

I3 - Algorithmique

Durée : 1h30

Documents autorisés : **AUCUN** (calculatrice comprise)

Remarques :

- Veuillez lire attentivement les questions avant de répondre.
- Le barème donné est un barème indicatif qui pourra évoluer lors de la correction.
- Rendez une copie propre.
- N'utilisez pas de crayon à papier.

1 Compréhension du cours (7 points)

La figure 1 présente l'analyse descendante du problème du chiffrement d'un message à l'aide du chiffre de César.

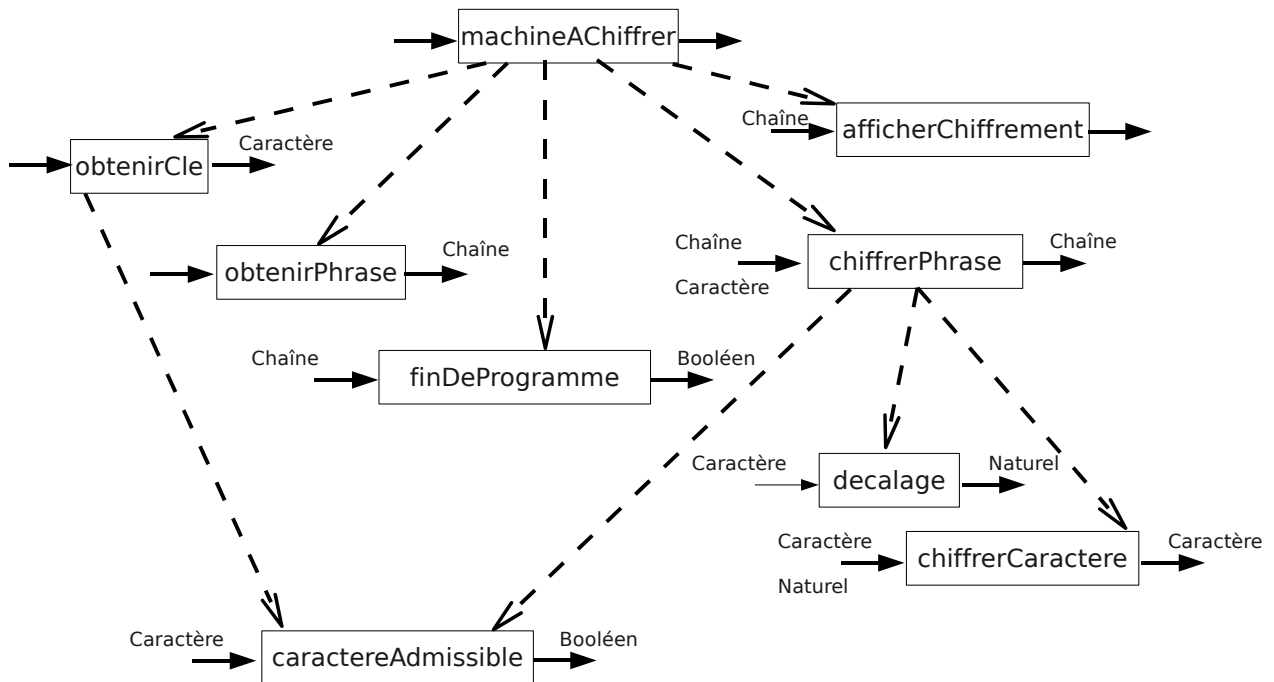


FIG. 1 – Analyse descendante

1. Conception préliminaire : donnez la signature des fonctions et procédures correspondant aux opérations proposées par cette analyse descendante.
2. Conception détaillée : en supposant que vous possédez les trois fonctions suivantes, donnez le code de la fonction/procédure correspondant à l'opération `chiffrerPhrase` :
 - **fonction** longueur (uneChaîne : **Chaîne de caractères**) : **Naturel**
 - **fonction** iemeCaractere (uneChaîne : **Chaîne de caractères**, position : **Naturel**) : **Caractère**
 - **fonction** caractereEnChaîne (unCaractere : **Caractère**) : **Chaîne de caractères**

2 Racine carrée d'un nombre : développement limité (5 points)

Lorsque x est proche de 0, $\sqrt{1+x}$ peut être approximé à l'aide de la formule suivante :

$$\sum_{i=1}^n (-1)^{i+1} \frac{(2i-2)!}{i!(i-1)!2^{2i-1}} x^i$$

Donnez le corps de la fonction suivante qui calcule une approximation de $\sqrt{1+x}$ jusqu'au rang n . Utilisez uniquement des variables locales et les paramètres donnés (n'utilisez aucune autre fonction, pas d'analyse descendante) :

– **fonction** `racineCarreeUnPlusX` (x : **Reel**, n : **Naturel**) : **Reel**

3 Racine carrée d'un nombre : recherche par dichotomie (8 points)

L'objectif de cet exercice est de rechercher une valeur approchée de la racine carrée d'un nombre réel positif n ($n \geq 1$) à ϵ près à l'aide d'un algorithme dichotomique.

Pour rappel :

« La dichotomie (“couper en deux” en grec) est, en algorithmique, un processus itératif [...] de recherche où, à chaque étape, on coupe en deux parties (pas forcément égales) un espace de recherche qui devient restreint à l'une de ces deux parties.

On suppose bien sûr qu'il existe un test relativement simple permettant à chaque étape de déterminer l'une des deux parties dans laquelle se trouve une solution. Pour optimiser le nombre d'itérations nécessaires, on s'arrangera pour choisir à chaque étape deux parties sensiblement de la même “taille” (pour un concept de “taille” approprié au problème), le nombre total d'itérations nécessaires à la complétion de l'algorithme étant alors logarithmique en la taille totale du problème initial. » (wikipédia).

1. Définir « l'espace de recherche » pour le problème de la recherche d'une racine carrée.
2. Quelle condition booléenne permet de savoir si il doit y avoir une nouvelle itération ?
3. Quel test va vous permettre de savoir dans laquelle des deux parties se trouve la solution ?
4. Proposez l'algorithme de la fonction suivante (on suppose que n et ϵ sont positifs et que n est supérieur ou égal à 1) :
– **fonction** `racineCarree` (n, ϵ : **Reel**) : **Reel**
5. Donnez le code Pascal correspondant.