile += nbNouveauxClientsOrdinaires
    nbClientsPrioritairesDansFile += nbNouveauxClientsPrioritaires
    if ((nbStations == sMin || nbStations == 5 + sMin) AND temps ≤ 20)
        nbNouveauxClientsOrdinaires, nbClientsOrdinairesDansFile, nbClientsPrioritairesDansFile
        SortieInfosFilesEtStations
    do while (iStation ≤ nbStations)
        if (stations[iStation].duréeService == 0)
            if (iStation == 0)
                nbClientsPrioritaires--
            GénérerDuréeService
        stations[iStation].duréeService = duréeService - 1
        coutClientPrioritaire += 40/60
        coutStationOccupée += 35/60
    
```

on est le que  
station [2] est  
initialisé

Pas compris ! A justifier

de l'affichage  
A l'oral, pour améliorer  
adapter le code pour afficher  
des infos



```
else
  if (nbClientsOrdinaires > 0)
    nbClientsOrdinaires--
    | GénérerDuréeService |
    0 0 ↓ duréeService
    stations[iStation].duréeService = duréeService -1
    coutClientOrdinaire += 25/60
    coutStationOccupée += 35/60
  else
    if (nbClientsPrioritaires > 0)
      nbClientsPrioritaires--
      | GénérerDuréeService |
      0 0 ↓ duréeService
      stations[iStation].duréeService = duréeService -1
      coutClientPrioritaire += 40/60
      coutStationOccupée += 35/60
    coutStationInnoccupée += 20/60
  else
    coutStationOccupée += 35/60
    if (station[iStation].typeClient == ordinaire)
      coutClientOrdinaire += 25/60
    else
      coutClientPrioritaire += 40/60
    station[iStation].duréeService--
    coutClientPrioritaire += nbClientsPrioritaire * 40/60
    coutClientOrdinaire += nbClientOrdinaire * 25/60
    if ((nbStations == sMin || nbStations == 5 + sMin) AND temps ≤ 20)
```

*Cela ne prouve rien la preuve !*

*OK*

*A justifier !*



François Donnay

Clément Vandendaelen

nbNouveauxClientsOrdinaires, nbClientsOrdinairesDansFile, nbClientsPrioritairesDansFile  
 |  
 SortieInfosFilesEtStations

couts[nbStations - sMin] = coutClientPrioritaire + coutClientOrdinaire + coutStationOccupée + coutStationInnocupée  
 + coutChangementTypeClient  
 |  
 Sortie coutClientPrioritaire, coutClientOrdinaire, coutStationOccupée, coutStationInnocupée, couts[nbStations - sMin], coutChangementTypeClient

où est ce que couts[] est initialisé?



```

0 → nbClientsOrdinairesDansFile, nbClientsPrioritairesDansFile
| Générer arrivées |
0 → nbNouveauxClientsOrdinaires, nbNouveauxClientsPrioritaires
* Générer arrivées
nbArrivées = 0
nbNouveauxClientsOrdinaires = 0
nbNouveauxClientsPrioritaires = 0
i = 0

```

GénérerNombreAléatoire

```

do while ( i < 5 AND tabPoissonsCumul[i] < x0)
i++
nbArrivées = i

```

```

i = 0
do while (i < nbArrivées)

```

GénérerNombreAléatoire

```

if (x0 < 0.3)
if (nbClientsPrioritairesDansFile == 5)
nbNouveauxClientsOrdinaires++
coutChangementTypeClient += 50
else
nbNouveauxClientsPrioritaires++
else
nbNouveauxClientsOrdinaires++
i++

```

Sortir nbNouveauxClientsOrdinaires, nbNouveauxClientsPrioritaires

$\mu_0$

Poissons...  $\mu_0$  et par  $x_0$ . (même pour  $x_0$  ?)   
 Prio, changement } OK

Prouve idée, simplifie le code,   
 MAIS...  $\mu_0$  et par  $x_0$ . (même pour  $x_0$  ?)   
 générer arrivées



$\frac{0}{\text{DuréeService}} \mid \frac{0}{\text{DuréeService}}$   
 $\frac{0}{\text{DuréeService}} \mid \frac{0}{\text{DuréeService}}$   
 \* DuréeService  
 nbMinutes = 0

GénérerNombreAleatoire

```

if (x0 < 0.3051)
  nbMinutes = 1
else
  if (x0 < 0.6610)
    nbMinutes = 2
  else
    if (x0 < 0.9153)
      nbMinutes = 3
    else
      if (x0 < 0.9661)
        nbMinutes = 4
      else
        if (x0 < 0.9831)
          nbMinutes = 5
        else
          nbMinutes = 6
  
```

Sortir nbMinutes

✓ Ce ne peut pas être  $x_0$  que l'on compare  
 avec les  $x_0$  (qui sont compris entre 0 et 1)  
 Si non OK.

\*  $x_0, a, c, m$  en global. etc.



François Donnay

Clément Vandendaelen

0 — 0 ↓ couts, nbStations

station Idéale

\* Recherche Station Idéale

coutMin = HV

```
iRecherche = 0
```

```
do while(iRecherche < nbStations)
```

```
if (couts[iRecherche] < coutMin)
```

coutMin = couts[iRecherche]

stationIdéale = iRecherche + sMin

Sortir stationIdéale, coutMin

```

0-----0 ↓ stations, nbStations, nbNouveauxClientsPrioritaires, nbNouveauxClientsOrdinaires,
nbClientsOrdinairesDansFile, nbClientsPrioritairesDansFile

```

SortieInfosFilesEtStations

\* SortieInfosFileEstStations

Sortir nbNouveauxClientsOrdinaire + nbNouveauxClientsPrioritaire,

nbNouveauxClientsOrdinaire, nbNouveauxClientsPrioritaire

```
// Sortie nb arrivée + type (1)
```

Sortir nbClientsOrdinaireDansFile, nbClientsPrioritaireDansFile

// Sortie des infos nécessaire en debut de minute pour comprendre le fonctionnement(3)

istation = 0

```
do while (iStation < nbStations)
```

```
- if (stations[iStation].dureeService == 0) //sortie des info par station(2)
```

Sortir station[istation].typeClient

Sortir station[1station].duréeService