**Câu 1: Quy tắc tính thứ tự( độ ưu tiên ) toán tử số học (Arithmetic operator) trong C ?**

1. **Phép nhân (\*) và phép chia (/):** Phép nhân và phép chia có cùng mức ưu tiên cao nhất. Chúng được thực hiện từ trái sang phải nếu xuất hiện cùng mức ưu tiên.
2. **Phép cộng (+) và phép trừ (-):** Phép cộng và phép trừ cũng có mức ưu tiên tương đối cao. Chúng được thực hiện từ trái sang phải nếu xuất hiện cùng mức ưu tiên.
3. **Phép gán (=):** Toán tử gán (=) có mức ưu tiên thấp hơn so với các toán tử số h

**Câu 2: Quy tắc tính thứ( độ ưu tiên ) toán tử quan hệ (Relational operator) trong C ?**

1. **Phép so sánh bằng (==) và phép so sánh không bằng (!=):** Các phép so sánh này có cùng mức ưu tiên cao nhất và được thực hiện từ trái sang phải nếu xuất hiện cùng mức ưu tiên.
2. **Phép so sánh lớn hơn (>) và phép so sánh nhỏ hơn (<):** Các phép so sánh lớn hơn và nhỏ hơn cũng có mức ưu tiên cao và được thực hiện từ trái sang phải nếu xuất hiện cùng mức ưu tiên.
3. **Phép so sánh lớn hơn hoặc bằng (>=) và phép so sánh nhỏ hơn hoặc bằng (<=):** Các phép so sánh này cũng có mức ưu tiên tương đối cao và được thực hiện từ trái sang phải nếu xuất hiện cùng mức ưu tiên.

**Câu 3: Quy tắc tính thứ( độ ưu tiên ) toán tử logical (Logical operator) trong C ?**

1. **Phép NOT (!):** Toán tử NOT (!) có độ ưu tiên cao nhất trong các toán tử logical. Nó được thực hiện trước cùng khi xuất hiện trong biểu thức.
2. **Phép AND (&&):** Toán tử AND (&&) có độ ưu tiên thấp hơn so với NOT. Nếu có nhiều phép AND trong biểu thức, chúng được thực hiện từ trái sang phải.
3. **Phép OR (||):** Toán tử OR (||) có độ ưu tiên thấp nhất trong các toán tử logical. Cũng giống như phép AND, nếu có nhiều phép OR trong biểu thức, chúng được thực hiện từ trái sang phải.

**Câu 4: Quy tắc tính thứ(độ ưu tiên) toán tử logic bitwise (Bitwise Logical operator) trong C ?**

1. **Phép AND bitwise (&):** Toán tử **&** có độ ưu tiên cao nhất trong các toán tử logic bitwise. Nếu có nhiều phép AND trong biểu thức, chúng được thực hiện từ trái sang phải.
2. **Phép OR bitwise (|):** Toán tử **|** có độ ưu tiên thấp hơn so với phép AND bitwise. Nếu có nhiều phép OR trong biểu thức, chúng được thực hiện từ trái sang phải.
3. **Phép XOR bitwise (^):** Toán tử **^** có độ ưu tiên thấp hơn so với phép OR bitwise. Nếu có nhiều phép XOR trong biểu thức, chúng được thực hiện từ trái sang phải.

**Câu 5: Cách chuyển đổi kiểu dữ liệu (type conversion) trong một biểu thức?**

**1 Chuyển đổi tường minh (Explicit Type Conversion):** Còn được gọi là "type casting," chuyển đổi tường minh là quá trình chuyển đổi một biến từ một kiểu dữ liệu sang một kiểu dữ liệu khác bằng cách sử dụng các toán tử hoặc hàm chuyển đổi tường minh

2. **Chuyển đổi tự động (Implicit Type Conversion):** Còn được gọi là "type coercion," chuyển đổi tự động xảy ra tự động khi bạn thực hiện phép toán hoặc gán giá trị giữa các biến có kiểu dữ liệu khác nhau. Các quy tắc chuyển đổi tự động được xác định bởi quy tắc độ ưu tiên của kiểu dữ liệu

**Câu 5: Độ ưu tiên của các toán tử trong một biểu thức ?**

1. **Phép gán (Assignment Operators):** **=** (gán giá trị), **+=** (cộng và gán), **-=** (trừ và gán), **\*=** (nhân và gán), **/=** (chia và gán), và các phép gán khác có mức ưu tiên thấp nhất. Chúng được thực hiện từ phải sang trái.
2. **Toán tử logic bitwise (Bitwise Logical Operators):** **&** (AND bitwise), **|** (OR bitwise), **^** (XOR bitwise) có mức độ ưu tiên thấp hơn so với các phép gán. Chúng được thực hiện từ trái sang phải.
3. **Phép so sánh (Relational Operators):** **<** (nhỏ hơn), **>** (lớn hơn), **<=** (nhỏ hơn hoặc bằng), **>=** (lớn hơn hoặc bằng), **==** (bằng), **!=** (không bằng) có mức độ ưu tiên thấp hơn so với các phép toán bitwise. Chúng được thực hiện từ trái sang phải.
4. **Toán tử logic (Logical Operators):** **&&** (AND logic) và **||** (OR logic) có mức độ ưu tiên thấp hơn so với phép so sánh. Chúng được thực hiện từ trái sang phải.
5. **Toán tử ba ngôi (Ternary Operator):** **? :** có mức độ ưu tiên thấp hơn so với các toán tử logic. Nó được sử dụng để tạo điều kiện và lựa chọn giữa hai giá trị dựa trên một điều kiện.
6. **Toán tử số học (Arithmetic Operators):** **+** (cộng), **-** (trừ), **\*** (nhân), **/** (chia), **%** (lấy phần dư) có mức độ ưu tiên thấp hơn so với các toán tử ba ngôi. Chúng được thực hiện từ trái sang phải.
7. **Toán tử đơn ngôi (Unary Operators):** **+** (đơn ngôi), **-** (đơn ngôi), **++** (tăng giá trị lên 1), **--** (giảm giá trị xuống 1), **!** (phủ định) có mức độ ưu tiên thấp hơn so với các toán tử số học.

* **Biểu thức có toán tử số học và toán tử quan hệ với toán tử logic thì thứ tự các loại toán tử như thế nào ?**

1. **Toán tử số học (arithmetic operators):** Các phép toán số học như cộng (+), trừ (-), nhân (\*), chia (/), và lấy phần dư (%) được thực hiện đầu tiên trong biểu thức. Chúng có mức độ ưu tiên thấp hơn so với toán tử quan hệ và toán tử logic.
2. **Toán tử quan hệ (relational operators):** Các phép so sánh như **<**, **>**, **<=**, **>=**, **==**, và **!=** được thực hiện sau các phép toán số học. Chúng có mức độ ưu tiên thấp hơn so với toán tử số học và toán tử logic.
3. **Toán tử logic (logical operators):** Các toán tử logic như **&&** (AND logic) và **||** (OR logic) được thực hiện sau cả toán tử số học và toán tử quan hệ. Chúng có mức độ ưu tiên thấp hơn so với cả hai loại toán tử trên.