Exercice Algo / Allegro C++

Série 1

Exercice 3 :

#include <iostream>

int main() {

int n;

float x;

std::cout << "Donnez un entier et un flottant: " << std::endl;

std::cin >> n >> x;

std::cout << "Le produit de " << n << " par " << x << " est " << n \* x << std::endl;

return 0;

}

Exercice 4 :

#include <iostream>

// échange deux entiers en utilisant les pointeurs (comme en C)

void echangerParAdresse(int\* a, int\* b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main()

{

int x = 5, y = 10;

std::cout << "Avant l'échange: x = " << x << ", y = " << y << std::endl;

echangerParAdresse(&x, &y);

std::cout << "Après l'échange: x = " << x << ", y = " << y << std::endl;

return 0;

}

#include <iostream>

void echangerParReference(int& a, int& b)

{

int temp = a;

a = b;

b = temp;

}

int main()

{

int x = 5, y = 10;

std::cout << "Avant l'échange: x = " << x << ", y = " << y << std::endl;

echangerParReference(x, y);

std::cout << "Après l'échange: x = " << x << ", y = " << y << std::endl;

return 0;

}

Exercice 5 :

#include <iostream>

struct essai {

int n;

float x;

};

void raz\_Adresse(essai\* e) {

e->n = 0;

e->x = 0.0f;

}

int main() {

essai e = {5, 10.5f};

std::cout << "Avant RAZ (par adresse) : n = " << e.n << ", x = " << e.x << std::endl;

raz\_Adresse(&e);

std::cout << "Après RAZ (par adresse) : n = " << e.n << ", x = " << e.x << std::endl;

return 0;

}

#include <iostream>

struct essai {

int n;

float x;

};

void raz\_Reference(essai& e) {

e.n = 0;

e.x = 0.0f;

}

int main() {

essai e = {5, 10.5f};

std::cout << "Avant RAZ (par référence) : n = " << e.n << ", x = " << e.x << std::endl;

raz\_Reference(e);

std::cout << "Après RAZ (par référence) : n = " << e.n << ", x = " << e.x << std::endl;

return 0;

}