

Nom(s):

Prénom(s):

Groupe:

# CORRECTION

## Contrôle continu

### Langage C++ et environnement

Durée : 30 minutes

Documents autorisés. Barème donné à titre indicatif.

#### Exercice 1 (4 pts.)

1. Décrivez succinctement et précisément ce qu'est un *fichier objet* ;

▽ **Correction**

Un fichier objet est la version compilée en code machine d'une unité de compilation, sans le code des fonctions et variables externes.

2. On considère le programme suivant :

```
double *create_buffer(size_t sz) {  
    double *x = new double[sz] {1.0};  
    return &(x[0]);  
}
```

```
int main(void) {  
    double &res = *create_buffer(10);  
    cout << res << endl;  
}
```

- (a) Quel est le résultat de l'exécution du programme (justifier la réponse)?

▽ **Correction**

Le programme affiche **1** sur la sortie standard : La fonction `create_buffer()` crée un tableau de 10 cases initialisées à 1.0 et retourne l'adresse de la première case; on déréférence cette adresse et `res` est un alias pour cette case.

- (b) Où est stockée la variable `x` (justifier la réponse)?

▽ **Correction**

La variable `x` est stockée dans la pile car c'est une variable locale

- (c) Où se trouve la valeur référencée par `res` (justifier la réponse)?

▽ **Correction**

La variable `res` est une référence sur une zone de mémoire allouée dynamiquement sur le tas.

3. Donner la définition du type C++ `fleur_t` permettant d'écrire le code suivant :

```
int main(void) {  
    vector<unsigned int> quantites { 0,  
                                     25, // Iris  
                                     13, // Bleuets  
                                     0,  
                                     98 }; // Narcisses
```

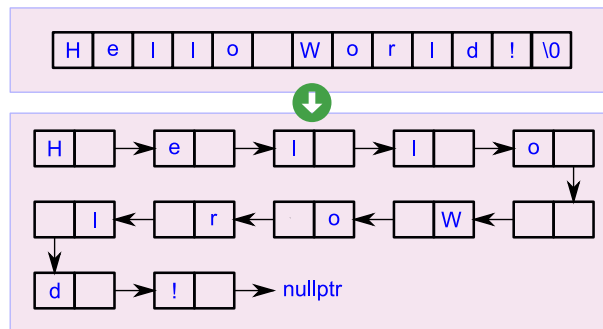
```
    cout << quantites[narcisse] << endl;  
    cout << quantites[iris] << endl;  
    cout << quantites[bleuet] << endl;  
}
```

### ▽ Correction

```
enum fleur_t { iris = 1,
               bleuet,
               narcisse = 4
};
```

### Exercice 2 (6 pts.)

On souhaite écrire une fonction pour transformer une chaîne de caractères représentée par un tableau en une liste simplement chaînée de caractères :



On a le code :

```
int main(void)
{
    str_t *chaine = creer_chaine("Hello World!");
}
```

1. Donner le code du type `str_t`, liste simplement chaînée de caractères;

### ▽ Correction

```
struct str_t {
    char c;
    str_t *next;
};
```

2. Donner le code de la fonction `str_t *creer_chaine(const char *s)` prenant en entrée une chaîne de caractères et retournant la liste chaînée de caractères correspondante.

### ▽ Correction

```
str_t *creer_chaine(const char* s)
{
    if (*s == '\0') {
        return nullptr;
    }
    return new str_t{*s, creer_chaine(s+1)};
}
```