class Film:

*"""*

*Initialiseerd Film object*

*:param: self, id, titel, rating*

*:return: Film object*

*"""*

def \_\_init\_\_(self, id, titel, rating):

*pass*

*"""*

*Verwijdert Film object instansie*

*:param: film objects*

*:return: string*

*"""*

def \_\_del\_\_(self):

*pass*

def create():

*""""creert een map*

*Args:*

*Returns:*

*Empty map"""*

*pass*

def destroy():

*""""verwijderd een map*

*Args:*

*Returns:*

*void (removes map)"""*

*pass*

def is\_empty(films):

*""""Bepaalt of films leeg is*

*Args:*

*films*

*Returns:*

*bool true or false"""*

*pass*

def get\_length(films):

*""""Geeft de lengte van films*

*Args:*

*films*

*Returns:*

*int (>= 0)*

*"""*

def insert(id, lijst):

*""""Voegt een film toe aan de map*

*Args:*

*id (hash, int>0)*

*lijst ([titel, rating])*

*Returns:*

*films met film toegevoegd*

*"""*

*pass*

def delete(id):

*""""verwijderd de film uit films*

*Args:*

*id (hash, int>0)*

*Returns:*

*films met film verwijderd"""*

*pass*

def retrieve(id):

*""""Haalt film op met de unieke ID en geeft de lijst terug*

*Args:*

*id (hash, int>0)*

*Returns:*

*films[id]:*

*lijst ([titel, rating])*

*"""*

*pass*

*import* LinkedList

class Gebruiker:

*"""*

*Initializeerd Gebruiker object*

*:param: self, id, voornaam, achternaam, emailadress*

*:return: Gebruiker object*

*"""*

def \_\_init\_\_(self, id, voornaam, achternaam, emailadress):

*pass*

*"""*

*Verwijdert een instansie van Gebruiker*

*:param: self*

*:return: string*

*"""*

def \_\_del\_\_(self):

*pass*

def create():

*""""creert een map*

*Args:*

*Returns:*

*Empty map"""*

*pass*

def destroy():

*""""verwijderd een map*

*Args:*

*Returns:*

*void (removes map)"""*

*pass*

def is\_empty(gebruikers):

*""""Bepaalt of gebruikers leeg is*

*Args:*

*gebruikers*

*Returns:*

*bool true or false"""*

*pass*

def get\_length(gebruikers):

*""""Geeft de lengte van gebruikers*

*Args:*

*gebruikers*

*Returns:*

*int (>= 0)*

*"""*

def insert(id, lijst):

*""""Voegt een gebruiker toe aan gebruikers*

*Args:*

*id (hash, int>0)*

*lijst ([voornaam, achternaam, email])*

*Returns:*

*gebruikers met gebruiker toegevoegd*

*"""*

*pass*

def delete(id):

*""""verwijderd de gebruiker uit gebruikers*

*Args:*

*id (hash, int>0)*

*Returns:*

*gebruikers met gebruiker verwijderd"""*

*pass*

def retrieve(id):

*""""Haalt gebruiker op met de unieke ID en geeft de lijst terug*

*Args:*

*id (hash, int>0)*

*Returns:*

*gebruikers[id]:*

*lijst ([voornaam, achternaam, email])*

*"""*

*pass*

class Reservatie:

*"""*

*Initializeerd een Reservatie Object*

*:param: self, id, userid, timestamp, vertoningid, plaatsen*

*:return: reservatie object*

*"""*

def \_\_init\_\_(self, id, userid, timestamp, vertoningid, plaatsen):

*pass*

def create():

*""""creert een map*

*Args:*

*Returns:*

*Empty map"""*

*pass*

def destroy():

*""""verwijderd een map*

*Args:*

*Returns:*

*void (removes map)"""*

*pass*

def is\_empty(reservaties):

*""""Bepaalt of een map leeg is, de voorstelling kan begginnen if true*

*Args:*

*reservaties*

*Returns:*

*bool true or false"""*

*pass*

def get\_length(reservaties):

*""""Geeft de lengte van reservaties*

*Args:*

*reservaties*

*Returns:*

*int (>= 0)*

*"""*

def insert(id, lijst):

*""""Voegt een reservatie toe aan reservaties*

*Args:*

*id (hash, int>0)*

*lijst ([userid, timestamp, vertoning, plaatsen])*

*Returns:*

*reservaties met reservatie toegevoegd*

*"""*

*pass*

def delete(id):

*""""verwijderd de reservatie uit reservaties (als de gast is aangekomen in kinepolis ==> vertoning starten if empty)*

*Args:*

*id (hash, int>0)*

*Returns:*

*reservaties met reservatie verwijderd"""*

*pass*

def retrieve(id):

*""""Haalt reservate op met de unieke ID en geeft de lijst terug*

*Args:*

*id (hash, int>0)*

*Returns:*

*reservaties[id]:*

*lijst ([userid, timestamp, vertoning, plaatsen])*

*"""*

*pass*

*import* LinkedList

*import* Gebruiker

*import* Film

*import* Tree

*import* Vertoning

*import* Queue

*import* Reservatie

*import* time

*import* Zaal

class ReservatieSysteem:

*"""*

*initializeerd object ReservatieSysteem*

*:param: self, zaal1, zaal2*

*:return: object reservatiesysteem*

*"""*

def \_\_init\_\_(self, zaal1, zaal2):

*pass*

*"""*

*maakt een gebruiker object en zet ze in de linkedlist van gebruikers*

*:param: self, id, voornaam, achternaam, emailadress*

*:return: gebruiker object*

*"""*

def maak\_gebruiker(self, id, voornaam, achternaam, emailadress):

*pass*

*"""*

*print alle gebruikers in de linkedlist van gebruiekrs*

*:param: self*

*:return: string*

*"""*

def toon\_gebruikers(self):

*pass*

*"""*

*maakt een film object en zet ze in de linkedlist van films*

*:param: self, id, titel, rating*

*:return: film objects*

*"""*

def maak\_film(self, id, titel, rating):

*pass*

*"""*

*print alle films in de linkedlist van films*

*:param: self*

*:return: string*

*"""*

def toon\_films(self):

*pass*

*"""*

*maakt een vertoning object en zet ze in de BST van vertoningen*

*:param: self, id, zaalnummer, slot, datum, filmid*

*:return: vertoning object*

*"""*

def maak\_vertoning(self, id, zaalnummer, slot, datum, filmid):

*pass*

*"""*

*print alle vertoningen in de BST van vertoningen*

*:param: self*

*:return: string*

*"""*

def toon\_vertoningen(self):

*pass*

*"""*

*Verhoogt de tijd met X units*

*:param: self, x*

*:return: string*

*"""*

def increment\_time(self, x):

*pass*

*"""*

*maakt een reservatie object en zet ze in de queue van reservaties*

*:param: self, id, userid, timestamp, vertoningid, plaatsen*

*:return: reservatie object*

*"""*

def maak\_reservatie(self, id, userid, timestamp, vertoningid, plaatsen):

*pass*

*"""*

*print alle reservaties in de queue van reservaties*

*:param: self*

*:return: string*

*"""*

def toon\_reservaties(self):

*pass*

*"""*

*verlaagt het aantal vrije plaatsen in een vertoningen adhv een reservatie*

*:param: reservatie object*

*:return: void*

*"""*

def verlaag\_plaatsen(self, reservatie):

*pass*

*"""*

*print de vrije plaatsen in een vertoning*

*:param: vertoning*

*:return: string*

*"""*

def printify(self, vertoning):

*pass*

class Vertoning:

*"""*

*Initializzerd een object Vertoning*

*:param: self, id, zaalnummer, slot, datum, filmid, vrijeplaatsen:*

*:return: Verotoning object*

*"""*

def \_\_init\_\_(self, id, zaalnummer, slot, datum, filmid, vrijeplaatsen):

*pass*

*"""*

*verwijdert een instansie vertoning*

*:param: self*

*:return: string*

*"""*

def \_\_del\_\_(self):

*pass*

def create():

*""""creert een map*

*Args:*

*Returns:*

*Empty map"""*

*pass*

def destroy():

*""""verwijderd een map*

*Args:*

*Returns:*

*void (removes map)"""*

*pass*

def is\_empty(vertoningen):

*""""Bepaalt of vertoningen leeg is*

*Args:*

*vertoningen*

*Returns:*

*bool true or false"""*

*pass*

def get\_length(vertoningen):

*""""Geeft de lengte van vertoningen*

*Args:*

*vertoningen*

*Returns:*

*int (>= 0)*

*"""*

def insert(id, lijst):

*""""Voegt een vertoning toe aan de map*

*Args:*

*id (hash)*

*lijst ([zaalnummer, slot, datum, filmid, vrije plaatsen])*

*Returns:*

*vertoningen met vertoning toegevoegd*

*"""*

*pass*

def delete(id):

*""""verwijderd de vertoning als ze voorbij is: tijd nu < datum*

*Args:*

*id (hash, int>0)*

*Returns:*

*vertoning met vertoning verwijderd"""*

*pass*

def retrieve(id):

*""""Haalt vertoning op met de unieke ID en geeft de lijst terug*

*Args:*

*id (hash, int>0)*

*Returns:*

*vertoningen[id]:*

*lijst ([zaalnummer, slot, datum, filmid, vrije plaatsen])*

*"""*

*pass*

class Zaal:

*"""*

*Initializeerd een zaal object*

*:param: self, nummer, plaatsen*

*:return: zaal object*

*"""*

def \_\_init\_\_(self, nummer, plaatsen):

*pass*

def create\_list():

*""""Creert een lijst*

*Args:*

*Returns:*

*Empty list*

*"""*

*pass*

def destroy\_list():

*""""Verwijdert lijst*

*Args:*

*Returns:*

*void (removes list)*

*"""*

*pass*

def is\_empty(zalen):

*""""Bepaalt of zalen leeg is*

*Args:*

*zalen*

*Returns:*

*bool true or false"""*

*pass*

def get\_length(zalen):

*""""Geeft de lengte van zalen*

*Args:*

*zalen*

*Returns:*

*int (>= 0)*

*"""*

def insert(zaalnummer, zitplaatsen):

*""""Voegt een zaal toe aan de zalen*

*Args:*

*zaalnummer (>= 0 && <=len(zalen))*

*zitplaatsen (>0)*

*Returns:*

*zalen met de zaal toegevoegd (plaatst zitplaatsen op idx zaalnummer+1 in de zalen)*

*"""*

*pass*

def delete(zaalnummer):

*""""Verwijdert een zaal uit zalen*

*Args:*

*zaalnummer (>= 0 && <=len(zalen))*

*Returns:*

*zalen met de zaal verwijderd (verwijdert zaal uit zalen op idx zaalnummer+1)*

*"""*

*pass*

def retrieve(zaalnummer):

*""""Haalt een zaal op met zaalnummer en geeft aantal zitplaatsen terug*

*Belangrijkste functie want niet vaak dat er een nieuwe zaal bijkomt of weggaat*

*Args:*

*zaalnummer (>= 0 && <=len(zalen))*

*Returns:*

*zalen[zaalnummer+1] (>0)*

*"""*

*pass*