





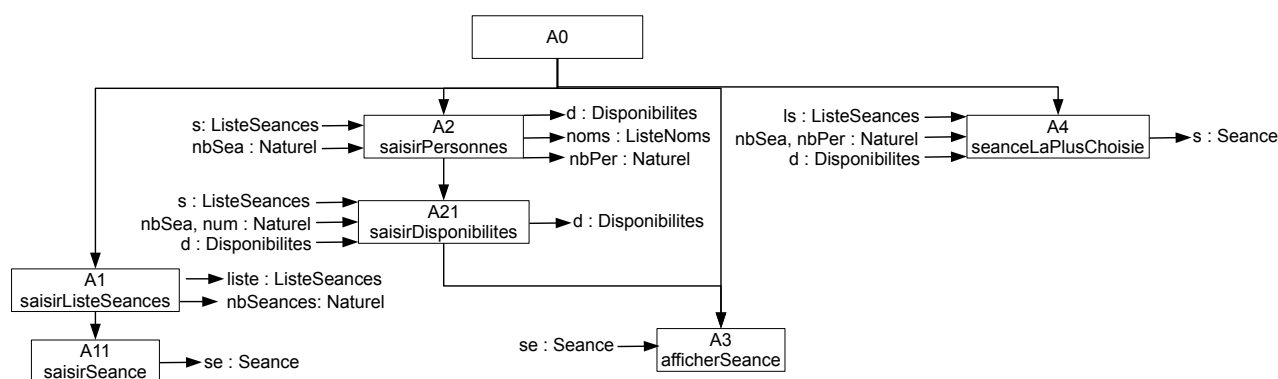
## 2 Écriture de code (9 points)

Vous devez réaliser la conception détaillée d'un programme permettant de définir plusieurs séances horaires pour recueillir les disponibilités de plusieurs personnes et trouver à quelle séance le plus grand nombre de personnes est disponible (une sorte de *Doodle* local sur une machine). Le programme demande dans un premier temps de saisir les séances possibles en indiquant pour chacune d'elle le jour (numéro de jour, de mois et l'année), une heure de début et une heure de fin. Seules les heures pleines (pas de minutes) seront considérées. Le programme doit s'assurer que le numéro de jour est bien entre 1 et 31, celui du mois entre 1 et 12, les heures entre 1 et 24 et que l'heure de fin n'est pas avant celle de début. Nous ne demandons pas de tester la cohérence entre le numéro de jour et celui du mois.

La deuxième étape consiste à saisir l'identité et la disponibilité de plusieurs personnes. Pour chacune, le programme demande un nom et si elle est disponible pour chacune des séances.

Enfin, il affiche la séance à laquelle le plus grand nombre de personnes est disponible. S'il y en a plusieurs, seule la première est affichée.

Une analyse descendante pour ce programme a été réalisée :



Les types de données suivants sont à définir :

- **Seance** pour représenter une séance horaire possible
- **ListeSeances** pour représenter plusieurs séances
- **ListeNoms** pour représenter les noms de tous les participants
- **Disponibilites** pour représenter les disponibilités de chaque participant à chaque séance

Les sous-programmes correspondent aux signatures suivantes :

**procédure saisirListeSeances (S liste : ListeSeances ; S nbSeances : Naturel)**

Cette procédure permet de saisir plusieurs séances proposées. Elle donne en sortie la liste des séances saisies et leur nombre.

**procédure saisirSeance (S se : Seance)**

Cette procédure remplit le paramètre de sortie *se* avec des informations entrées au clavier, en s'assurant qu'elle sont valides (cf ci-dessus).

**procédure saisirPersonnes (E s : ListeSeances ; E nbSea : Naturel ; S d : Disponibilites ; S noms : ListeNoms ; S nbPer : Naturel)**

Cette procédure permet de saisir les disponibilités des participants au sondage. Les entrées donnent les séances proposées et leur nombre. À l'issue de cette procédure les paramètres de sortie *d*, *noms* et *nbPer* sont remplis avec, respectivement, les disponibilités de chaque participant à chaque séance, le nom de tous les participants et leur nombre.

**procédure saisirDisponibilites (E s : ListeSeances ; E nbSea : Naturel ; E/S d : Disponibilites ; E num : Entier)**

Cette procédure permet de saisir les disponibilités d'un participant pour chacune des séances. Les entrées donnent les séances proposées et leur nombre, pour qu'on puisse les afficher à l'écran quand on demande si le participant est disponible, et *num* indique le numéro du participant. Le paramètre *d* est modifié en fonction de ce qui est saisi.

**procédure afficherSeance (E se : Seance)**

Cette procédure affiche à l'écran les informations liées à la séance donnée en paramètre d'entrée.

**fonction seanceLaPlusChoisie (ls : ListeSeances ; nbSea : Naturel ; d : Disponibilites ; nbPer : Naturel) : Seance**

Cette fonction trouve la séance qui recueille le plus de disponibilités et la retourne.

Nom :  
 Prenom :  
 Groupe :

## Questions

Définir en pseudo code les types **Seance**, **ListeNoms**, **ListeSeances** et **Disponibilites** (les constantes limitant le nombre maximum de séances et de personnes sont données).

**Constante** MAX\_SEANCES = 10

**Constante** MAX\_PARTICIPANTS = 20

**Type** Seance = **Structure**

  numJour : **Naturel** :  
 numMois : **Naturel** :  
 annee : **Entier** :  
 debut : **Naturel** :  
 fin : **Naturel** :

**finstructure**

**Type** ListeNoms = **Tableau**[1..MAX\_PARTICIPANTS] de **Chaine de caracteres**

**Type** ListeSeances = **Tableau**[1..MAX\_SEANCES] de **Seance**

**Type** Disponibilites = **Tableau**[1..MAX\_SEANCES][1..MAX\_PARTICIPANTS] de **Booleen**

Écrire en pseudo code la conception détaillée de tous les sous-programmes.

**procédure** saisirListeSeances (**S** liste : ListeSeances ; **S** nbSeances : **Naturel**)

**Déclaration** c : **Caractere**

**debut**

  nbSeances ← 0

**repeter**

**ecrire**('Voulez-vous ajouter une seance ? (o/n)')

**lire**(c)

**si** c = 'o' **alors**

      nbSeances ← nbSeances + 1

      saisirSeance(liste[nbSeances])

**finsi**

**jusqu'a ce que** (c ≠ 'o') ou (nbSeances = MAX\_SEANCES)

**fin**

**procédure** saisirSeance (**S** se : Seance)

**debut**

**repeter**

**ecrire**('Donnez le numero du jour :')

**lire**(se.numJour)

**jusqu'a ce que** (se.numJour > 0) et (se.numJour < 32)

**repeter**

**ecrire**('Donnez le numero du mois :')

**lire**(se.numMois)

**jusqu'a ce que** (se.numMois > 0) et (se.numMois < 13)

**ecrire**('Donnez l'annee :')

**lire**(se.annee)

**repeter**

**ecrire**('Donnez l'heure de debut :')

**lire**(se.debut)

**jusqu'a ce que** (se.debut > 0) et (se.debut < 25)

**repeter**

**ecrire**('Donnez l'heure de fin :')

**lire**(se.fin)

**jusqu'a ce que** (se.fin > 0) et (se.fin < 25) et (se.debut ≤ se.fin)

**fin**

**procédure** saisirPersonnes (**E** s : ListeSeances ; **E** nbSea : **Naturel** ; **S** d : Disponibilites ; **S** noms : ListeNoms ; **S** nbPer : **Naturel**)

**Déclaration** c : **Caractere**

**debut**

```

nbPer ← 0
repete
  ecrire('Voulez-vous ajouter une personne? (o/n)')
  lire(c)
  si c = 'o' alors
    nbPer ← nbPer + 1
    ecrire('Quel est son nom?')
    lire(noms[nbPer])
    saisirDisponibilites(s, nbSea, d, nbPer)
  finsi
jusqu'a ce que (c ≠ 'o') ou (nbPer = MAX_PERSONNES)
fin
procédure saisirDisponibilites (E s : ListeSeances; E nbSea : Naturel; E/S d : Disponibilites; E num : Entier)
  Déclaration c : Caractere; i : Naturel
debut
  pour i ← 1 à nbSea faire
    afficherSeance(s[i])
    ecrire('La personne est-elle disponible pour cette seance? (o/n)')
    si c = 'o' alors
      d[i][num] ← Vrai
    sinon
      d[i][num] ← Faux
    finsi
  finpour
fin
procédure afficherSeance (E se : Seance)
debut
  ecrire('Seance du ', se.numJour, '/', se.numMois, '/', se.annee)
  ecrire(' de ', se.debut, 'h a ', se.fin, 'h')
fin
fonction seanceLaPlusChoisie (ls : ListeSeances; nbSea : Naturel; d : Disponibilites; nbPer : Naturel) : Seance
  Déclaration i, j, nb, resInd, resNb : Naturel
debut
  si nbSea = 0 alors
    retourner nil
  sinon
    resNb ← 0
    pour i ← 1 à nbSea faire
      nb ← 0
      pour j ← 1 à nbPer faire
        si d[i][j] alors
          nb ← nb+1
        finsi
      finpour
      si nb > resNb alors
        resNb ← nb
        resInd ← i
      finsi
    finpour
    retourner ls[resInd]
  finsi
fin

```

Nom :  
Prenom :  
Groupe :

**Déclaration :** liste : ListeSeances ; nbSeances : **Naturel** ; dis : Disponibilites ; pers : ListeNoms ; nbPers : **Naturel** ; resultat : Seance

**debut**

```
saisirListeSeances(liste, nbSeances)
saisirPersonnes(liste, nbSeances, dis, pers, nbPers)
resultat ← seanceLaPlusChoisie(liste, nbSeances, dis, nbPers)
ecrire('La seance la plus choisie est :')
afficherSeance(resultat)
```

**fin**



**2) Analyse descendante**

Dessiner une analyse descendante possible pour ce programme.



**3) Donner les signatures des sous-programmes correspondant à cette analyse.**
