

CÁC PHÉP TOÁN VỚI CÁC SỐ NGUYÊN TRONG HỆ ĐẾM CƠ SỐ (-2)

Trước tiên, cần lưu ý rằng các phép toán MOD, DIV với các số nguyên có dấu khá phức tạp, cụ thể như sau:

- Phép toán MOD: $a \text{ MOD } b = \text{sign}(a) * (|a| \text{ MOD } |b|)$, nghĩa là, kết quả là số âm chỉ trong trường hợp a âm. Cũng tức là, dấu của kết quả là dấu của a.
- Phép toán DIV: $a \text{ DIV } b = \text{sign}(a/b) * (|a| \text{ DIV } |b|)$, nghĩa là, kết quả là âm chỉ khi a/b âm. Cũng tức là, dấu của kết quả là dấu của a/b.

Từ đó:

$a \text{ MOD } (-2)$ bằng 0 nếu a chẵn, bằng 1 nếu a lẻ dương, bằng -1 nếu a lẻ âm.

$a \text{ DIV } (-2)$ bằng $|a| \text{ DIV } 2$ nếu $a < 0$, bằng $-(a \text{ DIV } 2)$ nếu $a \geq 0$ (dấu của kết quả ngược với dấu của a).

Phép toán MOD, DIV với các số nguyên biểu diễn ở cơ số (-2)

```
Procedure ModDivAm2(a:int64;var m:byte;var d:int64);
//m = a MOD (-2), voi m=0..1; d=a DIV (-2)
Begin
  m:=abs(a) MOD 2;
  If m=1 Then d:=(1-a) DIV 2 Else d:= -a DIV 2;
End;
```

Chuyển số nguyên trong hệ thập phân sang hệ cơ số (-2)

```
Function ToAm2(a:int64):string;
Begin
  ToAm2:="";
  Repeat
    ToAm2:=chr(48+abs(a) MOD 2)+ToAm2;
    If odd(a) Then a:=(1-a) DIV 2 else a:=-a DIV 2;
  Until a=0;
End;
```

Chuyển số nguyên trong hệ cơ số (-2) sang hệ thập phân

```
Function FromAm2(s:string):int64;
Var i:byte;
    tmp:int64;
Begin
  FromAm2:=0;
  tmp:=1;
  For i:=length(s) downto 1 do
    Begin
      If s[i]='1' Then FromAm2:=FromAm2+tmp;
      tmp:=tmp*(-2);
    End;
  End;
```

Phép toán bit với các số nguyên biểu diễn ở cơ số (-2)

Mỗi số nguyên trong cơ số (-2) có dạng: $b_k b_{k-1} \dots b_0 = b_k(-2)^k + b_{k-1}(-2)^{k-1} + \dots + b_0$ trong đó $b_i \in \{0, 1\}$

Xét tổng $a_k(-2)^k \pm b_k(-2)^k = (a_k \pm b_k + \text{mem})(-2)^k$, trong đó $a_k, b_k \in \{0, 1\}$ và mem là số nhớ.

Với $t = (a_k \pm b_k + \text{mem})$ ta thu được bit ở hàng k sẽ là $c_k = |t| \text{ MOD } 2$. Ta cần xác định giá trị mới của mem từ giá trị của t.

Việc thay t bởi c_k đã khiến giá trị tại hàng này “tăng thêm” một lượng là $\partial = (c_k - t)(-2)^k = -((c_k - t) \text{ DIV } 2)(-2)^{k+1}$. Lượng này cần bị “bớt đi” ở hàng kế tiếp (k+1). Điều này có nghĩa là hàng kế tiếp cần được “cộng thêm” một lượng $-\partial = ((c_k - t) \text{ DIV } 2)(-2)^{k+1}$ mà hệ số của $(-2)^{k+1}$ là $\text{mem} = (c_k - t) \text{ DIV } 2$. Từ kết quả trên, ta thu được các thuật toán sau:

1. Phép cộng trong hệ cơ số (-2)

```
Function CongAm2(x, y: string): string; // x+y
Var mem: shortint;
    k: byte;
Begin
  While length(x) < length(y) Do x := '0' + x;
  While length(y) < length(x) Do y := '0' + y;
  x := '00' + x;
  y := '00' + y;
  mem := 0;
  For k := length(x) Downto 1 Do
    Begin
      mem := mem + ord(x[k]) + ord(y[k]) - 96;
      x[k] := chr(48 + abs(mem) MOD 2);
      If x[k] = '1' Then mem := (1 - mem) DIV 2 Else mem := - mem DIV 2;
    End;
  While (x[1] = '0') and (length(x) > 1) Do Delete(x, 1, 1);
  Exit(x);
End;
```

2. Phép trừ trong hệ cơ số (-2)

```
Function TruAm2(x, y: string): string; // x - y
Var mem: shortint;
    k: byte;
Begin
  While length(x) < length(y) Do x := '0' + x;
  While length(y) < length(x) Do y := '0' + y;
  x := '00' + x;
  y := '00' + y;
  mem := 0;
  For k := length(x) Downto 1 Do
    Begin
      mem := mem + ord(x[k]) - ord(y[k]);
      x[k] := chr(48 + abs(mem) MOD 2);
      If x[k] = '1' Then mem := (1 - mem) DIV 2 Else mem := - mem DIV 2;
    End;
  While (x[1] = '0') and (length(x) > 1) Do Delete(x, 1, 1);
  Exit(x);
End;
```

3. Phép nhân trong hệ cơ số (-2)

```
Function NhanAm2(x, y: string): string; // x * y
Var mem: integer;
    k:byte;
    z:string;
Begin
    z:=x+y;
    mem:=0;
    For k:=length(z) Downto 1 Do
        Begin
            For i:=max(1, k-length(y)) To min(k-1, length(x)) Do
                mem:=mem+(ord(x[i])-48)*(ord(y[k-i])-48);
            z[k]:=chr(48+abs(mem) MOD 2);
            If z[k]='1' Then mem:=(1-mem) DIV 2 Else mem:= - mem DIV 2;
        End;
    While (z[1]='0') and (length(z)>1) Do Delete(z,1,1);
    Exit(z);
End;
```