Correction TD2 C++

**Fait par :**

-Doha Karim

IDAI

**Ex1 :**

#include <iostream>

using namespace std;

void afficherCompteurAppels() {

static int compteurAppels = 0;

compteurAppels++;

cout << "Appel numéro " << compteurAppels <<endl;

}

int main() {

afficherCompteurAppels();

afficherCompteurAppels();

afficherCompteurAppels();

afficherCompteurAppels();

return 0;

}

**Ex2 :**

#include <iostream>

using namespace std;

bool MultipleDeux(int nbre) {

return nbre % 2 == 0;

}

bool MultipleTrois(int nbre) {

return nbre % 3 == 0;

}

int main() {

int nbre;

cout << "Donnez un entier : ";

cin >> nombre;

if (MultipleDeux(nbre)) {

cout << "Il est pair" << endl;

}

if (MultipleDeTrois(nbre)) {

cout << "Il est multiple de 3" << endl;

}

if (MultipleDeux(nbre) && MultipleTrois(nbre)) {

cout << "Il est divisible par 6" << endl;

}

return 0;

}

**Ex3 :**

#include <iostream>

using namespace std;

int main() {

int tab[10];

for (int i = 0; i < 10; i++) {

cout << "Entrez un nombre : ";

cin >> tab[i];

}

int plusGrand = tab[0];

int plusPetit = tabl[0];

for (int i = 1; i < 10; i++) {

if (tableau[i] > plusGrand) {

plusGrand = tab[i];

}

if (tableau[i] < plusPetit) {

plusPetit = tab[i];

}

}

cout << "Le plus grand nombre est : " << plusGrand << endl;

cout << "Le plus petit nombre est : " << plusPetit << endl;

return 0;

}

B\_

#include <iostream>

using namespace std;

int main() {

int tab [10];

int\* p = tab;

for (int i = 0; i < 10; i++) {

cout << "Entrez un nombre : ";

cin >> \*p;

p++;

}

p = tableau;

int\* plusGrand = p;

int\* plusPetit = p;

for (int i = 1; i < 10; i++) {

p++;

if (\*p> \*plusGrand) {

plusGrand = p;

}

if (\*p< \*plusPetit) {

plusPetit = p;

}

}

cout << "Le plus grand nombre est : " << \*plusGrand << endl;

cout << "Le plus petit nombre est : " << \*plusPetit << endl;

return 0;

}

**Ex4 :**

#include <iostream>

using namespace std;

int main() {

int taille;

int\* premierTableau = nullptr;

int\* carresTableau = nullptr;

cout << "Entrez la taille du tableau : ";

cin >> taille;

premierTableau = new int[taille];

for (int i = 0; i < taille; i++) {

cout << "Entrez un nombre entier : ";

cin >> premierTableau[i]; }

carresTableau = new int[taille];

for (int i = 0; i < taille; i++) {

carresTableau[i] = premierTableau[i] \* premierTableau[i];

}

cout << "Les carrés des nombres sont : ";

for (int i = 0; i < taille; i++) {

cout << carresTableau[i] << " ";

}

cout << endl;

delete[] premierTableau;

delete[] carresTableau;

return 0;

}

**Ex5 :**

#include <iostream>

using namespace std;

int main() {

int a = 42; int &ref\_a = a;

int \*p\_a = &a;

cout << "a = " << a << endl;

cout << "Adresse de a : " << &a << endl;

cout << "Valeur pointée par p\_a : " << \*p\_a << endl;

cout << "Adresse de ref\_a : " << &ref\_a << endl;

cout << "Valeur de ref\_a : " << ref\_a << endl;

return 0;

}

**Ex6 :**

#include <iostream>

using namespace std;

void incrementer(int \*p) {

(\*p)++;

}

void permuter(int \*a, int \*b) {

int C = \*a;

\*a = \*b;

\*b = C;

}

void permuter(int &a, int &b) {

int C = a;

a = b;

b = C;

}

int main() {

int num1 = 5;

int num2 = 10;

cout << "Valeur de num1 avant l'incrémentation : " << num1 << endl;

incrementer(&num1);

cout << "Valeur de num1 après l'incrémentation : " << num1 << endl;

cout << "Avant la permutation : num1 = " << num1 << ", num2 = " << num2 << endl;

permuter(&num1, &num2);

cout << "Après la permutation (adresse) : num1 = " << num1 << ",

num2 = " << num2 << endl;

permuter(num1, num2);

cout << "Après la permutation (référence) : num1 = " << num1 << ", num2 = " << num2 << endl;

return 0;

}

**Ex7 :**

#include <iostream>

using namespace std;

int main() {

int tab[10];

int temp;

int nbreEchanges;

for (int i = 0; i < 10; i++) {

cout << "Entrez un entier: ";

cin >> tab[i];

}

do {

nbreEchanges = 0;

for (int i = 0; i < 9; i++) {

if (tab[i] > tabl[i + 1]) {

temp= tab[i];

tab[i] = tab[i + 1];

tab[i + 1] = temp;

nbreEchanges++;

}

}

} while (nbreEchanges > 0);

cout << "Tableau croissant : ";

for (int i = 0; i < 10; i++) {

cout << tab[i] << " ";

}

cout << endl;

return 0;

}

Ex11 :

#include <iostream>

#include <cmath>

using namespace std;

class Vecteur3D {

private:

float x;

float y;

float z;

public:

Vecteur3D(float x = 0.0, float y = 0.0, float z = 0.0) : x(x), y(y), z(z) {}

void afficher() {

cout << "(" << x << ", " << y << ", " << z << ")";

}

Vecteur3D addition(const Vecteur3D& autre) {

return Vecteur3D(x + autre.x, y + autre.y, z + autre.z);

}

float produitScalaire(const Vecteur3D& autre) {

return x \* autre.x + y \* autre.y + z \* autre.z;

}

bool coincide(const Vecteur3D& autre) {

return (x == autre.x && y == autre.y && z == autre.z);

}

float norme() {

return sqrt(x \* x + y \* y + z \* z);

}

static Vecteur3D normax(const Vecteur3D& v1, const Vecteur3D& v2) {

return (v1.norme() >= v2.norme()) ? v1 : v2;

}

};

int main() {

Vecteur3D v1(1.0, 2.0, 3.0);

Vecteur3D v2(2.0, 3.0, 4.0);

cout << "Vecteur 1 : ";

v1.afficher();

cout << endl;

cout << "Vecteur 2 : ";

v2.afficher();

cout << endl;

cout << "Somme de v1 et v2 : ";

Vecteur3D somme1 = v1.addition(v2);

somme1.afficher();

cout << endl;

cout << "Produit scalaire de v1 et v2 : " << v1.produitScalaire(v2) << endl;

cout << "v1 et v2 coincident : " << (v1.coincide(v2) ? "Oui" : "Non") << endl;

cout << "Norme de v1 : " << v1.norme() << endl;

Vecteur3D maxNorme = Vecteur3D::normax(v1, v2);

cout << "Le vecteur avec la plus grande norme est : ";

maxNorme.afficher();

cout << endl;

return 0;

}

Ex12 :

#include <iostream>

using namespace std;

class Test {

public:

static int compteurAppels; void call() {

compteurAppels++;

cout << "Fonction call a été appelée " << compteurAppels << " fois." << endl;

}

};

int Test::compteurAppels = 0;

int main() {

Test objetTest;

objetTest.call();

objetTest.call();

objetTest.call();

Test autreObjetTest;

autreObjetTest.call();

return 0;

}

**Ex17 :**

#include <iostream>

using namespace std;

class DateTimeExtractor {

private:

string dateHeure;

public:

DateTimeExtractor(string dh) : dateHeure(dh) {}

void extractAndDisplay() {

if (dateHeure.length() != 12) {

cout << "Format de date et d'heure incorrect." << endl;

return;

}

string jour = dateHeure.substr(0, 2);

string mois = dateHeure.substr(2, 2);

string annee = dateHeure.substr(4, 4);

string heure = dateHeure.substr(8, 2);

string minute = dateHeure.substr(10, 2);

cout << "Jour : " << jour << endl;

cout << "Mois : " << mois << endl;

cout << "Année : " << annee << endl;

cout << "Heure : " << heure << endl;

cout << "Minute : " << minute << endl;

}

};

int main() {

string dateHeure;

cout << "Entrez la date et l'heure (JJMMAAAAHHNN) : ";

cin >> dateHeure;

DateTimeExtractor extractor(dateHeure);

extractor.extractAndDisplay();

return 0;

}