

Langage C

Interrogation du 24 mars 2021, durée 60 minutes

Le barème est donné à titre indicatif.

Consignes

Vous devez déposer vos copies sur moodle, il y a un dépôt par groupe de TP. Il y a aussi un dépôt séparé pour les étudiants handicapés qui bénéficient de tiers temps supplémentaire.

Quand je parle de copies il s'agit de deux fichiers avec le code sources, le premier fichier portera le nom `cNomPrenom.c` avec les deux fonctions de la première partie (chaînes de caractères) et le deuxième qui portera le nom `sNomPrenom` avec les fonction de la deuxième partie (Pointeurs, structures et tableaux).

Comme vous avez sans doute deviné, à la place de Nom et Prenom vous devez mettre **votre nom et votre prénom**. Mais surtout ce qui est le plus important que le nom du premier fichier commence par `c` et le nom du deuxième par `s`. **Le non-respect de cette consigne engendre le travail supplémentaire pour les correcteurs et sera pénalisé (un point de moins).**

Au début de chaque fichier vous mettrez votre nom, prénom, numéro d'étudiant et le groupe TP, tout cela comme un commentaire du programme C (donc entre `/* */`).

Vous pouvez aussi faire une copie manuscrite sur deux feuilles et déposer une photo (lisible) des feuilles. Mais cela doit rester exceptionnel. (Autant que je sache il existe le prêt de portable par l'université donc pourquoi sur une feuille manuscrite?)

En cas de difficulté d'accès au moodle (et uniquement dans ce cas) vous pouvez envoyer votre copie par mail à zielonka@irif.fr avec le sujet `interro In` si vous êtes en info où `n` est le numéro de votre groupe tp info, où `interro MIn` si vous êtes dans le groupe `n` de math-info. **Le mail doit être signé avec votre nom, prénom**. Les mails anonymes risquent d'attendre beaucoup de temps pour être corrigés. Vous envoyez votre copie comme une pièce jointe. N'attendez pas pour envoyez le mail, les copies qui arrivent par mail trop tard après la fermeture de dépôt risquent d'être pénalisés.

Votre code doit être écrit de façon lisible, avec des indentations et des accolades appropriées permettant de voir la fin de blocs de code (fin de boucles, etc.).

Tous les fichiers passeront le test de détecteur de plagiat <https://theory.stanford.edu/~aiken/moss/>

1 Chaînes de caractères

Exercice 1 : 9.5 points

Écrire la fonction

```
1 char *sans_lettre( const char *s, char x )
```

qui prend en paramètre l'adresse d'une chaîne de caractère et un caractère et construit et retourne une nouvelle chaîne de caractère obtenue de façon suivante :

1. on supprime toutes les occurrences de la lettre `x`,
2. les caractères qui restent sont rangés dans l'ordre inverse.

Par exemple si `s` pointe vers la chaîne `"ala ma kota"`, la suppression de `'a'` donne `"l m kot"` donc `sans_lettre` doit retourner le pointeur vers la chaîne `"tok m l"`.

Notez que le résultat peut être la chaîne vide `"`, par exemple dans `sans_lettre("aaaaa", 'a')`.

Exercice 2 : 0.5 points

Écrire la fonction `main` qui teste la fonction `sans_lettre` de l'exercice précédent sur trois chaînes de caractères, la chaîne vide, la chaîne qui ne contient que la lettre à supprimer, la chaîne qui contient la lettre à supprimer mais aussi d'autres lettres. Chaque fois afficher la chaîne passée en argument et la chaîne retournée par la fonction. (Rappel : format `"%s"` pour afficher une chaîne de caractères.)

2 Pointeurs, structures et tableaux

On définit

```
1 typedef struct{
2     int x;
3     int y;
4 } point;
```

Exercice 3 : 4 points

Écrire la fonction

```
1 void translate( point *p, int dist, char dir)
2
```

La fonction doit déplacer le point dont l'adresse `p` est donnée par le premier paramètre de `dist` unités dans la direction donnée par `dir`.

Si `dir == 'x'` alors on ajoute `dist` dans la coordonnée `x` du point, si `dir == 'y'` on ajoute `dist` dans la coordonnée `y` du point.

Si `dir` est un caractère différent de `'x'` ou `'y'` alors le point à l'adresse `p` ne sera pas modifié.

Exercice 4 : 2 points

Écrire la fonction

```
1 void translate_tab( int len, point v[], int dist, char dir)
2
```

qui applique `translate` à tous les points de tableau `v` qui contient `len` points.

Exercice 5 : 4 points

Tester la fonction de l'exercice précédent dans un `main`. Vous devez définir un tableau de 3 points et initialiser avec les points de votre choix : le premier avec deux coordonnées positives, deuxième avec deux coordonnées négatives, et troisième avec la coordonnées x positive et y négative.

On préfère l'initialisation du tableau au moment de la déclaration, comme une seule instruction. Mais si cela vous pose un problème, vous pouvez d'abord définir le tableau et ensuite initialiser le contenu élément par élément et champ par champ (mais cela donne moins de points).

Afficher tous les éléments du tableau en utilisant la fonction

```
1 void afficher_point( point q ){  
2     printf( "(%d,%d)\n" , q.x, q.y );  
3 }  
4
```

Appliquer `translate` au premier élément du tableau en faisant une translation de 10 dans la direction '`x`'.

Utiliser `translate_tab` pour déplacer les éléments d'indices 1 et 2 du tableau de -50 dans la direction '`y`' (il s'agit bien d'un seul appel à `translate_tab` et non pas de deux appels à `translate`).

Re-afficher tous les éléments du tableau.