

1)Pointer: Verinin bellek adresini gösteren değişkendir.yani direkt olarak verinin tutulduğu bellek üzerinden verimli kullanımı gibi avantajlara sahiptir,dizinin elemanlarına doğrudan erişebilirsiniz.Bu,diziye kopyalamaktan daha az maliyeti ve daha hızlıdır

İşaretçi değişken `int *ptr = &a` şeklinde gösterilebilir

C++ programlama dilinde güçlü bir araçtır ve dinamik bellek yönetimi, fonksiyon göstericileri, diziler, yapılar ve sınıflar gibi birçok farklı konuda kullanılabilir. Ancak, işaretçiler, yanlış kullanıldığında hatalara ve güvenlik açıklarına neden olabilir, bu nedenle dikkatli kullanılmalıdır.

2) Bitwise operasyonlar, özellikle donanım veya low-level programlama gibi alanlarda sıklıkla kullanılır. Bazı durumlarda, bu operasyonların kullanılması daha hızlı veya daha etkili bir yöntem sağlayabilir. Örneğin, sol kaydırma (`<<`) işlemi bir sayıyı 2'nin üssü kadar çarpmak için kullanılabilir. Bu, programlama dillerindeki çarpma işleminden daha hızlı olabilir. Benzer şekilde, sağ kaydırma (`>>`) işlemi, bir sayıyı 2'nin üssüne bölerek bölme işleminden daha hızlı bir şekilde sonuç verebilir. AND (`&`) işlemi, bir bit dizisi içinde belirli bitleri belirlemek için kullanılabilir. OR (`|`) işlemi ise belirli bitleri ayarlama veya değiştirme için kullanılabilir. Bu nedenlerle, bitwise operasyonlar özellikle düşük seviye programlama, veri işleme veya donanım tasarımı gibi alanlarda sıklıkla kullanılır.

`int x = 7 // 0111` ise

`int y = x << 2; y = 7 * (2^2)` şeklindedir binary `11100` =sonuç 28

3)bir koşulu çoklu test etme işlemidir

```
switch (koşullandırılan) {
case 1: // birinci koşul
    printf("deger 1'e esit");
    break;
case 2:
    printf("deger 2'ye esit");
    break;
default: // bitiş
    printf("deger tanımsiz");
    break;
}
```