

REVIEW

1. Biến và kiểu dữ liệu

- Tên biến:
 - Có thể bao gồm kí tự, chữ số và dấu gạch dưới
 - Không bắt đầu bằng chữ số
 - Được sử dụng nhiều lần trong chương trình
 - Phân biệt hoa thường
- Kiểu dữ liệu:
 - Kiểu kí tự: char

```
char c = 'a';
```

- Kiểu số nguyên: short, int, long,...

```
int A = 10;  
long B = 100000000;
```

- Kiểu số thực: float, double

```
float soPI = 3.14;
```

- Kiểu logic: bool

```
bool ok = true;
```

2. Toán tử (thường dùng)

- Toán tử số học: +, -, *, /, %
- Toán tử gán: =, +=, -=, *=, /=, %=
- Toán tử logic: && (và), || (hoặc), ! (phủ định)
- Toán tử quan hệ: == (bằng), != (khác), >, <, >=, <=

Ví dụ: Cho điểm Toán, Văn và điểm trung bình tất cả các môn của 1 học sinh. Kiểm tra học sinh đó có được loại Giỏi hay không?

```
int diemTrungBinh = 8;
int diemToan = 8;
int diemVan = 7;
if (diemTrungBinh < 8) || (diemToan < 6.5) || (diemVan < 6.5){
    printf("Tôi không được loại Giỏi");
} else
if (diemToan >=8 || diemVan >=8){
    printf("Tôi được loại Giỏi");
} else {
    printf("Tôi không được loại Giỏi");
}
```

3. Câu lệnh điều kiện

- Cú pháp:

```
if (điều kiện 1){
    // câu lệnh 1 (chỉ thực hiện khi điều kiện 1 đúng)
} else if (điều kiện 2){
    // câu lệnh 2 (chỉ thực hiện khi điều kiện 2 đúng)
} else ... //(chỉ thực hiện khi điều kiện 1 và 2 sai)
```

Toán tử 3 ngôi:

- Cú pháp:

```
<điều kiện> ? <lệnh khi điều kiện đúng> : <lệnh khi điều kiện sai>;
```

Ví dụ:

```
bool ok = true;
(ok == true) ? printf("ok") : printf("not ok");
```

Tương đương với:

```
bool ok = true;
if (ok){
    printf("ok");
} else {
    printf("not ok");
}
```

4. Câu lệnh lặp

- Có 2 loại:
 - Lặp với số lần biết trước (dùng **for**)
 - Lặp với số lần không biết trước (dùng **while** hoặc **do...while**)
 - while: kiểm tra điều kiện mới thực hiện câu lệnh
 - do...while: thực hiện câu lệnh trước khi kiểm tra điều kiện (đảm bảo câu lệnh được thực hiện ít nhất 1 lần).

```
for (int i=0; i<=10; i++){  
    printf("So %d:", i);  
}
```

```
int dem = 10;  
while (dem>=0){  
    printf("%d", dem);  
    dem--;  
}
```

```
int dem = 10;  
do {  
    printf("%d", dem);  
    dem--;  
} while (dem>=0)
```

5. Hàm (function)

Khái niệm: Hàm là đoạn chương trình thực hiện trọn vẹn một công việc nhất định (cụ thể). Hàm giúp chia cắt việc lớn thành nhiều việc nhỏ hơn, điều này tương đương với việc chia bài toán lớn thành các bài toán nhỏ hơn để giải, như vậy thì việc giải bài toán sẽ trở nên dễ dàng hơn. Ngoài ra, hàm còn giúp cho chương trình trở nên sáng sủa, dễ sửa, nhất là đối với các chương trình lớn.

Định nghĩa hàm:

```
Kiểu_Dữ_Liệu Tên_Hàm(Tham_Số_1, Tham_Số_2, ...) {  
    Xử lý;  
}
```

Giải thích:

- Kiểu dữ liệu có thể là **int**, **long**, **float**, **double**, **char** hoặc **void**. Đây chính là dữ liệu mà hàm có thể trả về (hay còn gọi là đầu ra – output).
- Tên hàm phải là: động từ, không chứa khoảng trắng, không trùng từ khoá, không chứa ký tự đặc biệt và đặt tên có ý nghĩa.

- Tham số là dữ liệu đầu vào của hàm và được khai báo theo cú pháp sau:
- Kiểu_Dữ_Liệu: Tên_Tham_Số_1, Kiểu_Dữ_Liệu Tên_Tham_Số_2,...

Ví dụ 1: hàm kiểm tra số chẵn

```
bool kiemTraSoChan(int n){  
    return (n%2==0);  
}
```

Ví dụ 2: Hàm tính tổng từ A đến B

```
#include <stdio.h>  
  
int tinhTongADenB(int A, int B){  
    int tong = 0;  
    for (int i=A; i<=B; i++)  
        tong += i;  
    return tong;  
}
```

Ví dụ 3: hàm kiểm tra số nguyên tố

```
#include <stdio.h>  
#include <math.h>  
  
bool kiemTraNguyenTo(int n){  
    if (n < 2) return false;  
    for (int i=2; i<=sqrt(n); i++)  
        if (n%i==0) return false;  
    return true;  
}
```

Ví dụ 4: hàm kiểm tra số chính phương

```
#include <stdio.h>  
#include <math.h>  
  
bool kiemTraChinhPhuong(int n){  
    int sqr = sqrt(n);  
    return (sqr*sqr == n);  
}
```

Bài tập hàm:

1. Viết hàm tính tổng các số lẻ trong đoạn từ A đến B (A, B là số nguyên được nhập từ bàn phím).
2. Viết hàm kiểm tra số hoàn hảo. Biết số hoàn hảo là số có tổng các ước nhỏ hơn nó bằng chính nó. Ví dụ:
6 có ước nhỏ hơn 6 là: 1, 2, 3. Do $(1+2+3=6)$ nên 6 là số hoàn hảo
3. Viết hàm đếm số lượng số nguyên tố từ số nguyên A đến số nguyên B (được nhập từ bàn phím)
4. Viết chương trình tìm số có số ước nhiều nhất trong đoạn từ A đến B. (ví dụ từ 1 đến 6 thì 6 là số có số lượng ước nhiều nhất (có 4 ước số)). Nếu có nhiều số có cùng số lượng ước nhiều nhất thì in ra bất kì.