

Examen de Langage C (V. Granet)

Durée : 1h30

Aucun document autorisé

Mobiles interdits

Notez que les affirmations (antécédents, conséquents, rôles, et invariants) dans vos codes C entreront pour partie dans la note finale.

- 1. En utilisant la notation de tableau, écrivez la fonction `strRchr` qui renvoie l'indice de la dernière occurrence d'un caractère `c` dans une chaîne de caractère `s`. Si le caractère n'est pas présent dans la chaîne, la fonction `strRchr` renvoie `-1`.

```

/*
 * Rôle : renvoie l'indice de la dernière d'occurrence de c dans s
 *       et -1 si non trouvée
 */
int strRchr(const char s[], const char c) {
    int i=0;
    while (s[i]!='\0') i++;
    assert(*s=='\0');
    while (--i>=0)
        if (s[i]==c)
            // dernière occurrence de c dans s
            return i;
    // c n'appartient pas à s
    return -1;
}

/*
 * Rôle : renvoie l'indice de la dernière d'occurrence
 *       de c dans s et -1 si non trouvée
 */
int strRchr(char s[], const char c) {
    int i=0, d_occ=-1;

    while (s[i]!='\0') {
        // d_occ est l'indice de la dernière
        // occurrence c de s[0] à s[i-1], ou -1
        if (s[i]==c)
            d_occ = i;
        i++;
    }
    return d_occ;
}

```

- 2. Récrivez la fonction `strRchr` précédente, mais cette fois-ci, en utilisant la notation de pointeur. La fonction `strRchr2` renverra l'adresse de la dernière occurrence d'un caractère `c` dans une chaîne

de caractère `s`. Si le caractère n'est pas présent dans la chaîne, la fonction `strRchr2` renvoie `NULL`.

```

/* Rôle : renvoie l'adresse de la dernière d'occurrence de c dans s
 *       et NULL si non trouvée
 */
const char *strRchr2(const char *s, const char c) {
    const char *s1 = s;
    while (*s) s++;
    assert(*s=='\0');
    while (--s >= s1)
        if (*s == c)
            // dernière occurrence de c dans s
            return s;
    // c n'appartient pas à s
    return NULL;
}

/* Rôle : renvoie l'adresse de la dernière d'occurrence
 *       de c dans s et NULL si non trouvée
 */
const char *sstrRchr2(const char *s, const char c)
{
    const char *d_occ = NULL;

    while (*s) {
        // d_occ est l'adresse de la dernière
        // occurrence c de s[0] à s-1, ou NULL
        if (*s == c)
            d_occ=s;
        s++;
    }
    return d_occ;
}

```

- 3. Écrivez une fonction `main` qui donne plusieurs exemples d'utilisation de vos deux fonctions `strRchr` précédentes.

```

int main(void) {
    printf("%d\n", strRchr("hello", 'l')); // => 3
    printf("%d\n", strRchr("hello", 'y')); // => -1
    printf("%s\n", strRchr2("hello", 'e')); // => "ello"
    printf("%c\n", *strRchr2("hello", 'e')); // => 'e'
    printf("%s\n", strRchr2("hello", 'x')); // => error: segmentation fault !

    return EXIT_SUCCESS;
}

```

- 4. Écrivez la déclaration du type `Date`, une structure formée de 3 champs entiers, `jour`, `mois` et `annee` pour représenter une date.

```

struct Date {
    int jours, mois, annee;
};

```

.....

- 5. Écrivez la déclaration du type `Personne`, une structure, pour représenter une personne. Elle sera formée de 3 champs : `nom` (une chaîne de caractères), `sexe` (homme ou femme, un type énuméré) et `naissance` (la date de naissance, de type `Date`).

```

#define MAXALPHA 20

```

```

struct Personne {
    char nom[MAXALPHA];
    enum {homme, femme} sexe;
    struct Date naissance;
};

```

.....

- 6. Écrivez la déclaration de type d'un tableau `tpers` de `MAXPERSONNES` `Personne`. Vous définirez la constante `MAXPERSONNES`, par exemple, égale à 100

```

#define MAXPERSONNES 100

```

```

struct Personne tpers[MAXPERSONNES];

```

.....

- 7. On considère que le tableau de `Personne` `tpers` est entièrement initialisé (i.e. les `MAXPERSONNES` personnes). Écrivez un programme qui écrit sur la sortie standard le nom et l'âge de toutes les femmes qui sont nées avant 1999. On considère que l'année courante est 2017. Pour calculer l'âge d'une personne, vous définirez la macro `AGE` avec une date en paramètre.

```

#define ANNEECOURANTE 2017

```

```

#define AGE(d) (ANNEECOURANTE - (d).annee)

```

```

int main(void) {
    struct Personne tpers[MAXPERSONNES];
    // initialiser la tableau tpers
    // ....
    // parcourir tout le tableau de Personne
    for (int i=0; i<MAXPERSONNES; i++)
        if (tpers[i].sexe == femme && tpers[i].naissance.annee<1999)
            printf("nom : %s - âge : %d\n", tpers[i].nom, AGE(tpers[i].naissance));
    // assert: le nom et l'âge de toutes les femmes nées avant 1999
    // sont écrits sur la sortie standard
    return EXIT_SUCCESS;
}

```

.....