**十六进制转八进制**

**问题描述**  
　　给定n个十六进制正整数，输出它们对应的八进制数。  
  
**输入格式**  
　　输入的第一行为一个正整数n （1<=n<=10）。  
　　接下来n行，每行一个由0~9、大写字母A~F组成的字符串，表示要转换的十六进制正整数，每个十六进制数长度不超过100000。  
  
**输出格式**  
　　输出n行，每行为输入对应的八进制正整数。  
  
　　**【注意**】  
　　输入的十六进制数不会有前导0，比如012A。  
　　输出的八进制数也不能有前导0。  
  
**样例输入**  
　　2  
　　39  
　　123ABC  
  
**样例输出**  
　　71  
　　4435274  
  
　　**【提示**】  
　　先将十六进制数转换成某进制数，再由某进制数转换成八进制。

锦囊1

使用二进制。

锦囊2

先把十六进制转成二进制，每位十六进制正好转成4位二进制，然后再将二进制转成八进制，每三个二进制转成一个八进制。

C++代码

#include <iostream>

#include <stdio.h>

#include <string.h>

#include <STDLIB.H>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int GetI(char c)

{

return c>>4&1?c&15:(c&15)+9;

}

int main(int argc, char \*argv[]) {

char arr[200001] = {'\0'};

char brr[400001] = {'\0'};

int n = 0;

int i = 0;

scanf("%d",&n);

for(i = 0;i < n;i++)

{

scanf("%s",arr);

int m[3] = {1,16,256};

int len = strlen(arr);

int j = len-1;

int a,b,c;

a = b = c = 0;

int k = 0,l = 0;

int count = 0;

while(j>-1)

{

a += (arr[j]>>4&1?arr[j]&15:(arr[j]&15)+9)\*m[k]; //个位

if(k==2||j==0)

{

while(a)

{

brr[l++] = ((a&7)|48);

a = a>>3;

count++;

}

while(j!=0&&count<4)

{

brr[l++] = '0';

count++;

}

count = 0;

}

k = (k+1)%3;

j--;

}

strrev(brr);

printf("%s\n",brr);

memset(arr,'\0',(sizeof(char)\*200001));

memset(brr,'\0',(sizeof(char)\*400001));

}

return 0;

}

C代码

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MaxSize 100000

void saveB(char \*b,char c3,char c2,char c1,char c0,int start)

{

b[start]=c3;

b[start+1]=c2;

b[start+2]=c1;

b[start+3]=c0;

}

int htob(char \*h,char \*b)

{

int i,j;

int hl=strlen(h);

for(i=0;i<hl;i++)

switch(h[i])

{

case '0':

{

saveB(b,'0','0','0','0',4\*i);

break;

}

case '1':

{

saveB(b,'0','0','0','1',4\*i);

break;

}

case '2':

{

saveB(b,'0','0','1','0',4\*i);

break;

}

case '3':

{

saveB(b,'0','0','1','1',4\*i);

break;

}

case '4':

{

saveB(b,'0','1','0','0',4\*i);

break;

}

case '5':

{

saveB(b,'0','1','0','1',4\*i);

break;

}

case '6':

{

saveB(b,'0','1','1','0',4\*i);

break;

}

case '7':

{

saveB(b,'0','1','1','1',4\*i);

break;

}

case '8':

{

saveB(b,'1','0','0','0',4\*i);

break;

}

case '9':

{

saveB(b,'1','0','0','1',4\*i);

break;

}

case 'A':

{

saveB(b,'1','0','1','0',4\*i);

break;

}

case 'B':

{

saveB(b,'1','0','1','1',4\*i);

break;

}

case 'C':

{

saveB(b,'1','1','0','0',4\*i);

break;

}

case 'D':

{

saveB(b,'1','1','0','1',4\*i);

break;

}

case 'E':

{

saveB(b,'1','1','1','0',4\*i);

break;

}

case 'F':

{

saveB(b,'1','1','1','1',4\*i);

break;

}

}

return 4\*hl;

}

int btoo(char \*b,char \*o,int bl)

{

int i,j;

int ol;

int value;

if(bl%3==0)

ol=bl/3;

else

ol=bl/3+1;

j=bl-1;

for(i=ol-1;i>=0;i--)

{

if(i>0)

o[i]=b[j]-48+(b[j-1]-48)\*2+(b[j-2]-48)\*4+48;

else

{

switch(j)

{

case 2:

o[i]=b[j]-48+(b[j-1]-48)\*2+(b[j-2]-48)\*4+48;

break;

case 1:

o[i]=b[j]-48+(b[j-1]-48)\*2+48;

break;

case 0:

o[i]=b[j];

break;

}

}

j=j-3;

}

return ol;

}

void printO(char \*o,int ol)

{

int i=0;

if(o[0]=='0')

i=1;

for(;i<ol;i++)

{

printf("%c",o[i]);

}

printf("\n");

}

main()

{

char h[MaxSize];

char b[4\*MaxSize];

char o[4\*MaxSize/3+1];

int n,i,bl,ol;

scanf("%d",&n);

getchar();

for(i=0;i<n;i++)

{

gets(h);

bl=htob(h,b);

ol=btoo(b,o,bl);

printO(o,ol);

}

}

Java代码

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

static void shiliuToEr(String s){

StringBuffer sbBuffer=new StringBuffer();

char[] a=s.toCharArray();

for (int i = 0; i < a.length; i++) {

switch (a[i]) {

case '0':

sbBuffer.append("0000");

break;

case '1':

sbBuffer.append("0001");

break;

case '2':

sbBuffer.append("0010");

break;

case '3':

sbBuffer.append("0011");

break;

case '4':

sbBuffer.append("0100");

break;

case '5':

sbBuffer.append("0101");

break;

case '6':

sbBuffer.append("0110");

break;

case '7':

sbBuffer.append("0111");

break;

case '8':

sbBuffer.append("1000");

break;

case '9':

sbBuffer.append("1001");

break;

case 'A':

sbBuffer.append("1010");

break;

case 'B':

sbBuffer.append("1011");

break;

case 'C':

sbBuffer.append("1100");

break;

case 'D':

sbBuffer.append("1101");

break;

case 'E':

sbBuffer.append("1110");

break;

case 'F':

sbBuffer.append("1111");

break;

}

}

ErToBa(sbBuffer);

}

static void ErToBa(StringBuffer sb) {

int len=sb.length();

if (len%3==0) {

if (sb.substring(0, 3).equals("000")){

sb.delete(0, 3);

}

}else if (len%3==1) {

if (sb.substring(0, 1).equals("0")) {

sb.delete(0, 1);

}else {

sb.insert(0, "00");

}

}else {

if (sb.subSequence(0, 2).equals("00")) {

sb.delete(0, 2);

}else {

sb.insert(0, "0");

}

}

StringBuffer sBuffer = new StringBuffer();

String s;

int n=sb.length()/3;

for (int i = 0; i < n; i++) {

s=sb.substring(i\*3, (i+1)\*3);

if (s.equals("000")) {

sBuffer.append("0");

}

if (s.equals("001")) {

sBuffer.append("1");

}

if (s.equals("010")) {

sBuffer.append("2");

}

if (s.equals("011")) {

sBuffer.append("3");

}

if (s.equals("100")) {

sBuffer.append("4");

}

if (s.equals("101")) {

sBuffer.append("5");

}

if (s.equals("110")) {

sBuffer.append("6");

}

if (s.equals("111")) {

sBuffer.append("7");

}

}

System.out.println(sBuffer.toString());

}

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader bReader=new BufferedReader(new InputStreamReader(System.in));

int n=Integer.parseInt(bReader.readLine());

String[] string=new String[n];

for (int i = 0; i < n; i++) {

string[i]=bReader.readLine();

}

for (int i = 0; i < n; i++) {

shiliuToEr(string[i]);

}

}

}