**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

---**o0o**---

****

**BÀI TẬP LAB MÔN CĐ TCDL**

**BÀI TẬP 1**

Giảng viên hướng dẫn : **VŨ QUỐC HOÀNG**

Sinh viên thực hiện: **CAO QUỐC VIỆT**

MSSV: **22810218**

Lớp **: CN2022/2**

Khoá  **: 2022/2**

***TP. Hồ Chí Minh, tháng 02 năm 2025***

**Bài làm**

# **Bài 1.A: Tính tổ hợp bằng giai thừa**

Code:

#include <iostream>

using namespace std;

// Tinh giai thua

unsigned long long factorial(int n) {

    unsigned long long result = 1;

    for(int i = 2; i <= n; i++) {

        result \*= i;

    }

    return result;

}

// Tinh to hop theo cong thuc giai thua

unsigned long long C(int n, int k) {

    if (k > n) return 0;

    return factorial(n) / (factorial(k) \* factorial(n - k));

}

int main() {

    // Bo du lieu 1

    int n1 = 5, k1 = 2;

    cout << "C(" << n1 << "," << k1 << ") = " << C(n1, k1) << endl;

    // Bo du lieu 2

    int n2 = 6, k2 = 3;

    cout << "C(" << n2 << "," << k2 << ") = " << C(n2, k2) << endl;

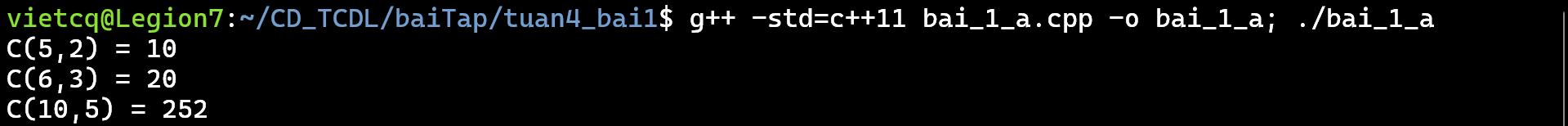
    // Bo du lieu 3

    int n3 = 10, k3 = 5;

    cout << "C(" << n3 << "," << k3 << ") = " << C(n3, k3) << endl;

    return 0;

}

Output:  


# **Bài 1.B: Tính tổ hợp không dùng giai thừa**

Code:

#include <iostream>

using namespace std;

// Tinh to hop khong dung giai thua

unsigned long long C(int n, int k) {

    if (k > n) return 0;

    if (k > n/2) k = n-k;  // Toi uu hoa

    unsigned long long result = 1;

    for(int i = 0; i < k; i++) {

        result = result \* (n - i) / (i + 1);

    }

    return result;

}

int main() {

    // Bo du lieu 1

    int n1 = 5, k1 = 3;

    cout << "C(" << n1 << "," << k1 << ") = " << C(n1, k1) << endl;

    // Bo du lieu 2

    int n2 = 7, k2 = 2;

    cout << "C(" << n2 << "," << k2 << ") = " << C(n2, k2) << endl;

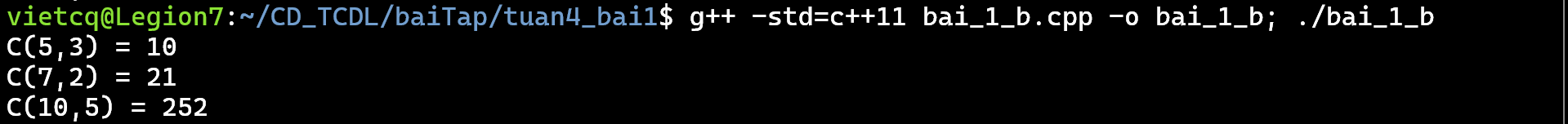
    // Bo du lieu 3

    int n3 = 10, k3 = 5;

    cout << "C(" << n3 << "," << k3 << ") = " << C(n3, k3) << endl;

    return 0;

}

Output:  


# **Bài 1.C: Tính tổ hợp bằng tam giác Pascal**

Code:

#include <iostream>

using namespace std;

// Tinh to hop dung tam giac Pascal

unsigned long long C(int n, int k) {

    if (k > n) return 0;

    unsigned long long C[100][100] = {0};

    for(int i = 0; i <= n; i++) {

        C[i][0] = 1;

        for(int j = 1; j <= i; j++) {

            C[i][j] = C[i-1][j-1] + C[i-1][j];

        }

    }

    return C[n][k];

}

int main() {

    // Bo du lieu 1

    int n1 = 4, k1 = 2;

    cout << "C(" << n1 << "," << k1 << ") = " << C(n1, k1) << endl;

    // Bo du lieu 2

    int n2 = 8, k2 = 3;

    cout << "C(" << n2 << "," << k2 << ") = " << C(n2, k2) << endl;

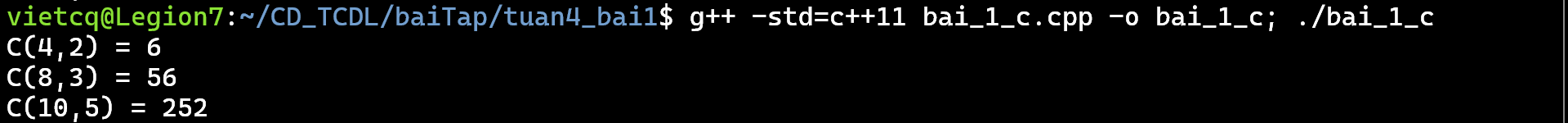
    // Bo du lieu 3

    int n3 = 10, k3 = 5;

    cout << "C(" << n3 << "," << k3 << ") = " << C(n3, k3) << endl;

    return 0;

}

Output:  


# **Bài 1.D: So sánh các phương pháp tính tổ hợp**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phương pháp** | **Công thức tính** | **Độ phức tạp** | **Nhận xét** |
| Giai thừa | C(n, k) = n! / [k! × (n – k)!] | O(n) | Dễ implement, chạy nhanh với n nhỏ, nhưng dễ tràn số khi n lớn. |
| Không dùng giai thừa | C(n, k) = (n/1) × [(n – 1)/2] × … × [(n – k + 1)/k] | O(k) | Chia số để tránh tràn số, chạy nhanh nhất trong ba phương pháp. |
| Tam giác Pascal | C(n, k) = C(n – 1, k – 1) + C(n – 1, k) | O(n·k) | Khó tràn số, nhưng chậm hơn phương pháp không dùng giai thừa khi n lớn. |

# **Bài 3.5.1: Làm tròn số thực**

Code:

#include <iostream>

#include <cmath>

using namespace std;

// Ham lam tron so thuc 'x' theo 'pos' chu so thap phan

double round\_num(double x, int pos) {

    double factor = pow(10.0, pos);

    return round(x \* factor) / factor;

}

int main() {

    // Bo du lieu 1: x = 3.14159, pos = 0

    double x1 = 3.14159;

    cout << "Bo du lieu 1 => x = 3.14159, pos = 0 => "

         << round\_num(x1, 0) << endl;

    // Bo du lieu 2: x = 2.71828, pos = -1

    double x2 = 2.71828;

    cout << "Bo du lieu 2 => x = 2.71828, pos = -1 => "

         << round\_num(x2, -1) << endl;

    // Bo du lieu 3: x = 1.99999, pos = 2

    double x3 = 1.99999;

    cout << "Bo du lieu 3 => x = 1.99999, pos = 2 => "

         << round\_num(x3, 2) << endl;

    // Bo du lieu 4: x = -3.14159, pos = 0

    double x4 = -3.14159;

    cout << "Bo du lieu 4 => x = -3.14159, pos = 0 => "

         << round\_num(x4, 0) << endl;

    // Bo du lieu 5: x = -2.71828, pos = -1

    double x5 = -2.71828;

    cout << "Bo du lieu 5 => x = -2.71828, pos = -1 => "

         << round\_num(x5, -1) << endl;

    // Bo du lieu 6: x = 0.12345, pos = -2

    double x6 = 0.12345;

    cout << "Bo du lieu 6 => x = 0.12345, pos = -2 => "

         << round\_num(x6, -2) << endl;

    return 0;

}

Output:  
A screen shot of a computer

AI-generated content may be incorrect.

# **Bài 4.1.6: Tìm trung vị bằng Quickselect**

Code:

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

using namespace std;

// Ham phan vung Quickselect

int partition(vector<double>& arr, int low, int high) {

    double pivot = arr[high];

    int i = low - 1;

    for (int j = low; j < high; ++j) {

        if (arr[j] <= pivot) {

            ++i;

            swap(arr[i], arr[j]);

        }

    }

    swap(arr[i + 1], arr[high]);

    return i + 1;

}

// Thuat toan Quickselect

double quickselect(vector<double>& arr, int low, int high, int k) {

    if (low == high) return arr[low];

    int pivotIndex = partition(arr, low, high);

    if (k == pivotIndex) return arr[k];

    else if (k < pivotIndex) return quickselect(arr, low, pivotIndex - 1, k);

    else return quickselect(arr, pivotIndex + 1, high, k);

}

// Ham tinh trung vi

double median(vector<double>& arr, int n) {

    if (n % 2 == 1) {

        return quickselect(arr, 0, n - 1, n / 2);

    } else {

        double left = quickselect(arr, 0, n - 1, n / 2 - 1);

        double right = quickselect(arr, 0, n - 1, n / 2);

        return (left + right) / 2;

    }

}

int main() {

    // Bo du lieu 1

    {

        vector<double> x = {4.5, 1.2, 3.9, 2.7, 5.0};

        cout << "Trung vi: " << median(x, x.size()) << endl;

    }

    // Bo du lieu 2

    {

        vector<double> x = {10, 8, 6, 4, 2, 12};

        cout << "Trung vi: " << median(x, x.size()) << endl;

    }

    // Bo du lieu 3

    {

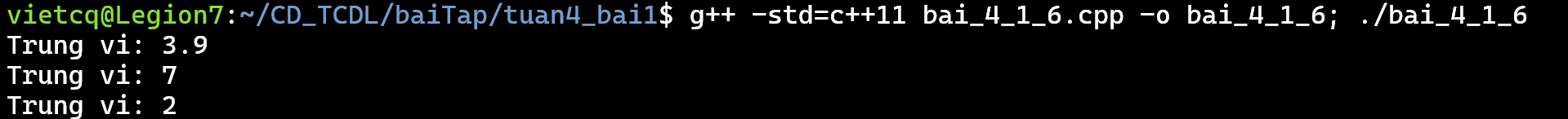
        vector<double> x = {1.0, 2.0, 3.0};

        cout << "Trung vi: " << median(x, x.size()) << endl;

    }

    return 0;

}

Output:  


# **Bài 4.2.8: Tìm “điểm yên ngựa”**

Code:

#include <iostream>

#include <vector>

using namespace std;

// Ham tim diem yen ngua

void tim\_yen\_ngua(const vector<vector<int>>& a, int m, int n) {

    for(int i = 0; i < m; i++) {

        for(int j = 0; j < n; j++) {

            bool la\_min\_hang = true;

            bool la\_max\_cot = true;

            // Kiem tra min hang & max cot

            for(int k = 0; k < n; k++) {

                if(a[i][k] < a[i][j]) {

                    la\_min\_hang = false;

                    break;

                }

            }

            for(int k = 0; k < m; k++) {

                if(a[k][j] > a[i][j]) {

                    la\_max\_cot = false;

                    break;

                }

            }

            if(la\_min\_hang && la\_max\_cot) {

                cout << "Diem yen ngua tai (" << i << "," << j << "): " << a[i][j] << endl;

            }

        }

    }

}

int main() {

    // Bo du lieu 1 (4x4) - Them so 1 o (0,0), so 0 o (1,0) de co diem yen ngua

    {

        vector<vector<int>> matrix\_1 = {

            {1,  2,  3,  4},

            {0,  5,  6,  7},

            {5,  6,  7,  8},

            {9, 10, 11, 12}

        };

        cout << "Ma tran 4x4:\n";

        for (int i = 0; i < 4; i++) {

            for (int j = 0; j < 4; j++) {

                cout << matrix\_1[i][j] << " ";

            }

            cout << endl;

        }

        tim\_yen\_ngua(matrix\_1, 4, 4);

    }

    // Bo du lieu 2 (5x5) - Dat 1 o hang dau, 0 o nhung hang ben duoi cung cot 0

    {

        vector<vector<int>> matrix\_2 = {

            {1,  2,  3,  4,  5},

            {0,  9,  9,  9,  9},

            {0,  9,  9,  9,  9},

            {0,  9,  9,  9,  9},

            {0,  9,  9,  9,  9}

        };

        cout << "Ma tran 5x5:\n";

        for (int i = 0; i < 5; i++) {

            for (int j = 0; j < 5; j++) {

                cout << matrix\_2[i][j] << " ";

            }

            cout << endl;

        }

        tim\_yen\_ngua(matrix\_2, 5, 5);

    }

    // Bo du lieu 3 (2x7) - Dat 1 o (0,0), 0 o (1,0) de tao diem yen ngua

    {

        vector<vector<int>> matrix\_3 = {

            {1,  2,  3,  4,  5,  6,  7},

            {0, 10, 11, 12, 13, 14, 15}

        };

        cout << "Ma tran 2x7:\n";

        for (int i = 0; i < 2; i++) {

            for (int j = 0; j < 7; j++) {

                cout << matrix\_3[i][j] << " ";

            }

            cout << endl;

        }

        tim\_yen\_ngua(matrix\_3, 2, 7);

    }

    return 0;

}

Output:  
A screen shot of a computer

AI-generated content may be incorrect.

# **Bài 5.A: Chuyển chuỗi số thành số nguyên**

Code:

#include <iostream>

using namespace std;

// Chuyen chuoi so sang so nguyen

unsigned long long to\_number(const char\* s) {

    unsigned long long num = 0;

    for(int i = 0; s[i] != '\0'; i++) {

        num = num \* 10 + (s[i] - '0');

    }

    return num;

}

int main() {

    // Bo du lieu 1

    {

        const char\* s = "123456";

        cout << "So: " << to\_number(s) << endl;

    }

    // Bo du lieu 2

    {

        const char\* s = "99999";

        cout << "So: " << to\_number(s) << endl;

    }

    // Bo du lieu 3

    {

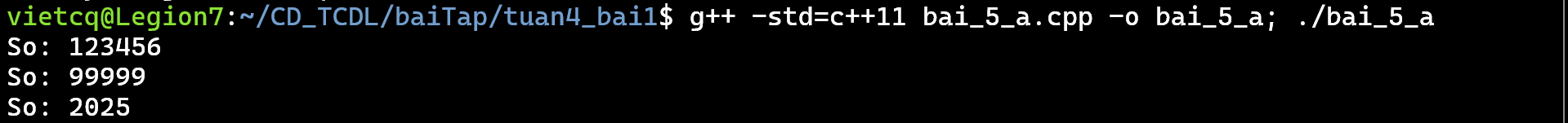
        const char\* s = "2025";

        cout << "So: " << to\_number(s) << endl;

    }

    return 0;

}

Output:  


# **Bài 5.B: Chuyển số nguyên thành chuỗi**

Code:

#include <iostream>

using namespace std;

// Ham chuyen so nguyen 'num' thanh chuoi va luu vao 's'

void to\_str(unsigned long long num, char\* s) {

    if(num == 0) {

        s[0] = '0';

        s[1] = '\0';

        return;

    }

    int len = 0;

    unsigned long long temp = num;

    while(temp > 0) {

        temp /= 10;      // Chia dần để đếm số chữ số

        len++;

    }

    s[len] = '\0';       // Ky tu ket thuc chuoi

    while(num > 0) {

        s[--len] = (num % 10) + '0';  // Lay chu so cuoi cung gan vao chuoi

        num /= 10;

    }

}

int main() {

    // Bo du lieu 1: num = 123456

    {

        unsigned long long num = 123456;

        char s[100];

        to\_str(num, s);

        cout << "Bo du lieu 1 => num = 123456 => Chuoi: " << s << endl;

    }

    // Bo du lieu 2: num = 99999

    {

        unsigned long long num = 99999;

        char s[100];

        to\_str(num, s);

        cout << "Bo du lieu 2 => num = 99999 => Chuoi: " << s << endl;

    }

    // Bo du lieu 3: num = 2025

    {

        unsigned long long num = 2025;

        char s[100];

        to\_str(num, s);

        cout << "Bo du lieu 3 => num = 2025 => Chuoi: " << s << endl;

    }

    return 0;

}

Output:  
