

Devoir surveillé du 17 novembre 2020

Durée : une heure – Documents autorisés : une feuille A4 recto-verso + fiche de traduction Algo ↔ C

Exercice 1 (~ 4 points)

On considère les fonctions f1, f2 et f3 et f4 ci-dessous :

```
int f1 (int x) {  
    return x+1;  
}
```

```
int f2 (int x) {  
    x = x+1 ;  
    return x ;  
}
```

```
int f3 (int x) {  
    int *p ;  
    p = &x ;  
    x = x+1 ;  
    return *p ;  
}
```

```
int f4 (int x) {  
    int *p ;  
    p = &x ;  
    *p = *p+1 ;  
    return x ;  
}
```

On considère le programme principal suivant :

```
int main() {  
    int a, b ;  
    a = 42 ;  
    b = f(a) ;  
    printf("%d %d\n", a, b) ;  
    return 0 ;  
}
```

Indiquer les valeurs affichées par ce programme dans les quatre cas suivants :

1. f est la fonction f1;
2. f est la fonction f2;
3. f est la fonction f3.
4. f est la fonction f4.

Exercice 2 (~ 8 points)

On considère des ensembles d'éléments représenté par des tableaux de booléens selon le lexique ci-dessous :

Element : un type ne contenant pas plus de 256 valeurs

Taille : la constante entière 255

Ensemble: le type tableau sur $[0, \text{Taille}]$ de booléens

Q1. Donnez une traduction de ce lexique en langage C lorsque **Element** est le type **char**. On rappelle que les booléens sont représentés par des entiers en C.

Q2. Peut-on adapter ce lexique pour représenter des ensembles dont les éléments seraient des jours de la semaine, c'est à dire des valeurs allant de **Lundi**, **Mardi**, ... à **Dimanche**? Si oui, comment, si non pourquoi ...

Q3. Si **E** est une variable de type **Ensemble** alors on considère que l'élément **x** est présent dans **E** si **E[x]** vaut *vrai* et qu'il est absent de **E** sinon. On donne ainsi le code C de la fonction suivante :

```
int estPresent (Ensemble E, Element x)
  // vaut vrai si et seulement si x est un element de E
  return E[x] != 0 ;
}
```

Ecrivez en C le code de l'action suivante :

```
Inserer : action (la donnee-resultat E : un Ensemble, la donnee x : un Element) ;
{ etat final : l'ensemble E contient l'element x }
```

Q4. Ecrivez le code de l'action suivante :

```
Union : action (le resultat E : un Ensemble E,
                la donnee E1 : un Ensemble, la donnee E2 : un Ensemble) ;
{ etat final : E contient l'union de E1 et E2 }
```

Q5. Ecrivez le code de la fonction suivante :

```
estInclus: fonction (E1 : un Ensemble, E2 : un Ensemble) -> booleen ;
{ renvoie vrai si et seulement si E1 est inclus dans E2 }
```

Exercice 3 (~ 8 points)

En langage C les chaînes de caractères sont représentées dans des tableaux de caractères (de type **char**) et se terminent toujours par le caractère **'\0'** (marque de fin de chaîne). On donne les spécifications de trois fonctions inspirées de la bibliothèque **<string.h>** :

- `int strlen (char s[]);`
Renvoie le nombre de caractères de la chaîne `s`, sans compter le caractère `'\0'`.
- `char * strchr (char s[], char c);`
Renvoie un pointeur sur la **première** occurrence du caractère `c` dans la chaîne `s` (terminée par `'\0'`) et renvoie le pointeur `NULL` si `c` n'est pas présent dans `s`. Par exemple si `s` contient la chaîne "bonjour" alors `strchr(s, 'o')` renvoie l'adresse de `s[1]` et `strchr(s, 'n')` renvoie `NULL`.
- `char *strpbrk(char s[], char a[]);`
Renvoie un pointeur sur la **première** occurrence dans la chaîne `s` de n'importe quel caractère présent dans la chaîne `a` (ces deux chaînes étant terminées par `'\0'`), et renvoie `NULL` si aucun caractère de `a` n'est présent dans `s`. Par exemple si `s` contient la chaîne "bonjour" et `a` la chaîne "nord" alors `strpbrk(s, a)` renvoie l'adresse de `s[1]`¹.

Q1. On considère la fonction `f()` dont le code est donné ci-dessous :

```
char * f(char *dest, const char *src, unsigned int n)
{
    unsigned int dest_len = strlen(dest);
    unsigned int i;

    for (i = 0 ; i < n && src[i] != '\0' ; i++) {
        dest[dest_len + i] = src[i];
    } ;
    dest[dest_len + i] = '\0';
    return dest;
}
```

Décrivez en quelques lignes la spécification de cette fonction (que renvoie t-elle en fonction des valeurs de ses trois paramètres, quels sont ses paramètres qui sont modifiés et comment).

Q2. Complétez le programme principal ci-dessous (déclaration de la variable `z` et paramètres manquants de la fonction `f()`) afin qu'il affiche à l'écran la chaîne "M2 CCI". Vous pouvez ajouter d'autres déclarations de variables si besoin.

```
int main(){
    // completer la declaration de la variable z
    z = f(..., "CCI 20-21", ...) // completer les parametres manquants
    printf ("%s\n", z) // affiche "M2 CCI"
    return 0 ;
}
```

Q3. Ecrivez le code de la fonction `strpbrk` en utilisant la fonction `strchr`. Donnez des exemples d'appels à cette fonction (et les résultats attendus) que vous pourriez utiliser pour *tester* si votre solution est correcte.

1. et non l'adresse de `s[2]` ou de `s[6]` car ce ne sont pas les premiers caractères de `s` présents dans `a`