P1 Transformer

《计算机组织结构》课程助教组

南京大学 NANJING UNIVERSITY

作业要求

1. 将整数真值（十进制表示）转化成补码表示的二进制，默认长度32位

```
public static String intToBinary(String numStr)
```

2. 将补码表示的二进制转化成整数真值（十进制表示）

```
public static String binaryToInt(String binStr)
```

3. 将十进制整数的真值转化成NBCD表示（符号位用4位表示）

```
public static String decimalToNBCD(String decimalStr)
```

4. 将NBCD表示（符号位用4位表示）转化成十进制整数的真值

```
public static String NBCDToDecimal(String NBCDStr)
```

5. 将浮点数真值转化成32位单精度浮点数表示

```
public static String floatToBinary(String floatStr)
```

6. 将32位单精度浮点数表示转化成浮点数真值

```
public static String binaryToFloat(String binStr)
```

解题步骤

1. 将整数真值（十进制表示）转化成补码表示的二进制，默认长度32位

```
public static String intToBinary(String numStr)
```

实现思路：

1.使用推荐的parse 函数 Long.parseLong(numStr)将numStr转换成Long类型计算

2.考虑得到的Long型数字的正负，如果是负数，可以考虑：

```
num = (long) (Math.pow(2, 32) + num);
```

3.最后逐位计算Binary串的值（注意：下面方法得到的结果顺序是反的）

```
for(int i = 0; i < 32; i++){
        array[i] = (char) ('0' + num % 2);
        num /= 2;
    }
```

解题步骤

2. 将补码表示的二进制转化成整数真值（十进制表示）

```
public static String binaryToInt(String binStr)
```

实现思路：判断正负，如为负取反加一，然后从右到左依次计算

1.不妨先算符号位外的值

```
for(int i = 31; i >= 0; i--){
        num += (long) ((binStr.charAt(i) - '0') * Math.pow(2, 31 - i));
    }
```

2.如果为负，再

```
num -= (long) (Math.pow(2, 32));
```

解题步骤

3. 将十进制整数的真值转化成NBCD表示（符号位用4位表示）

```
public static String decimalToNBCD(String decimalStr)
```

注意符号位的表示：“1100”表示正，“1101”表示负

关键步骤：

```
int buffer = num % 10;
        for(int j = 0; j < 4; j++){
            array[i] = (char) ('0' + buffer % 2);
            buffer /= 2;
            i--;
        }
        num /= 10;
```

解题步骤

4. 将NBCD表示（符号位用4位表示）转化成十进制整数的真值

```
public static String NBCDToDecimal(String NBCDStr)
```

注意符号位的表示：“1100”表示正，“1101”表示负

关键步骤：

```
int num = 0;
    for(int i = 4; i < 32;){
        num *= 10;
        for(int j = 0; j < 4; j++){
            num += (int) ((NBCDStr.charAt(i) - '0') * Math.pow(2, 3 - j));
            i++;
        }
    }
```

解题步骤

5. 将浮点数真值转化成32位单精度浮点数表示

```
public static String floatToBinary(String floatStr)
```

实现逻辑:

1.讨论正负符号、边界情况判断（+Inf、-Inf……）

2.根据数字num处理后的绝对值与1比较，

```
if ( num >= 1){
    //通过不断进行 num /=2 并且相应 阶数++ ， 直到num符合尾数的格式( num >= 1 && num < 2)
}
else{
    //通过不断进行 num *= 2 并且相应 阶数-- ， 直到num符合尾数的格式( num >= 1 && num < 2)
}
```

要注意的点就是控制好 num(我们想将这个值化为尾数) 和 阶数的转换关系，以及最后将 阶数 转换为 阶码

3.将得到的符号位、阶码、尾数转换成相应二进制表示并组合在一起（需要格外注意尾数的转换，即最后是否为规格化数）

解题步骤

6. 将32位单精度浮点数表示转化成浮点数真值

```
public static String binaryToFloat(String binStr)
```

实现逻辑:

1.讨论正负、边界情况判断（+Inf、-Inf……）

2.讨论是否为规格化数并计算 阶码 对应的 阶数

3.将尾数转换成float类型并乘以 阶数 对应的值

```
//mantissa * Math.pow( 2, exponent )
```