

corrige

November 29, 2024

1 Corrigé 29/11/24

1.1 Sujet 1

1.1.1 Exercice 1

1. On utilise un variant de boucle : $x + y$

A chaque étape, le variant décroît strictement

2. On utilise l'indice pour avoir l'invariant de boucle, on remarque aussi que $\text{pgcd}(0, x) = x$

1.1.2 Exercice 2

```
[9]: #include <stdio.h>
#include <string.h>

void javanais(char* mot, char* buffer){
    int n = strlen(mot);
    int j = 0;
    for(int i = 0; i < n; i++){
        if(mot[i] == 'a' || mot[i] == 'e' || mot[i] == 'i' || mot[i] == 'o' || 
        ↪ mot[i] == 'u'){
            buffer[j] = 'a';
            j++;
            buffer[j] = 'v';
            j++;
        }
        buffer[j] = mot[i];
        j++;
    }
}

int main(){
    char* mot = "jour";
    char buffer[100] = {0};
    javanais(mot, buffer);
    printf("%s -> %s", mot, buffer);
    return 0;
}
```

jour -> javoavur

```
[11]: #include <stdio.h>
#include <string.h>
#include <stdbool.h>

void javanaïs2(char* mot, char* buffer){
    int n = strlen(mot);
    int j = 0;
    bool prev_v = false;
    for(int i = 0; i<n; i++){
        if(!prev_v && (mot[i]=='a' || mot[i]=='e' || mot[i]=='i' || mot[i]=='o' ||
↪ || mot[i]=='u')){
            buffer[j] = 'a';
            j++;
            buffer[j] = 'v';
            j++;
            prev_v = true;
        } else {
            prev_v = false;
        }
        buffer[j] = mot[i];
        j++;
    }
}

int main(){
    char* mot = "jour";
    char buffer[100] = {0};
    javanaïs2(mot,buffer);
    printf("%s -> %s", mot, buffer);
    return 0;
}
```

jour -> javour

1.2 Sujet 2

1.2.1 Exercice 1

```
[ ]: bool recherche_dichotomique(int* t, int n, int x){
    int debut = 0, fin = n;
    while (debut < fin){
        int milieu = (debut + fin) / 2;
        if (t[milieu] == x){
            return true;
        } else if (t[milieu] < x){
            debut = milieu + 1;
        }
    }
}
```

```

    } else {
        fin = milieu;
    }
}
return false;
}

```

2. On utilise le variant fin-debut. La correction est dans le cours

1.2.2 Exercice 2

1.

x = 5; y = 7; z = 9

x = 5; y = 7; z = 9

x = 7; y = 7; z = 9

x = 5; y = 9; z = 9

1.2.3 Exercice 3

```

[12]: bool est_palindrome(char* mot){
    int n = strlen(mot);
    for(int i = 0; i<n/2; i++){
        if(mot[i]!=mot[n-i]) return false
    }
    return true;
}

```

/tmp/tmp1pf848jb.c:1:1: erreur: nom de type « bool » inconnu

```

1 | bool est_palindrome(char* mot){
  | ^~~~

```

/tmp/tmp1pf848jb.c:1:1: note: « bool » est défini dans l'en-tête

« <stdbool.h> » ; ceci peut probablement être corrigé en ajoutant « #include <stdbool.h> »

```

+++ |+#include <stdbool.h>

```

```

1 | bool est_palindrome(char* mot){

```

/tmp/tmp1pf848jb.c: Dans la fonction « est_palindrome »:

/tmp/tmp1pf848jb.c:2:13: erreur: déclaration implicite de la fonction « strlen »
[-Wimplicit-function-declaration]

```

2 |     int n = strlen(mot);
  |               ^~~~~~

```

/tmp/tmp1pf848jb.c:1:1: note: incluez « <string.h> » ou fournissez une déclaration de « strlen »

```

+++ |+#include <string.h>

```

```

1 | bool est_palindrome(char* mot){

```

/tmp/tmp1pf848jb.c:2:13: attention: déclaration implicite incompatible pour la fonction interne « strlen » [-Wbuiltin-declaration-mismatch]

```

2 |     int n = strlen(mot);

```

```

|           ^~~~~~
/tmp/tmp1pf848jb.c:2:13: note: incluez « <string.h> » ou fournissez une
déclaration de « strlen »
/tmp/tmp1pf848jb.c:4:37: erreur: « false » non déclaré (première utilisation
dans cette fonction)
    4 |         if(mot[i]!=mot[n-i]) return false
      |                                     ^~~~~~
/tmp/tmp1pf848jb.c:4:37: note: « false » est défini dans l'en-tête
« <stdbool.h> » ; ceci peut probablement être corrigé en ajoutant « #include
<stdbool.h> »
/tmp/tmp1pf848jb.c:4:37: note: chaque identificateur non déclaré est rapporté
une seule fois pour chaque fonction dans laquelle il apparaît
/tmp/tmp1pf848jb.c:4:42: erreur: expected « ; » before « } » token
    4 |         if(mot[i]!=mot[n-i]) return false
      |                                     ^
      |                                     ;
    5 |     }
      |     ~
/tmp/tmp1pf848jb.c:6:12: erreur: « true » non déclaré (première utilisation dans
cette fonction)
    6 |     return true;
      |           ^~~~
/tmp/tmp1pf848jb.c:6:12: note: « true » est défini dans l'en-tête
« <stdbool.h> » ; ceci peut probablement être corrigé en ajoutant « #include
<stdbool.h> »
[C kernel] GCC exited with code 1, the executable will not be executed

```

```

[ ]: void est_palindrome(char* mot){
    int n = strlen(mot);
    for(int i = 0; i<n/2; i++){
        char tmp = mot[i];
        mot[i]=mot[n-i];
        mot[n-i] = tmp;
    }
}

```

1.3 Sujet 3

1.3.1 Exercice 1

```

[ ]: int mcCarthy91(int n) {
    if (n > 100) {
        return n - 10;
    } else {
        return mcCarthy91(mcCarthy91(n + 11));
    }
}

```

2. On utilise le variant $\max(0, 100 - n)$

3. Meme raisonnement

1.3.2 Exercice 2

```
[ ]: int estVide(File *f) {  
    return f->taille == 0;  
}
```

```
[ ]: int estPleine(File *f) {  
    return f->taille == TAILLE_MAX;  
}
```

```
[ ]: int enfiler(File *f, int valeur) {  
    if (estPleine(f)) {  
        return 0; // Erreur : la file est pleine  
    }  
    f->elements[f->fin] = valeur;  
    f->fin = (f->fin + 1) % TAILLE_MAX;  
    f->taille++;  
    return 1; // Succès  
}
```

```
[ ]: int defiler(File *f, int *valeur) {  
    if (estVide(f)) {  
        return 0; // Erreur : la file est vide  
    }  
    *valeur = f->elements[f->debut];  
    f->debut = (f->debut + 1) % TAILLE_MAX;  
    f->taille--;  
    return 1; // Succès  
}
```

```
[ ]: #include <stdio.h>  
  
int main() {  
    File file; // Initialisation de la file  
    file.debut = 0;  
    file.fin = 0;  
    file.taille = 0;  
  
    // Test des opérations  
    enfiler(&file, 5);  
    enfiler(&file, 10);  
    enfiler(&file, 15);  
  
    int valeur;  
    while (!estVide(&file)) {  
        defiler(&file, &valeur);  
    }  
}
```

```
        printf("Valeur défiler : %d\n", valeur);  
    }  
  
    return 0;  
}
```