**十进制转十六进制**

问题描述

　　十六进制数是在程序设计时经常要使用到的一种整数的表示方式。它有0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F共16个符号，分别表示十进制数的0至15。十六进制的计数方法是满16进1，所以十进制数16在十六进制中是10，而十进制的17在十六进制中是11，以此类推，十进制的30在十六进制中是1E。  
　　给出一个非负整数，将它表示成十六进制的形式。

输入格式

　　输入包含一个非负整数a，表示要转换的数。0<=a<=2147483647

输出格式

　　输出这个整数的16进制表示

样例输入

30

样例输出

1E

锦囊1

按除16取余倒数（也可使用格式输出）。

锦囊2

可以使用递归来处理，递归参数为要转换的数字，如果该数字大于等于16，则将递归处理该数整除16的值，然后再输出最后一位。 C/C++语言可以使用格式输出直接输出转换后的结果。

C++代码

#include<iostream>

#include<cstdio>

using namespace std;

int main()

{

int n;

char s[100000];

while(cin>>n)

{

int k=0;

if(n==0)

{

cout<<0;

}

else

{

while(n!=0)

{

if(n%16>=10)

{

s[k++]='A'+n%16-10;

}

else

{

s[k++]='0'+n%16;

}

n=n/16;

}

for(int i=k-1;i>=0;i--)

{

cout<<s[i];

}

}

cout<<endl;

}

return 0;

}

C代码

#include <stdio.h>

#include <stdlib.h>

char data[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

//输出

void Print(char \*num,int n)

{

int i;

for(i=n-1;i>=0;i--) printf("%c",num[i]);

printf("\0");

printf("\n");

}

//将十六进制数转换为十进制数

int Transform(char \*num,long long value)

{

int n=0;

while(value>=16)

{

num[n++]=data[value%16];

value/=16;

}

num[n++]=data[value%16];

return n;

}

int main()

{

long long value;

char num[10];

int n;

scanf("%I64d",&value);

n=Transform(num,value);

Print(num,n);

return 0;

}

Java代码

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

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

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

printToHexString(buf.readLine());

}

public static void printToHexString(String s) {

StringBuffer st = new StringBuffer();

int n = Integer.valueOf(s);

if(n == 0) {

System.out.println(0);

return;

}

int m;

while(n > 0) {

m = n % 16;

n = n / 16;

switch(m) {

case 0:

st.insert(0, "0");

break;

case 1:

st.insert(0, "1");

break;

case 2:

st.insert(0, "2");

break;

case 3:

st.insert(0, "3");

break;

case 4:

st.insert(0, "4");

break;

case 5:

st.insert(0, "5");

break;

case 6:

st.insert(0, "6");

break;

case 7:

st.insert(0, "7");

break;

case 8:

st.insert(0, "8");

break;

case 9:

st.insert(0, "9");

break;

case 10:

st.insert(0, "A");

break;

case 11:

st.insert(0, "B");

break;

case 12:

st.insert(0, "C");

break;

case 13:

st.insert(0, "D");

break;

case 14:

st.insert(0, "E");

break;

case 15:

st.insert(0, "F");

break;

}

}

System.out.println(st);

}

}