

Programmation et Langages

Devoir surveillé du 20 octobre 2022

Durée : une heure – document autorisé : une feuille A4 recto-verso**Exercice 1 (~ 4 points)**

On rappelle qu'en C les variables de type `char` sont des entiers codés sur 8 bits. S'il sont "signés" leur plage de valeur est $[-128, 127]$. S'il sont "non signés" (*unsigned*) leur plage de valeur est $[0, 255]$.

On considère le fragment de programme C suivant :

```
#include <stdio.h>
#define X ... // valeur initiale de x

int main(){
    unsigned char x = X;
    if (x < 255) {
        if (x+10 < 255)
            printf("ok1\n") ;
        else
            printf("ko1\n") ;
    } ;

    if (x+10 > 255) {
        if (x > 255)
            printf("ok2\n") ;
        else
            printf("ko2\n") ;
    } ;
}
```

Q1. donner l'affichage obtenu à l'écran dans les trois cas suivants :
X vaut 0; X vaut 10; X vaut 240.

Q2. Lorsque X vaut 250 on obtient l'affichage suivant :

```
ko1
ko2
```

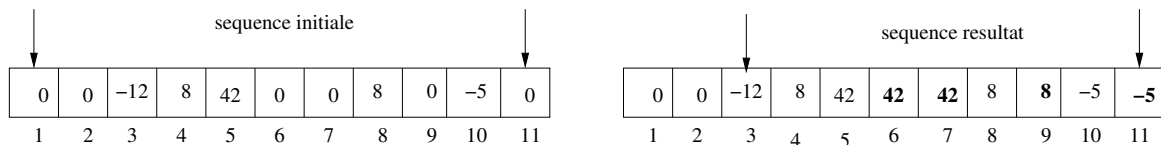
D'après vous pourquoi? Justifiez votre réponse en donnant les valeurs de `x` et de `x+10`.

Exercice 2 (~ 8 points)

On considère une séquence de $L+1$ entiers positifs mémorisée dans un tableau T . On souhaite remplacer dans cette séquence chaque entier égal à 0 par le premier entier non nul qui le précède. Les entiers égaux à 0 non précédés par un entier non nul sont supprimés.

Sur l'exemple ci-dessous :

- les valeurs 0 aux indices 6 et 7 sont remplacées par la valeur **42**
- la valeur 0 à l'indice 9 est remplacée par la valeur **8**
- la valeur 0 à l'indice 11 est remplacée par la valeur **-5**
- la séquence résultat débute à l'indice 3 (les valeurs 0 des indices 1 et 2 sont supprimées)



L'algorithme suivant permet de réaliser cette opération : il mémorise la séquence initiale (lue au clavier) dans un tableau T , il modifie le contenu de ce tableau pour produire la séquence résultat et il affiche à l'écran cette séquence résultat.

lexique

L : la constante entière 7

T : un tableau sur $[0 .. L]$ d'entiers ≥ 0

i , $ideb$, $cour$: des entiers ≥ 0

debut

{lecture de la séquence dans le tableau T }
pour i parcourant $0 .. L$: lire(T_i)

{recherche du premier element non nul de T }

$i \leftarrow 0$

tantque $i \leq L$ et puis $T_i = 0$: $i \leftarrow i+1$

$ideb \leftarrow i$ { $ideb$ est l'indice du premier élément de la séquence résultat}

{mise a jour des éléments suivants de T }

tantque $i \leq L$

$cour \leftarrow T_i$ { $cour$ est l'élément non nul courant}

{on remplace par $cour$ tous les éléments nuls consécutifs suivant $cour$ dans T }

$i \leftarrow i+1$

tantque $i \leq L$ et puis $T_i = 0$

$T_i \leftarrow cour$; $i \leftarrow i+1$

{affichage de la séquence résultat }

pour i parcourant $ideb .. L$: écrire(T_i)

fin

Q1. Donnez une traduction en C (la plus fidèle possible!) de cet algorithme.

Q2. Donnez des exemples d'entrées qui permettraient de tester cet algorithme (pas plus de 5 exemples, en justifiant ces choix).

Q2. Au lieu d'afficher la séquence résultat à l'écran on souhaite désormais la mémoriser dans un tableau `R`. Ce tableau sera terminé par la marque de fin `0`. Reprenez votre réponse à la **Q1** en :

- ajoutant la déclaration de `R` dans le lexique ;
- modifiant le corps du programme pour que la séquence résultats soit mémorisée dans `R`.

Il n'est pas nécessaire de ré-écrire complètement ce programme ...

Q3. Est-ce que l'on aurait pu utiliser `0` comme marque de fin pour la séquence initiale dans `T` ? Et pourquoi peut-on le faire pour `R` ?

Exercice 3 (~ 8 points)

On souhaite définir en C un type produit de nom `Article` constitué des champs suivants :

- son prix, qui est un entier strictement positif ;
- sa couleur, qui est un élément de l'ensemble {Rouge, Noir, Bleu, Vert} ;
- sa taille, qui est constituée d'un couple (largeur, longueur), tout deux étant des entiers strictement positifs ;
- son identifiant, qui est une chaîne de 10 caractères maximum ;
- enfin un booléen qui indique si ce produit est ou non "en stock".

Q1. Donner une définition en C de ce type `Article`

Q2. Ecrire un fragment de programme C qui indique si deux variables `a1` et `a2` de type `Article` sont ou non égales. On rappelle que la fonction `strcmp(s1, s2)` permet de comparer deux chaînes de caractères `s1` et `s2` en renvoyant la valeur `0` si et seulement si ces deux chaînes son égales.

Q3. Ecrire un fragment de programme C qui associe un entier à une variable `a1` de type `Article` en additionnant les valeurs suivantes :

- son prix ;
- sa couleur (en considérant que Rouge vaut 0, Noir vaut 1, etc.)
- sa largeur et sa longueur ;
- la somme des codes Ascii de son identifiant ;
- la valeur 1 si l'article est en stock et la valeur 0 sinon.

On rappelle que la fonction `strlen (s)` renvoie le nombre de caractères de la chaîne `s`.

Q4. L'entier obtenu à la question **Q3** est-il toujours "unique" (i.e., est-ce qu'un même entier peut être associé à deux articles différents) ? Pourquoi ?