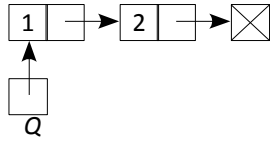


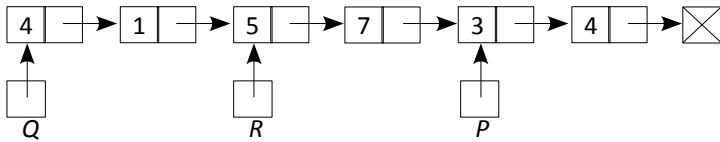
TD d'algorithmique INF220 – TD4 – Listes

Exercice 1 – Manipulation des fonctions de base sur les listes

Q1. Dessinez l'ensemble des listes *P*, *Q* et *R* après chaque instruction de l'algorithme **initialiseListes** ci-contre. Voici par exemple le dessin de la situation après la ligne 1 :



Q2. Quelle instruction ajoutez-vous à la fin de l'algorithme pour obtenir les listes suivantes ?



Algorithme initialiseListes

Variables : *P*, *Q*, *R*, listes d'entiers

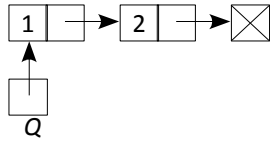
Début

1. $Q \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{NULL}))$
 2. $P \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{creeListe}(3, \text{NULL})))$
 3. $Q \leftarrow \text{creeListe}(4, P)$
 4. $\text{tete}(\text{suivant}(P)) \leftarrow 6$
 5. $R \leftarrow \text{creeListe}(5, \text{suivant}(\text{suivant}(P)))$
 6. $\text{suivant}(P) \leftarrow R$
 7. $P \leftarrow \text{suivant}(R)$
 8. $\text{suivant}(P) \leftarrow \text{creeListe}(\text{tete}(Q), \text{NULL})$
- Fin

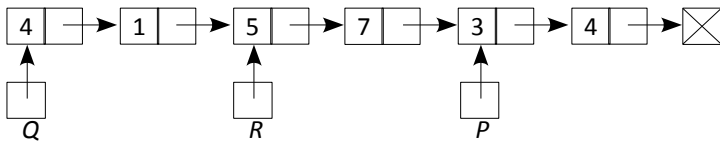
TD d'algorithmique INF220 – TD4 – Listes

Exercice 1 – Manipulation des fonctions de base sur les listes

Q1. Dessinez l'ensemble des listes *P*, *Q* et *R* après chaque instruction de l'algorithme **initialiseListes** ci-contre. Voici par exemple le dessin de la situation après la ligne 1 :



Q2. Quelle instruction ajoutez-vous à la fin de l'algorithme pour obtenir les listes suivantes ?



Algorithme initialiseListes

Variables : *P*, *Q*, *R*, listes d'entiers

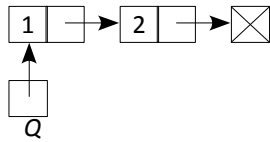
Début

1. $Q \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{NULL}))$
 2. $P \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{creeListe}(3, \text{NULL})))$
 3. $Q \leftarrow \text{creeListe}(4, P)$
 4. $\text{tete}(\text{suivant}(P)) \leftarrow 6$
 5. $R \leftarrow \text{creeListe}(5, \text{suivant}(\text{suivant}(P)))$
 6. $\text{suivant}(P) \leftarrow R$
 7. $P \leftarrow \text{suivant}(R)$
 8. $\text{suivant}(P) \leftarrow \text{creeListe}(\text{tete}(Q), \text{NULL})$
- Fin

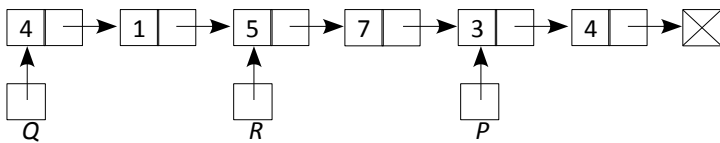
TD d'algorithmique INF220 – TD4 – Listes

Exercice 1 – Manipulation des fonctions de base sur les listes

Q1. Dessinez l'ensemble des listes *P*, *Q* et *R* après chaque instruction de l'algorithme **initialiseListes** ci-contre. Voici par exemple le dessin de la situation après la ligne 1 :



Q2. Quelle instruction ajoutez-vous à la fin de l'algorithme pour obtenir les listes suivantes ?



Algorithme initialiseListes

Variables : *P*, *Q*, *R*, listes d'entiers

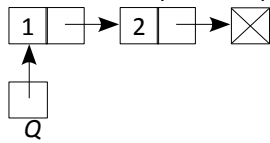
Début

1. $Q \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{NULL}))$
 2. $P \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{creeListe}(3, \text{NULL})))$
 3. $Q \leftarrow \text{creeListe}(4, P)$
 4. $\text{tete}(\text{suivant}(P)) \leftarrow 6$
 5. $R \leftarrow \text{creeListe}(5, \text{suivant}(\text{suivant}(P)))$
 6. $\text{suivant}(P) \leftarrow R$
 7. $P \leftarrow \text{suivant}(R)$
 8. $\text{suivant}(P) \leftarrow \text{creeListe}(\text{tete}(Q), \text{NULL})$
- Fin

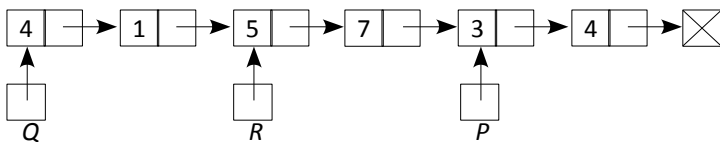
TD d'algorithmique INF220 – TD4 – Listes

Exercice 1 – Manipulation des fonctions de base sur les listes

Q1. Dessinez l'ensemble des listes *P*, *Q* et *R* après chaque instruction de l'algorithme **initialiseListes** ci-contre. Voici par exemple le dessin de la situation après la ligne 1 :



Q2. Quelle instruction ajoutez-vous à la fin de l'algorithme pour obtenir les listes suivantes ?



Algorithme initialiseListes

Variables : *P*, *Q*, *R*, listes d'entiers

Début

1. $Q \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{NULL}))$
 2. $P \leftarrow \text{creeListe}(1, \text{creeListe}(2, \text{creeListe}(3, \text{NULL})))$
 3. $Q \leftarrow \text{creeListe}(4, P)$
 4. $\text{tete}(\text{suivant}(P)) \leftarrow 6$
 5. $R \leftarrow \text{creeListe}(5, \text{suivant}(\text{suivant}(P)))$
 6. $\text{suivant}(P) \leftarrow R$
 7. $P \leftarrow \text{suivant}(R)$
 8. $\text{suivant}(P) \leftarrow \text{creeListe}(\text{tete}(Q), \text{NULL})$
- Fin

Exercice 2 – Écriture de fonctions sur les listes et implémentation d'une file

- Q1. En utilisant une boucle **Tant que**, écrivez en pseudo-code un algorithme **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Par exemple, **afficheListe(creeListe(1,creeListe(2,NULL))** affichera 1 puis 2.
- Q2. Sans utiliser de boucle **Tant que**, écrivez en pseudo-code un algorithme récursif **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Vous constatez qu'un parcours de liste peut s'effectuer de manière récursive, ou avec une boucle.
- Q3. Écrivez un algorithme **longueurListe** qui prend en entrée une liste L d'entiers et renvoie le nombre de cases de L . Si l'on exécute cette instruction juste après l'instruction 8 de l'algorithme **initialiseListes** de l'exercice 1, que renvoie **longueurListe(R)** ?
- Q4. Écrivez un algorithme **insereAvantCase** qui prend en entrée une liste L d'entiers et deux entiers i et x , et insère avant la i -ième case de la liste une case contenant l'entier x . La question Q2 de l'exercice 1 pourra vous être utile.
- Q5. En utilisant les algorithmes **longueurListe** et **insereAvantCase**, écrivez un algorithme **enfile** qui prend en entrée une liste L d'entiers et un entier a , et insère à la fin de L une nouvelle case contenant a .
- Q6. Écrivez un algorithme **defile** qui renvoie la valeur de la première case de la liste d'entiers en entrée, et supprime cette case.

Exercice 2 – Écriture de fonctions sur les listes et implémentation d'une file

- Q1. En utilisant une boucle **Tant que**, écrivez en pseudo-code un algorithme **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Par exemple, **afficheListe(creeListe(1,creeListe(2,NULL))** affichera 1 puis 2.
- Q2. Sans utiliser de boucle **Tant que**, écrivez en pseudo-code un algorithme récursif **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Vous constatez qu'un parcours de liste peut s'effectuer de manière récursive, ou avec une boucle.
- Q3. Écrivez un algorithme **longueurListe** qui prend en entrée une liste L d'entiers et renvoie le nombre de cases de L . Si l'on exécute cette instruction juste après l'instruction 8 de l'algorithme **initialiseListes** de l'exercice 1, que renvoie **longueurListe(R)** ?
- Q4. Écrivez un algorithme **insereAvantCase** qui prend en entrée une liste L d'entiers et deux entiers i et x , et insère avant la i -ième case de la liste une case contenant l'entier x . La question Q2 de l'exercice 1 pourra vous être utile.
- Q5. En utilisant les algorithmes **longueurListe** et **insereAvantCase**, écrivez un algorithme **enfile** qui prend en entrée une liste L d'entiers et un entier a , et insère à la fin de L une nouvelle case contenant a .
- Q6. Écrivez un algorithme **defile** qui renvoie la valeur de la première case de la liste d'entiers en entrée, et supprime cette case.

Exercice 2 – Écriture de fonctions sur les listes et implémentation d'une file

- Q1. En utilisant une boucle **Tant que**, écrivez en pseudo-code un algorithme **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Par exemple, **afficheListe(creeListe(1,creeListe(2,NULL))** affichera 1 puis 2.
- Q2. Sans utiliser de boucle **Tant que**, écrivez en pseudo-code un algorithme récursif **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Vous constatez qu'un parcours de liste peut s'effectuer de manière récursive, ou avec une boucle.
- Q3. Écrivez un algorithme **longueurListe** qui prend en entrée une liste L d'entiers et renvoie le nombre de cases de L . Si l'on exécute cette instruction juste après l'instruction 8 de l'algorithme **initialiseListes** de l'exercice 1, que renvoie **longueurListe(R)** ?
- Q4. Écrivez un algorithme **insereAvantCase** qui prend en entrée une liste L d'entiers et deux entiers i et x , et insère avant la i -ième case de la liste une case contenant l'entier x . La question Q2 de l'exercice 1 pourra vous être utile.
- Q5. En utilisant les algorithmes **longueurListe** et **insereAvantCase**, écrivez un algorithme **enfile** qui prend en entrée une liste L d'entiers et un entier a , et insère à la fin de L une nouvelle case contenant a .
- Q6. Écrivez un algorithme **defile** qui renvoie la valeur de la première case de la liste d'entiers en entrée, et supprime cette case.

Exercice 2 – Écriture de fonctions sur les listes et implémentation d'une file

- Q1. En utilisant une boucle **Tant que**, écrivez en pseudo-code un algorithme **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Par exemple, **afficheListe(creeListe(1,creeListe(2,NULL))** affichera 1 puis 2.
- Q2. Sans utiliser de boucle **Tant que**, écrivez en pseudo-code un algorithme récursif **afficheListe** qui prend en entrée une liste d'entiers et affiche le contenu de chaque case. Vous constatez qu'un parcours de liste peut s'effectuer de manière récursive, ou avec une boucle.
- Q3. Écrivez un algorithme **longueurListe** qui prend en entrée une liste L d'entiers et renvoie le nombre de cases de L . Si l'on exécute cette instruction juste après l'instruction 8 de l'algorithme **initialiseListes** de l'exercice 1, que renvoie **longueurListe(R)** ?
- Q4. Écrivez un algorithme **insereAvantCase** qui prend en entrée une liste L d'entiers et deux entiers i et x , et insère avant la i -ième case de la liste une case contenant l'entier x . La question Q2 de l'exercice 1 pourra vous être utile.
- Q5. En utilisant les algorithmes **longueurListe** et **insereAvantCase**, écrivez un algorithme **enfile** qui prend en entrée une liste L d'entiers et un entier a , et insère à la fin de L une nouvelle case contenant a .
- Q6. Écrivez un algorithme **defile** qui renvoie la valeur de la première case de la liste d'entiers en entrée, et supprime cette case.