



Examen d'Algorithmique et programmation C++ (DVE11)

Session Principale

Durée 1h30

UGEM

Exercice 1 [4 pts] :

On va Ecrire une fonction qui prend en paramètre un entier et retourne combien de fois il est divisible par 2, Par exemple : $24/2=12$ et $12/2 = 6$ et $6/2 = 3$ et 3 n'est pas divisible par 2

⇒ 24 est divisible par 2 trois fois.

1. Donnez son prototype.
2. Ecrire le code de la fonction.
3. Écrire également son programme de test (main).

Exercice 2 [3 pts] :

La suite de Fibonacci est définie comme suit :

- $F_0=F_1=1$
- $F_n=F_{n-2}+F_{n-1}$ pour $n>2$

1. Écrire une fonction récursive qui prend comme paramètre un entier n , et qui calcule et retourne F_n .
2. Tester la fonction avec une valeur aléatoire de l'intervalle $[0,100]$

Exercice 3 [4 pts] :

1. Écrire une fonction récursive qui prend comme paramètre une chaîne de caractères, puis elle remplace les caractères espaces par `_`. Par exemple, si la chaîne est "Bonne chance au devoir", La fonction devrait renvoyer la chaîne "Bonne_chance_au_devoir"

Voici le prototype de la fonction :

`string fonction (string phrase, int i)`

2. Tester votre programme avec une phrase saisie par l'utilisateur

Exercice 4 [3 pts] :

Une matrice symétrique est une matrice carrée qui est égale à sa propre transposée, c'est-à-dire telle que $a_{i,j} = a_{j,i}$ pour tous i et j compris entre 0 et $n-1$.

1. Ecrire une fonction `Est_Symétrique()` qui prend comme paramètres une matrice carrée M d'ordre n et affiche si M est symétrique ou non.
2. Tester la fonction avec la matrice :

$$M = \begin{pmatrix} 1 & 4 & 6 \\ 4 & 5 & 9 \\ 6 & 9 & 2 \end{pmatrix}$$

Exercice 5 [6 pts] :

1. Ecrire une fonction `Initialiser()` permettant de remplir un tableau T et sa taille n avec des valeurs réels saisie par l'utilisateur.
2. Ecrire une fonction `PremOccurrence ()` qui prend comme paramètre un tableau T et sa taille n et un réel x puis elle retourne la position de la première occurrence de x dans T , sinon elle retourne -1
3. Ecrire la fonction `DemOccurr ()` qui retourne la position de la dernière occurrence de x dans T .
4. En utilisant les deux fonctions précédentes écrire la fonction `UneOccurrence()` permettant de tester si le nombre d'occurrence x dans T est égal à 1.
5. Tester les fonctions.

Aucun document n'est autorisé. Les calculatrices ne sont pas autorisées. L'indentation des programmes est obligatoire.

Exercice 1 [4 pts]. Question de cours

1. C'est quoi un code ASCII ?
2. Convertir en binaire le nombre -180 en utilisant la méthode de complément à 2.
3. Expliquer le rôle des technologies Intel Hyper-threading et Turbo boost
4. C'est quoi la différence entre SRAM et DARM ?

Exercice 2 [3 pts] : Qu'affiche le programme C++ suivant :

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int i=1, j;
5     while(i <=5){
6         j= i%3;
7         switch(j) {
8             case 0: cout<<+j*3<<endl; break;
9             case 1: cout<<(i+j++) %2 <<endl;
10            case 2: cout<<(i+j) % 4 <<endl;
11        }
12        i++;
13    }
14    return 0;
15 }
```

Exercice 3 [4 pts].

Écrire un programme C++ qui demande à l'utilisateur de saisir une chaîne de caractères alphabétiques et d'au moins dix lettres. Le programme force l'utilisateur de continuer à refaire la saisie si la chaîne ne respecte pas les deux conditions. Votre programme affiche ensuite le nombre de caractères majuscules et minuscules de la chaîne.

Exercice 4 [4 pts].

Soit U_n la suite définie comme suit.

- $U_0 = \sqrt{2}$

- $U_1 = 1$
 - $U_n = 2 * U_{n-1} + U_{n-2}$, pour tout $n > 1$
1. Donner le prototype de la fonction $U(n)$ qui retourne U_n .
 2. Écrire le code C++ de la fonction.

Exercice 5 [5 pts].

Écrire une fonction qui prend comme paramètre un entier n et qui affiche le triangle de Pascal d'ordre n (voir ci-dessous le résultat d'affichage pour $n=5$).

Chaque ligne commence par 1 et se termine par 1. Un élément interne d'une ligne i est obtenu par la somme deux éléments de la ligne $i-1$: l'élément au-dessus et celui qui est à sa gauche. Indication : utiliser une matrice carrée d'entiers d'ordre n .

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```



3 of 5

le Mauritanie
ment Supérieuret de la Recherche Scientifique
Institut Supérieur du Numérique
Site : www.supnum.mrالجمهورية الإسلامية الموريتانية
وزارة التعليم العالي والبحث العلمي
المعهد العالي للرقمية
www.supnum.mr : الموقع الإلكتروني

Elément de module : [DEV11] Algorithmique et programmation C++

Contrôle de compétences N° 1

Durée : 1H30

Document non autorisé

UGEM

Exercice 1 : [5 Pts]

Étant donné une matrice carrée, écrire une fonction qui calcule la différence absolue entre les sommes de ses diagonales. Par exemple, la matrice carrée est représentée ci-dessous :

int M [3][3] = {5, 8, 9, 4, 3, 6, 6, 8, 8}

5 8 9

4 3 6

6 8 8

La diagonale de gauche à droite est $5+3+8=16$. La diagonale de droite à gauche est $9+3+6=18$. Leur différence absolue est de $|16-18|=2$.

Exercice 2 : [10 Pts]

1. Écrire la fonction Min qui prend comme paramètre un tableau T d'entiers et retourne le minimum de T.
2. Écrire la fonction Max qui prend comme paramètre un tableau T d'entiers et retourne le maximum de T.
3. Écrire la fonction Somme qui prend comme paramètre un tableau T d'entiers et retourne la somme des éléments de T.
4. Écrire une fonction qui prend comme paramètre un tableau d'entiers de 6 éléments. La fonction retourne les valeurs minimale et maximale qui peuvent être calculées en additionnant exactement 5 des six entiers.
NB : valeurs minimale = $\text{Somme}(T) - \text{Max}(T)$
valeurs maximale = $\text{Somme}(T) - \text{Min}(T)$
5. Tester la fonction avec un tableau dont les éléments sont donnés par l'utilisateur. Voici la sortie de votre programme :

Tableau = [1,3,5,7,9,4]

Addition Min 20

Addition Max : 28

Exercice 3 : [5 Pts]

La suite de Syracuse d'un nombre entier $N > 0$ est définie par récurrence, de la manière suivante : $U_0 = N$

$$u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair,} \\ 3u_n + 1 & \text{si } u_n \text{ est impair.} \end{cases}$$

La conjecture affirme que pour tout entier $N > 0$, il existe un indice r tel que $U_r = 1$.

1. Écrire une fonction récursive qui affiche le plus petit indice r t.q. $U_r = 1$.
2. Tester votre fonction

Examen de rattrapage d'Algorithmique et programmation C++ (DVE11)

UGEM

25/02/2022

Durée 1h30

Aucun document n'est autorisé. Les calculatrices ne sont pas autorisées. L'indentation des programmes est obligatoire.

Exercice 1 [5 pts]. Écrire une fonction `int U(int n)` qui renvoie la valeur U_n de la suite définie comme suit.

- $U_0=1$
- $U_n=2U_{n-1}+1$ si n est pair
- $U_n=2n-5$ si n est impair

Exercice 2 [5 pts]. Écrire une fonction qui prend comme paramètre une chaîne de caractères et qui retourne la chaîne avec une inversion des majuscules et minuscules. Voir les exemples ci-dessous :

AbCd → aBCd

aBcd → AbCD

Exercice 3 [5 pts].

1. Écrire une fonction `void tri(int T[], int n)` qui trie le tableau `T` dans un ordre croissant. Utilisez l'algorithme de tri à bulles vu en cours.
2. Écrire un programme qui demande à l'utilisateur de saisir un entier n et les valeurs d'un tableau `T` de n éléments (entiers). Le programme affiche le tableau avant et après tri.

Exercice 4 [5 pts]. Écrire un programme qui ajoute les points de fin de ligne qui manquent à un fichier texte. Votre programme parcourt un fichier nommé `texte_source.txt` et produit un fichier nommé `text_cible.txt`.

Examen de rattrapage Élément de module DEV11. Algorithmique et programmation C++ Les documents ne sont pas autorisés	Examen de rattrapage. Élément de module DEV11. Algorithmique et programmation C++. Les documents ne sont pas autorisés.
texte_source.txt	texte_cible.txt

Devoir d'Algorithmique et programmation C++ (DVE11)

UGEM

Durée 1h30

Exercice 1 [2 pts] :

1. C'est un quoi un langage machine ?
2. C'est un quoi la différence entre les langages interprétés et les langages compilés ?

Exercice 2 [4 pts] :

En utilisant une boucle pour écrire un algorithme qui demande à l'utilisateur un nombre puis il affiche la table de multiplication de ce nombre, présentée comme suit (par exemple si l'utilisateur saisie 5) le programme affiche :

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
...
5 x 10 = 50
```

Exercice 3 [3 pts] :

Qu'affiche le programme suivant :

```
#include <iostream>
using namespace std;
int main() {
    char c = 'B';
    int a = 0;
    int b = 0;
    switch(c){
        case 'A' : b = a++ + 2; break;
        case 'B' : b = a++ + 2;
        case 'C' : ++b;
        case 'D' : b++;break;
        default : a++;
    }
    cout<<a<<endl;
    cout<<b<<endl;
    return 0;
}
```

Exercice 4 [3 pts] :

Un entier est dit premier s'il a exactement deux diviseurs différents 1 et lui-même, Ecrire un programme C++ qui lit un entier puis détermine s'il est premier ou non.

Exercice 5 [4 pts] :

Ecrire un programme C++ qui demande à l'utilisateur un nombre entier et l'affiche inversé, par exemple si l'entrée est 12345 on doit afficher l'entier 54321.

Exercice 6 [4 pts] :

Nous souhaitons écrire un programme C++ permettant de saisir une suite des notes entre 0 et 20 et calculer la moyenne des notes, la meilleure note et la mauvaise note, le programme s'arrête si l'utilisateur saisie une note supérieure à 20 ou une note inférieure à 0.

1. Citez les variables nécessaires.
2. Ecrire le code d'une boucle permettant de saisir une suite des notes entre 0 et 20, elle s'arrête si l'utilisateur saisie une note supérieure à 20 ou une note inférieure à 0.
3. Ajoutez le code permettant de calculer la moyenne des notes, la meilleure note et la mauvaise note, Commentez vos lignes.