课程设计报告

设计题目：数的进制转换

学生姓名：黄聖翔

专业：智能科学与技术

班级：22-1班

学号：2022212383

指导老师：赵仲秋

完成日期：2022年12月24日

**合肥工业大学计算机与信息学院**

# 需求和规格说明

（1）将输入的 2 进制数（一个非“ 0” 即“ 1” 的字符串）化为 10 进制数。提示：用字符数组 a 盛放所输入的二进制数；而后从后往前逐一计算每一位的“ 位权” w (2 的 0 次方、2 的 1 次方、...)，再计算“ 位权” 乘以“ 位值” 并累加到一个初值为 0 的变量 value 上，最后输出该value。

（2）如何把 8 进制数或 16 进制数化为 10 进制数。

（3）如何把某一个k 进制的数化为 10 进制数呢？

# （二） 设计

根据题目所述，第一问将二进制数转化为十进制数采用的是“位权法”的算法，同时需注意小数点的问题，即整数部分，从小数点往前数第一位乘2的0次方，第二位乘2的1次方……第n为乘2的n-1次方；小数部分，从小数点往后数第一位乘2的-1次方，第二位乘2的-2次方……第n位乘2的-n次方，再将这些结果相加则可得到十进制的表示法。

第二问也是采用“位权法”，八进制的转换只需在第一问的基础上将2改成8，而16进制还需加一个判断字符是否为A~F的范围，再进行位权计算。

第三问如果使用位权法，则需添加一个判断k与10的关系，因为在10进制以上的数据使用英文字母构成，并且每一位所乘的数的底数要等于k；如果不使用位权法，则可以使用strtol函数，strtol函数的用法是strtol(const char \*nptr, char \*\*endptr, int base)，strtol()会将nptr指向的字符串，根据参数base，按权转化为十进制, 然后返回这个值，参数base的范围为2~36。char \*\*endptr位置为接收这个数据中非法字符，即2进制中的3，八进制中的9等，如果为NULL则不会返回非法字符。

同时，该程序还完成了对非法字符的判断功能。

**（三） 用户手册**

二/八/十六进制转十进制：首先，提示用户输入二/八/十六进制字符；

其次，当用户输入完毕后需在结尾处添加小写字母“e”表示输入的数据结束；

最后，按回车键即可显示十进制的字符

K进制转十进制：首先，提示用户需要输入多少进制的数据；

其次，提示用户输入二/八/十六进制字符；

接着，当用户输入完毕后需在结尾处添加小写字母“e”表示输入的数据结束；

最后，按回车键即可显示十进制的字符

# （四） 调试及测试

在判断输入的合法性时，需要考虑当输入为“.”和“m”时也是合法的，而且要分为十进制以内和十进制以外的进行讨论，因为在十进制以外包含了字母，同时由于使用的是字符数组进行储存，所以需要使用ASCII码将字符型转为整型，再进行比较。

# 运行实例：

1. 对于二进制转换为十进制：

|  |  |  |
| --- | --- | --- |
| 输入二进制 | 11111111.11 | 010 |
| 输出 |  |  |

1. 对于八进制转换为十进制：

|  |  |  |
| --- | --- | --- |
| 输入八进制 | 3506.2 | 251 |
| 输出 |  |  |

3.对于十六进制转换为十进制：

|  |  |  |
| --- | --- | --- |
| 输入十六进制 | 0.2A | A2 |
| 输出 |  |  |

4.对于k进制转换为十进制：

|  |  |  |
| --- | --- | --- |
| 输入k | 5 | 17 |
| 输入数据 | 1223.2 | A45CD |
| 输出 |  |  |

**附录**源程序

（1）

#include<iostream>

#include<string>

#include<math.h>

using namespace std;

int judge(char p[],int n)

{

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

{

if (p[i] == '.')

{

return 1;

}

}

return -1;

}

void fun\_1(char p[],int n)

{

int result = 0;

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

{

result += (p[i] - '0') \* pow(2, n - 1 - i);

}

cout << result;

}

int f(char p[],int n)

{

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

{

if (p[i] == '.')return i;

}

}

void fun\_2(char p[], int n)

{

double result = 0;

int x = f(p, n);

cout<<x<<endl;

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

{

result+= (p[i] - '0') \* pow(2, x-i-1);

}

for (int i = x + 1; i < n; i++)

{

result += (p[i] - '0') \* pow(2, -(i - x));

}

cout << result << endl;

}

int main()

{

char p[100];

int n =0;

for (int i = 0;; i++)

{

cin >> p[i];

if (p[i] >= '2')

{

cout << "输入数据有误" << endl;

break;

}

if (p[i] == '3')

{

cout << "输入结束" << endl;

break;

}

n++;//

}

int y = judge(p,n);

//不含小数

if (y == -1)

{

fun\_1(p,n);

}

//含有小数

else if (y == 1)

{

fun\_2(p, n);

}

return 0;

}

（2）

八进制

#include<iostream>

#include<string>

#include<math.h>

using namespace std;

int judge(char p[],int n)

{

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

{

if (p[i] == '.')

{

return 1;

}

}

return -1;

}

void fun\_1(char p[],int n)

{

int result = 0;

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

{

result += (p[i] - '0') \* pow(8, n - 1 - i);

}

cout << result;

}

int f(char p[],int n)

{

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

{

if (p[i] == '.')return i;

}

}

void fun\_2(char p[], int n)

{

double result = 0;

int x = f(p, n);

cout<<x<<endl;

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

{

result+= (p[i] - '0') \* pow(8, x-i-1);

}

for (int i = x + 1; i < n; i++)

{

result += (p[i] - '0') \* pow(8, -(i - x));

}

cout << result << endl;

}

int main()

{

char p[100];

int n =0;

for (int i = 0;; i++)

{

cin >> p[i];

if (p[i] == '9')

{

cout << "输入结束" << endl;

break;

}

n++;//

}

int y = judge(p,n);

//不含小数

if (y == -1)

{

fun\_1(p,n);

}

//含有小数

else if (y == 1)

{

fun\_2(p, n);

}

return 0;

}

十六进制

#include<iostream>

#include<string>

#include<math.h>

using namespace std;

int judge(char p[],int n)

{

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

{

if (p[i] == '.')

{

return 1;

}

}

return -1;

}

void fun\_1(char p[],int n)

{

int result = 0;

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

{

if (p[i] >= '0' && p[i] <= '9')

result += (p[i] - '0') \* pow(16, n - 1 - i);

else if (p[i] >= 'A' && p[i] <= 'Z')

result += (p[i] - 55) \* pow(16, n - 1 - i);

}

cout << result;

}

int f(char p[],int n)

{

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

{

if (p[i] == '.')return i;

}

}

void fun\_2(char p[], int n)

{

double result = 0;

int x = f(p, n);

cout<<x<<endl;

//小数点前

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

{

if (p[i] >= '0' && p[i] <= '9')

result += (p[i] - '0') \* pow(16, x - i - 1);

else if (p[i] >= 'A' && p[i] <= 'Z')

result += (p[i] - 55) \* pow(16, x - 1 - i);

}

//小数点后

for (int i = x + 1; i < n; i++)

{

if (p[i] >= '0' && p[i] <= '9')

result += (p[i] - '0') \* pow(16, -(i - x));

else if (p[i] >= 'A' && p[i] <= 'Z')

result += (p[i] - 55) \* pow(16, -(i - x));

}

cout << result << endl;

}

int main()

{

char p[100];

int n =0;

for (int i = 0;; i++)

{

cin >> p[i];

if (p[i] == 'm')

{

cout << "输入结束" << endl;

break;

}

n++;//

}

int y = judge(p,n);

//不含小数

if (y == -1)

{

fun\_1(p,n);

}

//含有小数

else if (y == 1)

{

fun\_2(p, n);

}

return 0;

}

（3）

#include<iostream>

using namespace std;

int main()

{

cout << "请输入进制：";

int k;

cin >> k;

char p[100];

int n = 0;

cout << "请输入"<<k<<"进制数：";

for (int i = 0;; i++)

{

cin >> p[i];

if (p[i] == 'm')

{

//cout << "输入结束" << endl;

break;

}

if (k > 0 && k <= 10 && (p[i]-'0') >= k && p[i] != '.' && p[i] != 'm')

{

cout << "输入有误";

n += 1;

break;

}

else if (k > 10 && (p[i] - 'A') >= (k - 10 ) && p[i] != '.' && p[i] != 'm')

{

cout << "输入有误";

n += 1;

break;

}

}

if (n == 0)

{

double result = strtol(p, NULL, k);

cout << "转换为十进制：" << result << endl;

}

return 0;

}