

KỸ THUẬT DUYỆT CÁC Ô LIÊN KỀ TRONG MẢNG 2 CHIỀU



28TECH
Become A Better Developer

5. Kỹ thuật duyệt các ô liên kề:

$i-1, j-1$	$i-1, j$	$i-1, j+1$
$i, j-1$	i, j	$i, j+1$
$i+1, j-1$	$i+1, j$	$i+1, j+1$

13

28TECH.COM.VN



28TECH
Become A Better Developer

5. Kỹ thuật duyệt các ô liên kề:

Duyệt 4 ô chung cạnh với ô $[i][j]$

```
#include <bits/stdc++.h>
using namespace std;
int dx[4] = {-1, 0, 0, 1};
int dy[4] = {0, -1, 1, 0};

int main(){
    int a[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int i = 1, j = 1;
    for(int k = 0; k < 4; k++){
        int i1 = i + dx[k], j1 = j + dy[k];
        cout << a[i1][j1] << " ";
    }
}
```

OUTPUT: 2 4 6 8

4

28TECH.COM.VN



5. Kỹ thuật duyệt các ô liền kề:

Duyệt 8 ô chung đỉnh với ô [i][j]

```
#include <bits/stdc++.h>
using namespace std;
int dx[8] = {-1, -1, -1, 0, 0, 1, 1, 1};
int dy[8] = {-1, 0, 1, -1, 1, -1, 0, 1};

int main(){
    int a[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int i = 1, j = 1;
    for(int k = 0; k < 8; k++){
        int i1 = i + dx[k], j1 = j + dy[k];
        cout << a[i1][j1] << " ";
    }
}
```

OUTPUT: 1 2 3 4 6 7 8 9

15



5. Kỹ thuật duyệt các ô liền kề:

1	2	3	4	5	6
7	8	9	1	2	5
1	2	1	0	3	5
1	2	1	3	4	9
1	2	1	3	0	4
1	8	7	6	2	9

Duyệt 8 ô xung quanh nước đi của quân mã

```
#include <bits/stdc++.h>
using namespace std;
int dx[8] = {-2, -2, -1, -1, +1, +1, +2, +2};
int dy[8] = {-1, +1, -2, +2, -2, +2, -1, +1};

int main(){
    int a[6][6] = {
        {1, 2, 3, 4, 5, 6},
        {7, 8, 9, 1, 2, 5},
        {1, 2, 1, 0, 3, 5},
        {1, 2, 1, 3, 4, 9},
        {1, 2, 1, 3, 0, 4},
        {1, 8, 7, 6, 2, 9}
    };
    int i = 2, j = 3;
    for(int k = 0; k < 8; k++){
        int i1 = i + dx[k], j1 = j + dy[k];
        cout << a[i1][j1] << " ";
    }
}
```

OUTPUT: 3 5 8 5 2 9 1 0

16

=> **Nhận xét:** Kỹ thuật này rất hay dùng trong các bài toán (Đường di chuyển của quân cờ, Bài toán đảo island, Tìm min max trong mảng...vv)