Contrôle de Info C

Durée : 1h30 Aucun document autorisé
Nom : Prénom :

▶ 1. Dans la fonction suivante, <u>insérez</u> les affirmations qui donnent sa sémantique et qui prouvent la validité de l'énoncé itératif.

```
// Antécédent: ...
// Conséquent: ...
int mystere(int n) {
  int i=0, j=1, s=0;
  // Invariant : ....
  while(i<n) {
    // ...
// ...
    s = s + j;
    // ...
    // ...
j = j + 2;
    // ...
    // ...
    i = i + 1;
    // ...
  }
  // ...
  return s;
```

Soit d'une fonction f continue sur un intervalle réel [a,b], on veut trouver les x, appelés <u>zéros</u>, tels que f(x)=0. Pour cela, nous utiliserons la méthode <u>dichotomique</u> suivante qui part du principe que si f(a)f(b)<0 alors il existe <u>au moins un zéro</u>:

	1. choisir $[a, b]$ qui contient a priori un zéro;						
	2. choisir le milieu m de l'intervalle $[a,b]$;						
	3. si $f(m) = 0$ alors on a trouvé un zéro						
	 4. si f(a)f(m) < 0 alors définir le nouvel intervalle [a, m]; 5. si f(b)f(m) < 0 alors définir le nouvel intervalle [m, b]; 6. recommencer à partir de 2. 						
•	2. Écrivez la fonction zero qui renvoie un zéro d'une fonction f selon l'algo rithme <u>itératif</u> précédent. Un zéro sera trouvé à un epsilon près, à passer en paramètre. Enfin, pour garantir l'achèvement de l'énoncé itératif, un nombre maximal d'itérations sera également passé en paramètre.						

•	3. Écrivez la <u>fonction</u> <u>lireEntier</u> qui lit sur l' <u>entrée standard</u> caractère à caractère (à l'aide de la fonction <u>getchar()</u> , à l'exclusion de toute autre fonction), un entier décimal (positif ou négatif) et qui renvoie la valeur entière décimale (de type <u>int</u>) que représente la suite de caractères. Quelques exemples : 123 +34 -230 77 +10000 -255						
	Vous devrez gérer les dépassements de capacité. Les constantes INT_MAX et INT_MIN du fichier limits.h définissent les bornes de l'intervalle du type INT. Si le nombre n'appartient pas à l'intervalle, vous signalerez l'erreur par un message approprié sur la sortie d'erreur standard.						

On lit l'entrée standard (uniquement avec getchar) qui contient une suite de lettres $a,\ b$ et c, et uniquement ces 3 lettres. Dans cette suite, on souhaite reconnaître des sous-suites qui commencent par un ou plusieurs a, suivis d'un nombre pair de b, suivis par un ou plusieurs c.

▶ 4. Écrivez un <u>programme</u> qui lit <u>toute</u> l'entrée standard qui écrit sur la sortie standard le nombre de sous-suites, appelées <u>motifs</u>, bâtis sur le modèle précédent. Pensez à programmer un automate fini. Par exemple, votre programme écrira 3, si l'entrée standard contient :

les motifs reconnus sont en rouge ci-dessous :						
cbbbaabbbbccbbcacabcabbcccabbbbbaaabbbbbcaabbbccccbbccc						

faible, 20, est à l'indice 0, et le bit de poids fort, 2NB_BITS-1, est à l'indice NB_BITS-1. Les questions qui suivent sont liées. ▶ 5. À l'aide d'un define, définissez la constante NB_BITS égale au nombre d'éléments du tableau nécessaires à la représentation d'un entier non signé. ▶ 6. Écrivez la fonction valeurDecimale qui renvoie la valeur décimale d'un entier non signé donné à partir de sa représentation binaire contenue dans un tableau. Vous ne ferez aucune évaluation à la puissance. L'en-tête de cette fonction est le suivant : unsigned int valeurDecimale(const short bits[])

On représente un entier non signé (unsigned int) sous forme <u>binaire</u> à l'aide d'un <u>tableau</u> d'entiers courts (short) contenant des 0 et des 1. Le bit de poids

void dec	alageDroi	it(sho	rt bit	s[], o	const	int n)	