

# BÀI TẬP 1

- 1. Tính tổng các số từ 1 đến N với N nhập từ bàn phím**
- 2. Tính tổng  $S = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$  với n nhập từ bàn phím**
- 3. Nhập vào 1 mảng gồm N phần tử, tìm số lớn nhất và số nhỏ nhất trong mảng**

Cách khai báo mảng a gồm 100 phần tử số nguyên: int a[100]

Link bài học: <https://howkteam.vn/course/khoa-hoc-lap-trinh-c-can-ban/mang-1-chieu-trong-c-arrays-1377> (Kiến thức cần học bổ sung: mảng 1 chiều)

- 4. Nhập vào số nguyên N. In ra màn hình một mảng 2 chiều có dạng các lớp hình vuông đồng tâm, trong đó mỗi lớp có giá trị giảm dần từ ngoài vào trong.**

Output (kết quả in ra màn hình):

```
N = 3
3 3 3 3 3
3 2 2 2 3
3 2 1 2 3
3 2 2 2 3
3 3 3 3 3
```

Link bài học: <https://howkteam.vn/course/khoa-hoc-lap-trinh-c-can-ban/mang-2-chieu-trong-c-two-dimensional-arrays-1381> (Kiến thức cần bổ sung: mảng 2 chiều)

Ý tưởng: Ban đầu, chương trình nhận giá trị n từ người dùng và tính kích thước mảng là  $2 * n - 1$  để chứa các lớp từ n đến 1. Bốn cạnh của mỗi lớp được gán giá trị n bằng cách sử dụng các vòng lặp để điền lần lượt các ô ở cạnh trên, cạnh phải, cạnh dưới và cạnh trái. Sau khi hoàn thành việc điền giá trị cho một lớp, các biên giới hạn được cập nhật để thu nhỏ vào trong, và giá trị n giảm đi 1 để tiếp tục xử lý lớp kế tiếp. Quá trình lặp lại cho đến khi không còn lớp nào để điền (khi  $n < 1$ ). Cuối cùng, mảng 2D được in ra để hiển thị cấu trúc với các lớp hình vuông có giá trị giảm dần từ ngoài vào trong, tạo ra một mẫu

hình đối xứng và trực quan.

Mã giả:

BEGIN

PRINT "N = "

READ n

canh = n \* 2 - 1

lim[0] = 0 // trái

lim[1] = canh - 1 // phải

lim[2] = 0 // trên

lim[3] = canh - 1 // dưới

WHILE n >= 1 DO

// Vẽ cạnh trên

FOR i FROM lim[0] TO lim[1] DO

a[i][lim[2]] = n

END FOR

// Vẽ cạnh phải

FOR i FROM lim[2] TO lim[3] DO

a[lim[1]][i] = n

END FOR

// Vẽ cạnh dưới

FOR i FROM lim[0] TO lim[1] DO

a[i][lim[3]] = n

END FOR

// Vẽ cạnh trái

FOR i FROM lim[2] TO lim[3] DO

a[lim[0]][i] = n

END FOR

// Cập nhật giới hạn

lim[0] = lim[0] + 1

lim[1] = lim[1] - 1

lim[2] = lim[2] + 1

lim[3] = lim[3] - 1

n = n - 1

END WHILE

// In mảng ra màn hình

FOR i FROM 0 TO canh - 1 DO

FOR j FROM 0 TO canh - 1 DO

PRINT a[i][j] + " "

END FOR

PRINT newline

END FOR

END

Code mẫu:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n, a[100][100], canh, i, j;
```

```
    int lim[4]; // trai, phai, tren, duoi
```

```
    cout << "N = ";
```

```
    cin >> n;
```

```
    canh = n*2-1;
```

```
    lim[0] = 0;
```

```
    lim[1] = canh - 1;
```

```
    lim[2] = 0;
```

```
    lim[3] = canh - 1;
```

```
// tao mang theo yeu cau
```

```
while (n >= 1)
```

```
{
```

```
// canh tren
```

```
for (i = lim[0]; i <= lim[1]; i++) a[i][lim[2]] = n;
```

```

// canh phai

for (i = lim[2]; i <= lim[3]; i++) a[lim[1]][i] = n;

// canh duoi

for (i = lim[0]; i <= lim[1]; i++) a[i][lim[3]] = n;

// canh trai

for (i = lim[2]; i <= lim[3]; i++) a[lim[0]][i] = n;

// set lai gia tri

lim[0]++;
lim[1]--;
lim[2]++;
lim[3]--;

n--;

}

// xuat ra man hinh

for(i = 0; i < canh; i++)
{
    for(j = 0; j < canh; j++) cout << a[i][j] << " ";
    cout << "\n";
}

return 0;

```

}

**5. Tam giác Pascal** là một cấu trúc toán học có dạng hình tam giác, trong đó mỗi số là tổng của hai số nằm ngay phía trên nó. Tam giác này được đặt theo tên nhà toán học người Pháp Blaise Pascal, mặc dù nó đã được biết đến từ lâu trước đó ở các nền văn hóa khác như Ấn Độ, Ba Tư và Trung Quốc.

**Đặc điểm của tam giác Pascal:**

- Hàng đầu tiên (hàng 0) chứa một số 1.
- Hàng tiếp theo (hàng 1) chứa hai số 1.
- Mỗi số trong các hàng tiếp theo là tổng của hai số nằm ở vị trí liền kề phía trên nó. Các số ở rìa ngoài của tam giác đều là 1.

**Ví dụ về tam giác Pascal với số hàng N = 5:**

1	(hàng 0)
1 1	(hàng 1)
1 2 1	(hàng 2)
1 3 3 1	(hàng 3)
1 4 6 4 1	(hàng 4)

**Viết chương trình in ra tam giác Pascal với số hàng N được nhập vào từ bàn phím.**

Ý tưởng giải thuật:

- Sử dụng một mảng để lưu các giá trị của hàng hiện tại và hàng trước đó.
- Áp dụng công thức tính giá trị trong tam giác Pascal: mỗi phần tử là tổng của hai phần tử phía trên nó từ hàng trước ( $C(n, k) = C(n-1, k-1) + C(n-1, k)$ ).
- Dùng vòng lặp để xây dựng và in từng hàng của tam giác Pascal.