

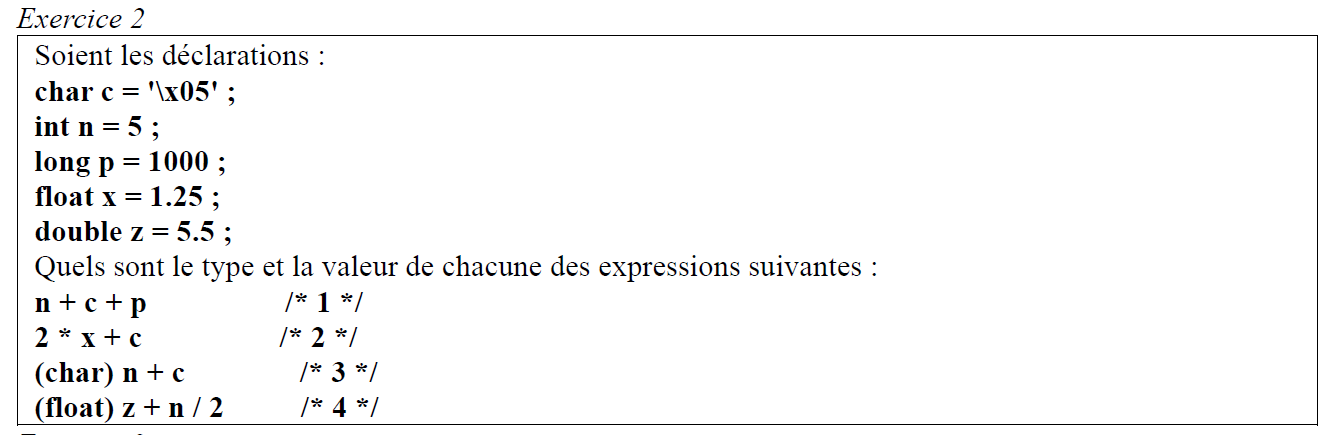
Tout les expressions suivantes sont de type int.

/\*1\*/ p + 3 = 13

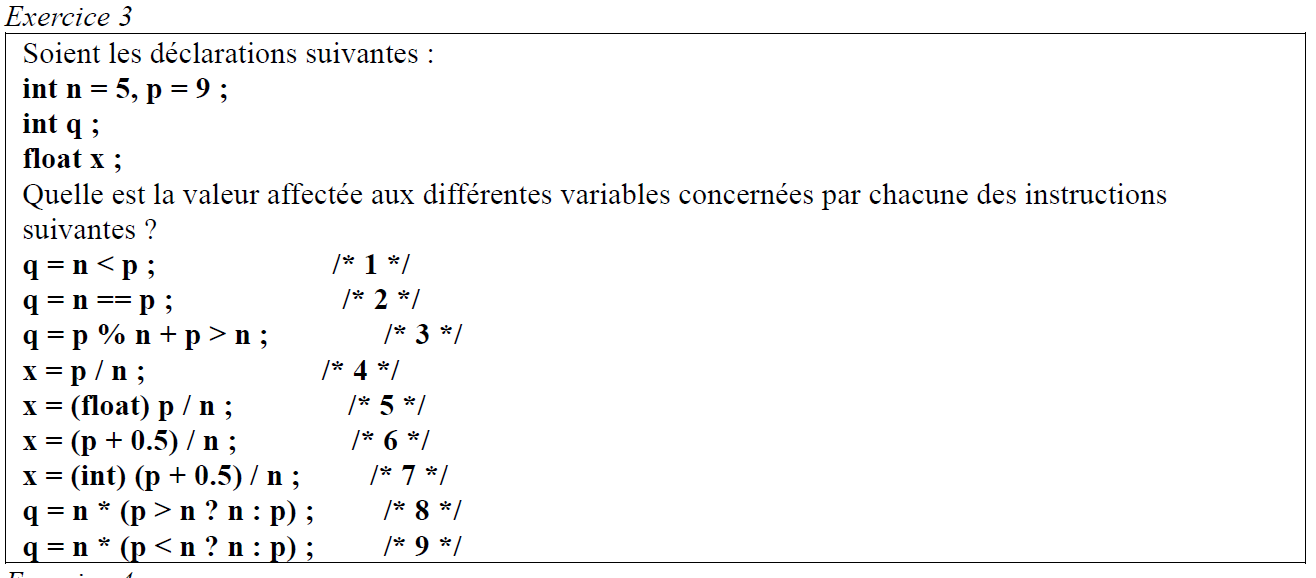
/\*2\*/ c + 1 = 2

/\*3\*/ p + c = 11

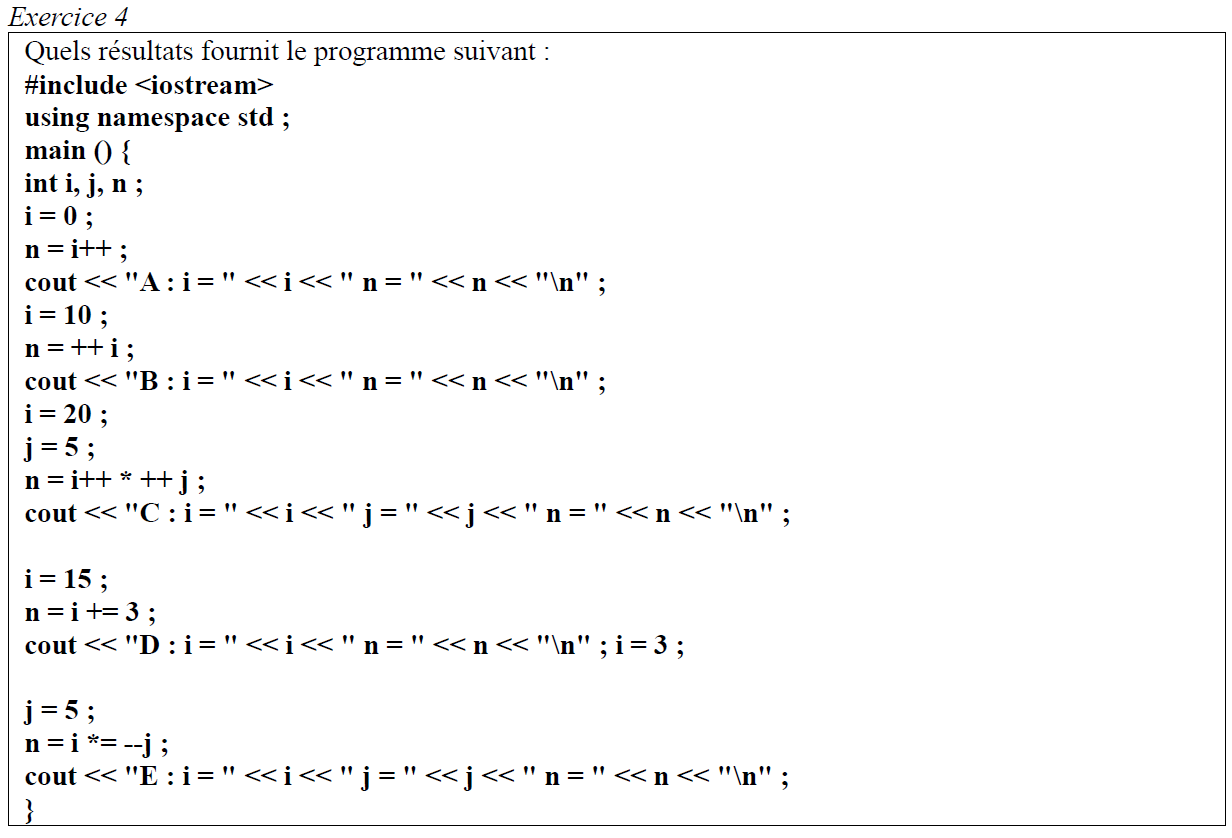
/\*4\*/ 3\*p + 5\*c = 35



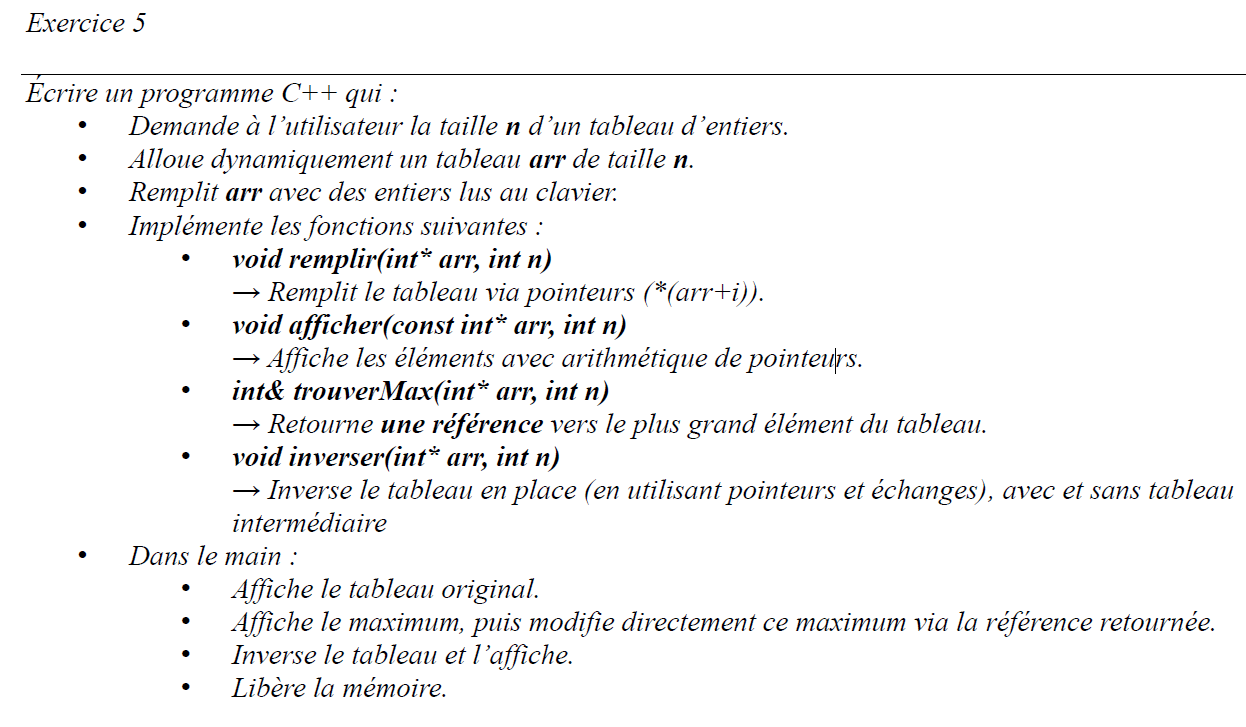
|  |  |  |
| --- | --- | --- |
| **Expression** | **Type** | **Valeur** |
| **/\*1\*/** | **int** | **1010** |
| **/\*2\*/** | **Float** | **7.5** |
| **/\*3\*/** | **Int** | **10** |
| **/\*4\*/** | **Float** | **7.5** |



|  |  |  |
| --- | --- | --- |
| **/\*1\*/ q = 1** | **/\*2\*/ q = 0** | **/\*3\*/ q = 1** |
| **/\*4\*/ x = 1** | **/\*5\*/ x = 1.8** | **/\*6\*/ x = 1.9** |
| **/\*7\*/ x = 1** | **/\*8\*/ q = 25** | **/\*9\*/ q = 45** |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A : i = 1 n = 0 | B : i = 11 n = 11 | C : i = 21 j = 6 n = 120 | D : i = 18 n = 18 | E : i = 12 j = 4 n = 12 |



#include <iostream>

using namespace std;

void remplir(int\* arr, int n) {

    for (int i = 0; i < n; i++) {

        cout << "arr[" << i << "] = ";

        cin >> \*(arr + i);

    }

}

void afficher(const int\* arr, int n) {

    for (int i = 0; i < n; i++)

        cout << \*(arr + i) << " ";

    cout << endl;

}

int& trouverMax(int\* arr, int n) {

    int maxIndex = 0;

    for (int i = 1; i < n; i++)

        if (\*(arr + i) > \*(arr + maxIndex))

            maxIndex = i;

    return \*(arr + maxIndex);

}

void inverser(int\* arr, int n) {

    for (int i = 0; i < n / 2; i++) {

        int temp = \*(arr + i);

        \*(arr + i) = \*(arr + n - 1 - i);

        \*(arr + n - 1 - i) = temp;

    }

}

int main() {

    int n;

    cout << "Taille du tableau: ";

    cin >> n;

    int\* arr = new int[n];

    remplir(arr, n);

    cout << "\nTableau original: ";

    afficher(arr, n);

    int& maxVal = trouverMax(arr, n);

    cout << "Max = " << maxVal << endl;

    maxVal += 10;

    cout << "Après modification du max: ";

    afficher(arr, n);

    inverser(arr, n);

    cout << "Après inversion: ";

    afficher(arr, n);

    delete[] arr;

    return 0;

}