



28TECH
Become A Better Developer

MỘT SỐ HÀM TOÁN HỌC THÔNG DỤNG





Lập trình liên quan nhiều tới việc **tính toán**, vì thế các **hàm toán học** được sử dụng thường xuyên. Ngôn ngữ lập trình C cung cấp một **thư viện** là **<math.h>** hỗ trợ các bạn lập trình viên những hoàn quan trọng được sử dụng nhiều.



Để sử dụng các hàm này, các bạn cần khai báo thư viện **math.h** vào chương trình của mình



1. HÀM ABS

- `int abs(int a)`: Trả về trị tuyệt đối của số nguyên a

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main(){  
    printf("%d\n", abs(10));  
    printf("%d\n", abs(-10));  
}
```

Output :

10

10



2. HÀM SQRT



`double sqrt(double a)` : Trả về căn bậc 2 của a

`sqrt (Square root)` là hàm tính căn bậc 2, hàm này trả về căn bậc 2 của một số không âm, kết quả được trả về dưới kiểu double

2. HÀM SQRT

VÍ DỤ

```
#include <stdio.h>
#include <math.h>

int main(){
    int n = 17;
    double res1 = sqrt(n);
    printf("%.5lf\n", res1);
    int res2 = sqrt(n);    // res2 = 4.12311 = 4
    printf("%d\n", res2);
    int res3 = sqrt(16);   // res3 = 4.0000 = 4
    printf("%d\n", res2);
}
```

Output :

4.12311

4

4

Chú ý: Khi sử dụng hàm `sqrt`, các bạn nên nhớ rằng nó trả về **số double** chứ không phải **số nguyên**.

3. HÀM POW

- `double pow(double a, double b)` : Tính a^b
- Pow là hàm giúp tính lũy thừa, nó cũng trả về số double

```
#include <stdio.h>
#include <math.h>
```

```
int main(){
    printf("%21f\n", pow(2, 10));
    printf("%d\n", pow(2, 10)); // sai đặc tả
    int res = pow(2, 10);
    printf("%d\n", res);
}
```

Output :

1024.000000

0

1024

Chú ý: Hàm này cũng có thể được sử dụng thay cho hàm `sqrt` bằng cách lấy số mũ là 0.5.

4. HÀM CEIL

- `double ceil(double a)`: Trả về số nguyên lớn hơn gần a nhất ở kiểu số double

```
#include <stdio.h>
#include <math.h>
```

```
int main(){
    printf("%21f\n", ceil(1.2));
    printf("%d\n", ceil(5.6)); // sai đặc tả
    printf("%d\n", (int)ceil(5.6));
}
```

Output :

2.00

0

6

Chú ý: Hàm này trả về số nguyên lớn hơn gần nhất với tham số nhưng nó lại ở dạng double.



5. HÀM FLOOR

- `double floor(double a)` : Trả về số nguyên nhỏ hơn gần a nhất ở kiểu số double

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main(){  
    printf("%.2f\n", floor(1.2));  
    printf("%d\n", floor(5.6)); // sai đặc tả  
    printf("%d\n", (int)floor(5.6));  
}
```

Output :

1.00

0

5

6. HÀM ROUND

- `double round(double a)` : Trả về số nguyên gần a nhất ở kiểu `double`

```
#include <stdio.h>
#include <math.h>
```

```
int main(){
    printf("%d\n", (int)round(3.5));
    printf("%d\n", (int)round(3.6));
    printf("%d\n", (int)round(3.2));
}
```

Output :

4

4

3

7. HÀM FMIN

- `double fmin(double a, double b)` : Trả về số nhỏ hơn trong 2 số a và b ở dạng `double`
- Bạn có thể sử dụng nó để tìm min giữa 2 số nguyên nhưng nhớ ép kiểu kết quả.

```
#include <stdio.h>
#include <math.h>
```

```
int main(){
    printf("%d\n", fmin(5, 10)); // sai đặc tả
    printf("%d\n", (int)fmin(5, 10));
}
```

Output :

0
5

8. HÀM FMAX

- `double fmax(double a, double b)` : Trả về số lớn hơn trong 2 số a và b ở dạng `double`
- Bạn có thể sử dụng nó để tìm max giữa 2 số nguyên nhưng nhớ ép kiểu kết quả.

```
#include <stdio.h>
#include <math.h>
```

```
int main(){
    printf("%d\n", fmax(5, 10)); // sai đặc tả
    printf("%d\n", (int)fmax(5, 10));
    int res = fmax(10, 20);
    printf("%d\n", res);
}
```

Output :

0

10

20

