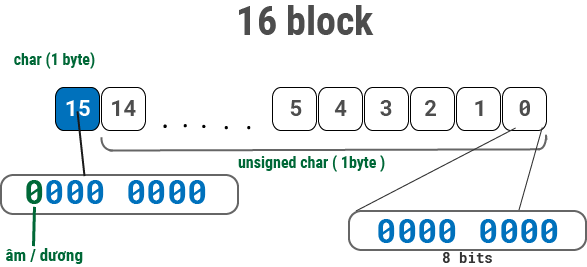
**NGUYÊN LÝ LƯU THÔNG TIN SỐ NGUYÊN LỚN CỦA ĐỒ ÁN**

* Chia 128 bits dữ liệu thành 1 mảng 16 block độ lớn 1 byte (unsigned char)
* Riêng block cao nhất được định nghĩa là [char] để có thể dễ nhận dạng số âm.
* Bit cao nhất (tức là bit cao nhất của block thứ 15) dùng để đánh dấu số âm.
* Lưu số nguyên theo định nghĩa số bias bù 2.



* Mỗi block có giá trị từ 0 – 255.

**CHUYỂN CHUỖI THÀNH SỐ NGUYÊN LỚN (16 BYTES)**

1. **Chuỗi kí tự bit**

* Ví dụ: “10101011”, “00111001”, ….
* Mỗi kí tự chỉ có 2 giá trị là ‘0’ và ‘1’
* Ý tưởng:
  + Kí tự thứ n trong chuỗi sẽ là giá trị của bit thứ n trong số nguyên
  + Do ta chia số nguyên thành 16 block nên ta cần phải tính toán vị trí bit trong số ứng với vị trí của block nào và vị trí bit nào trong block đó.
  + Sử dụng các phép logic để gán bit đó vào vị trí cụ thể của block.

**bit** = chuoiBin[len – position - 1];

vi\_tri\_block = position / 8;

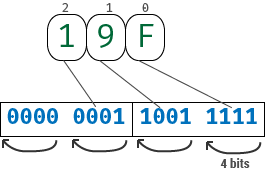
vi\_tri\_trong\_block = position % 8;

//su dung phep toan ^ de gan bit vao 1 so

block[vi\_tri\_block] = block[vi\_tri\_block] ^ (**bit** << vi\_tri\_trong\_block);

1. **Chuỗi kí tự hex**

* Ví dụ : “00FFAA”, “13B”, …
* Mỗi kí tự là một số tương ứng trong hệ số thập lục phân (1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)



* Ý tưởng giống với chuyển chuỗi kí tự bit nhưng:
  + Giá trị ở hệ số n trong chuỗi sẽ tương ứng với 4 bits thứ n trong số nguyên 16 bytes.
  + Sử dụng các phép logic để gán 4 bit vào vị trí cụm 4 bit thứ n.

**Hex** = chuoiHex**[**position**];**

Vi\_tri\_block = position / 2;

Vi\_tri\_trong\_block = position % 2;

Block**[**vi\_tri\_block**]** = block**[**vi\_tri\_block**]** ^ (**hex** << 4\*(vi\_tri\_trong\_block));

**Hex** = chuoiHex**[**position**];**

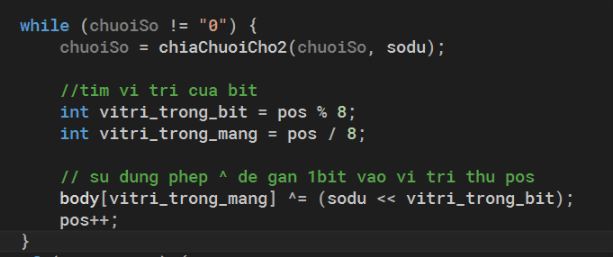
Vi\_tri\_block = position / 2;

Vi\_tri\_trong\_block = position % 2;

Block**[**vi\_tri\_block**]** = block**[**vi\_tri\_block**]** ^ (**hex** << 4\*(vi\_tri\_trong\_block));

1. **Chuỗi kí tự thập phân:**

* Ví dụ: “12345”, “93”, …
* Mỗi kí tự là số 0 – 9.
* Do cơ số 10 không là bội ước của 255, nên việc chia thành từng dãy bits không dễ dàng. Cách duy nhất chúng ta có thể làm được là “chia 2” chuỗi số dần để lấy số dư – là bit thứ 2n của chuỗi số đó.
* Ý tưởng
  + Thiết lập thuật toán “chia 2” chuỗi số và xuất ra số dư
  + Chia dần chuỗi số, tìm số dư
  + Gán bit (là số dư tìm được) vào vị trí tăng dần khi chia.



Trong hình: **body** là *dãy block*, **pos** là vị trí đếm từ 0 lên.