// Structures :

Client { statut, duréeAttente, duréeTraitement}

Stations { Clients } Liste de clients

File { Clients } Liste de clients

ClientsPerdus {Clients} Liste de clients

NouveauxClients { Clients } Liste de clients

clientsEjectes { Client } liste de clients

couts tableaux de cout

// Constantes :

COUT\_PRESENCE\_SYSTEME\_ORDI = 25 / 60

COUT\_PRESENCE\_SYSTEME\_RELATIF = 35 / 60

COUT\_PRESENCE\_SYSTEME\_ABSOLU = 45 / 60

COUT\_OCCUPATION\_ORDI = 30 / 60

COUT\_OCCUPATION\_PRIO = 32 / 60

COUT\_INOCCUPATION = 18 / 60

COUT\_PERTE\_CLIENT\_ORDI = 15

COUT\_PERTE\_CLIENT\_PRIO = 20

o──────────────────────o ↓ clients, nbClient, statut

│ contientClientStatut │

o──────────────────────o ↓ contientClient

┌─── \* contientClientStatut

│ iClient = 0

│ contientClient = False

│╔══ do while(iClient < nbClient)

│║┌── if(clients[iClient].statut == statut)

│║│ contientClient = True // Quitter via un return pour optimiser

│║└──

│╙──

└──────────

o───────────o ↓ ci, ri, longueurTableau

│ esperance │

o───────────o ↓ esperance

┌─── \* esperance

│ iTableau = 0

│ sommeCiRi = 0

│ sommeRi = 0

│╔══ do while(iTableau < longueurTableau)

│║ sommeCiRi += ci[iTableau] \* ri[iTableau]

│║ sommeRi =+ ri[iTableau]

│╙──

│ esperance = sommeCiRi / sommeRi

└──────────

o─────────────────────o ↓ listeClients, statut

│ dernierClientStatut │

o─────────────────────o ↓ index

// La liste est trier par statut client

┌─── \* dernierClientStatut

│ index = 0

│ nbClients = longueur de la file

│╔══ do while index < nbClients and listeClients[index].statut == statut:

│║ index += 1

│╙──

└──────────

o────────────o ↓ k,v

│ loiPoisson │

o────────────o ↓ probabilitéPoisson

┌─── \* loiPoisson

│ probabilitéPoisson = (e\*\*-v) \* (v\*\*k) / k!

└──────────

o──────────────────────o ↓ 𝜆, DS

│ nombreStationMinimum │

o──────────────────────o ↓ nbStationMin

┌─── \* nombreStationMinimum

│ nbStationMinimum = (𝜆 \* DS) // Arrondir au supérieur

└──────────

o─────────────────o ↓ a,c,m,x0

│ nombreAleatoire │

o─────────────────o ↓ nombreAleat

┌─── \* nombreAleatoire

│ nombreAleat = (x \* x0 + c) % m

└──────────

o───────────────────o ↓ a,c,m,x0,v

│ genererNbArrivées │

o───────────────────o ↓ k, x0

┌─── \* genererNbArrivées

│ o─────────────────o ↓ a, c, m, x0

│ │ nombreAleatoire │

│ o─────────────────o ↓ x1

│ x0 = x1

│ u1 = x1 / m

│ k = 0

│ o─────────────o ↓ k, v

│ │ loi\_poisson │

│ o─────────────o ↓ probabilité

│╔══ do while (u1 ≥ probabilité)

│║ k += 1

│║ o────────────o ↓ k,v

│║ │ loiPoisson │

│║ o────────────o ↓ probabilité

│╙──

└──────────

o─────────────────o ↓ a,c,m,x0

│ duréeTraitement │

o─────────────────o ↓ duréeTraitement, x1

┌─── \* duréeTraitement

│ o─────────────────o ↓ a,c,m,x0

│ │ nombreAleatoire │

│ o─────────────────o ↓ x1

│ u1 = x1/ m

│┌── if (u1 < 2/62)

││ duréeTraitement = 6

│├── if (u1 < 5/62)

││ duréeTraitement = 5

│├── if (u1 < 9/62)

││ duréeTraitement = 4

│├── if (u1 < 20/62)

││ duréeTraitement = 3

│├── if (u1 < 38/62)

││ duréeTraitement = 2

│├── else

││ duréeTraitement = 1

│└──

└──────────

↓ file, longueurFile, duréeTotaleClientOrdinaire, duréeTotaleClientAbsolu, duréeTotaleClientRelatif

o───────────────────o

│ gestionImpatience │

o───────────────────o

↓ file, longueurFile, duréeTotaleClientOrdinaire, duréeTotaleClientAbsolu, duréeTotaleClientRelatif, clientsPerdus

┌─── \* gestionImpatience

│ clientsPerdus = []

│ pos\_file = 0

│ iClientPerdu = 0

│╔══ do while (pos\_file < longueurFile)

│║ file[pos\_file].duree\_attente += 1

│║┌── if (file[pos\_file].statut == "ABSOLU")

│║│ duréeTotaleClientAbsolu += 1

│║├── else if file[pos\_file].statut == "RELATIF":

│║│ duréeTotaleClientRelatif += 1

│║├── else:

│║│ duréeTotaleClientOrdinaire += 1

│║└──

│║┌── if (file[pos\_file].duree\_attente ≥ 10 and pos\_file > 3)

│║│ clientsPerdus[iClientPerdu] = file[pos\_file]

│║│ iClientPerdu ++

│║└──

│║ pos\_file += 1

│╙──

│ longueurPerdu = iClientPerdu

│ iClientPerdu = 0

│╔══ do while (iClientPerdu < longueurPerdu)

│║ supprimer client de la file // Python

│╙──

└──────────

o─────────────────o ↓ vPrioritaire, vOrdinaire, a, c, m, x0

│ nouveauxClients │

o─────────────────o ↓ clients, x0

┌─── \* nouveauxClients

│ o───────────────────o ↓ a,c,m,x0,vOrdinaire

│ │ genererNbArrivées │

│ o───────────────────o ↓ nbOrdinaire, x0

│

│ o───────────────────o ↓ a,c,m,x0,vPrioritaire

│ │ genererNbArrivées │

│ o───────────────────o ↓ nbPrioritaire, x0

│ nbAbsolu = 0

│ nbRelatif = 0

│ iPrioritaire = 0

│╔══ do while (iPrioritaire < nbPrioritaire)

│║ nombreAléatoire = generer nombre aléatoire [0;1]

│║┌── if nombreAléatoire < 0.3)

│║│ nbAbsolu ++

│║├── else

│║│ nbRelatif ++

│║└──

│║ iPrioritaire++

│╙──

│ nbArrivées = nbOrdinaire + nbRelatif + nbAbsolu

│ iNouvClient = 0

│ iAbsolu = 0

│╔══ do while(iAbsolu < nbAbsolu)

│║ créer nouvClient // struct client

│║ nouvClient.duréeAttente = 0

│║ NouvClient.duréeTraitement = null

│║ nouvClient.Statut = "ABSOLU"

│║ nouveauxClients[iNouvClient] = nouvClient

│║ iAbsolu ++

│║ iNouvClient ++

│╙──

│ iRelatif = 0

│╔══ do while(iRelatif < nbRelatif)

│║ créer nouvClient

│║ nouvClient.duréeAttente = 0

│║ NouvClient.duréeTraitement = null

│║ nouvClient.Statut = "RELATIF"

│║ nouveauxClients[iNouvClient] = nouvClient

│║ iRelatif ++

│║ iNouvClient ++

│╙──

│ iOrdinaire = 0

│╔══ do while(iOrdinaire < nbOrdinaire)

│║ créer nouvClient

│║ nouvClient.duréeAttente = 0

│║ NouvClient.duréeTraitement = null

│║ nouvClient.Statut = "ORDINAIRE"

│║ nouveauxClients[iNouvClient] = nouvClient

│║ iOrdinaire ++

│║ iNouvClient ++

│╙──

└──────────

o─────────────o ↓ file, longueurFile, clients, longueurClient

│ gestionFile │

o─────────────o ↓ file, longueurFile

┌─── \* gestionFile

│ o──────────────────────o ↓ file, longueurFile, "ABSOLU"

│ │ contientClientStatut │

│ o──────────────────────o ↓ contientClientAbsolu

│

│┌── if (contientClientAbsolu)

││ o─────────────────────o ↓ file, "ABSOLU"

││ │ dernierClientStatut │

││ o─────────────────────o ↓ dernierAbsolu

│├── else:

││ dernierAbsolu = 0

│└──

│ o──────────────────────o ↓ file, longueurFile, "RELATIF"

│ │ contientClientStatut │

│ o──────────────────────o ↓ contientClientRelatif

│┌── if (contientClientRelatif)

││ o─────────────────────o ↓ file, "RELATIF"

││ │ dernierClientStatut │

││ o─────────────────────o ↓ dernierRelatif

│├── else

││┌── if(dernierAbsolu == 0)

│││ dernierRelatif = 0

││├── else

│││ dernierRelatif = dernierAbsolu + 1

││└──

│└──

│

│ iClient = 0

│╔══ do while(iClient < longueurClients)

│║┌── if(file[iFile].statut = "ABSOLU")

│║│ insérer client après le dernier client absolu

│║│ dernierAbsolu ++

│║│ dernierRelatif++

│║├── else if(file[iFile].statut = "RELATIF")

│║│ insérer client après le dernier client relatif

│║│ dernierRelatif++

│║├── else

│║│ insérer client à la fin de la file

│║└──

│╙──

└──────────

↓ stations, nbStation, file, longueurFile, a, c, m, x0

o───────────────────────────o

│ gestionClientsPrioritaire │

o───────────────────────────o ↓ file, longueurFile

┌─── \* gestionClientsPrioritaire

│ Déclarer tableau clientsEjectes

│ iClient = 0

│ nbClientEjecté = 0

│╔══ do while (iClient < longueurFile) and file[iClient].statut == "ABSOLU")

│║ tempsTraitementMax = LV

│║ nbStationMax = LV

│║ iStation = 0

│║

│║╔══ do while(iStation < nbStation)

│║║┌── if(stations[iStation].statut == null) // On accéde au statut du client en station, cas station vide

│║║│ numStationMax = iStation

│║║│ longueurFile --

│║║├── else if(stations[iStation].statut == "ORDINAIRE" and station[iStation].duree\_traitement > temps\_traitement\_max) // Cas client ordinaire en station

│║║│ numStationMax = iStation

│║║│ temps\_traitement\_max = station[iStation].dureeTraitement

│║║└──

│║║ iStation++

│║╙──

│║

│║┌── if(numStationMax ≠ -1)

│║│┌── if(stations[numStationMax].statut ≠ null)

│║││ clientsEjectes[iClientEject] = station[numStationMax]

│║││ nbClientEjecté ++

│║│└──

│║│ stations[numStationMax] = file[iClient]

│║│ o─────────────────o ↓ a,c,m,x0

│║│ │ duréeTraitement │

│║│ o─────────────────o ↓ duréeTraitement, x0

│║│ station[numStationMax].dureeTraitement = duréeTraitement

│║└──

│║

│║ iClient ++

│╙──

│

│ iClientEjecté = 0

│╔══ do while (iClientEjecté < nbClientEjecté)

│║ clientsEjectés[iClientEjecté].statut = "ABSOLU" // Python - Orienté objet → Les clients dans la file seront aussi affectés

│║ Insérer clientsEjectés[iClientEjecté] en début de file

│║ iClientEjecté ++

│╙──

└──────────

↓ vPrioritaire, vOrdinaire, a, c, m, x0, nbStationsMin, nbStationsMax, tempsSimulation

o───────────────────────o

│ simulationFileAttente │

o───────────────────────o ↓ nbStationsOptimal

┌─── \* simulationFileAttente

│ coutMin = HV

│ déclarer tableau couts

│ nbStationOptimal = nbStationMin

│ nbStation = nbStationMin

│

│╔══ do while(nbStation < nbStationMax)

│║ dureeTotaleClientAbsolu = 0

│║ dureeTotaleClientOrdinaire = 0

│║ dureeTotaleClientRelatif = 0

│║ dureeOccupationPrio = 0

│║ dureeOccupationOrdi = 0

│║ dureeInoccupation = 0

│║ clientsPrioPerdus = 0

│║ clientsOrdinairesPerdus = 0

│║ o───────────────────o ↓ stations, nbStation

│║ │ initStationsVides │

│║ o───────────────────o ↓ stations

│║ o──────────────o

│║ │ initFileVide │

│║ o──────────────o ↓ file, longueurFile

│║

│║╔══ do while(temps < tempsSimulation)

│║║↓ file, longueurFile, duréeTotaleClientOrdinaire, duréeTotaleClientAbsolu, │║║duréeTotaleClientRelatif

│║║ o───────────────────o

│║║ │ gestionImpatience │

│║║ o───────────────────o ↓ file, longueurFile, duréeTotaleClientOrdinaire, │║║duréeTotaleClientAbsolu, duréeTotaleClientRelatif, clientsPerdus

│║║

│║║ iClientPerdu = 0

│║║ nbClientPerdus = longueur clientsPerdus

│║║╔══ do while (iClientPerdu < nbClientPerdus)

│║║║┌── if(clientsPerdus[iClientPerdu].statut == "ORDINAIRE")

│║║║│ clientsOrdinairePerdu ++

│║║║├── else

│║║║│ clientPrioPerdus ++

│║║║└──

│║║╙──

│║║ o─────────────────o ↓ vPrioritaire, vOrdinaire, a, c, m, x0

│║║ │ nouveauxClients │

│║║ o─────────────────o ↓ clients, x0

│║║ longueurFile = taille file

│║║ longueurClient = taille clients

│║║ o─────────────o ↓ file, longueurFile, clients, longueurClient

│║║ │ gestionFile │

│║║ o─────────────o ↓ file, longueurFile

│║║ o───────────────────────────o ↓ stations, nbStation, file, longueurFile, │║║a, c, m, x0

│║║ │ gestionClientsPrioritaire │

│║║ o───────────────────────────o ↓ file, longueurFile

│║║

│║║ iStation = 0

│║║╔══ do while(iStation < nbStation)

│║║║┌── if(stations[iStation].statut == null)

│║║║│┌── if(longueurFile ≠ 0)

│║║║││ o─────────────────o ↓ a,c,m,x0

│║║║││ │ duréeTraitement │

│║║║││ o─────────────────o ↓ duréeTraitement, x1

│║║║││ stations[iStation] = file[0]

│║║║││ stations[iStation].dureeTraitement = dureeTraitement -1

│║║║│├── else

│║║║││ dureeInoccupation ++

│║║║│└──

│║║║├── else

│║║║│ station[iStation].dureeTraitement --

│║║║└──

│║║║┌── if(station[iStation].statut == "ABSOLU")

│║║║│ dureeTotaleClientAbsolu ++

│║║║│ dureeOccupationPrio++

│║║║├── else if(station[iStation].statut = "RELATIF")

│║║║│ dureeTotaleClientRelatif++

│║║║│ dureeOccupationPrio++

│║║║├── else

│║║║│ dureeTotaleClientOrdinaire ++

│║║║│ dureeOccupationOrdi++

│║║║└──

│║║║┌── if(station[iStation].dureeTraitement == 0)

│║║║│ station[istation].statut = null

│║║║└──

│║║║ iStation ++

│║║╙──

│║║ temps++

│║╙──

│║ couts[nbStation - nbStationMin] = (COUT\_INOCCUPATION \* dureeInoccupation) │║ +

│║ (COUT\_PERTE\_CLIENT\_ORDI \* clientsOrdinairesPerdus) +

│║ (COUT\_PERTE\_CLIENT\_PRIO \* clientsPrioPerdus) +

│║ (COUT\_OCCUPATION\_PRIO \* dureeOccupationPrio) +

│║ (COUT\_OCCUPATION\_ORDI \* dureeOccupationOrdi) +

│║ (COUT\_PRESENCE\_SYSTEME\_ABSOLU \* dureeTotaleClientAbsolu) +

│║ (COUT\_PRESENCE\_SYSTEME\_RELATIF \* dureeTotale\_clientRelatif) +

│║ (COUT\_PRESENCE\_SYSTEME\_ORDI \* dureeTotaleClientOrdinaire))

│║┌── if(coutMin > couts[nbStation - nbStationMin])

│║│ coutMin = couts[nbStation - nbStationMin]

│║│ nbStationOptimal = nbStation

│║└──

│║

│║ nbStation ++

│╙──

└──────────