# 实验二

## 题目：设计算法把一个十进制整数转化为二至九进制之间的任一进制数输出。

1. **问题分析**

本程序要求实现不同进制之间相互转化的操作。

要完成该实验任务，必须完成如下两个子任务：

①读取输入的整数

②转化成其他进制

测试数据设计为如下几种情况：

①读取错误

②没有转化成需要的进制

1. **概要设计**

为了实现上述程序功能，需要：

①读取输入的整数

②转化成需要的进制

③输出该进制下对应的数字

1. **详细设计**

本程序包含九个函数：

①主函数main()

②二进制函数Two()

③三进制函数Three()

④四进制函数Four()

⑤五进制函数Five()

⑥六进制函数Six()

⑦七进制函数Seven()

⑧八进制函数Eight()

⑨九进制函数Nine()

各函数间关系如下：

main()

Two() Three() Four() Five() Six() Seven() Eight() Nine()

1. **调试分析**
2. 遇到的问题及解决方案

在该实验过程中，遇到的一个问题为：在每次进制转化结束后末尾会多一个零。经过对代码的调试发现是因为c的值多运行了一次。解决方法为将for循环的判断条件由c>=0改为c>0即可解决。

1. 算法的时空分析

该程序的时间复杂度为Tn=O(n²)。

1. 改进设想

虽然自定义了八个函数，但是每个函数大同小异，觉得可以用循环的嵌套来解决自定义函数的编写。

1. **用户使用说明**

程序名为实验二.exe，运行环境为DOS，运行程序后显示：

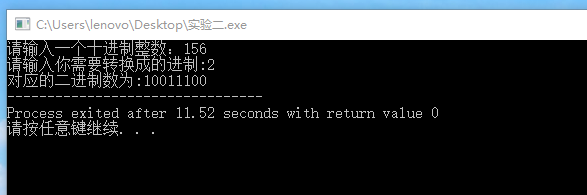
请输入一个十进制整数：

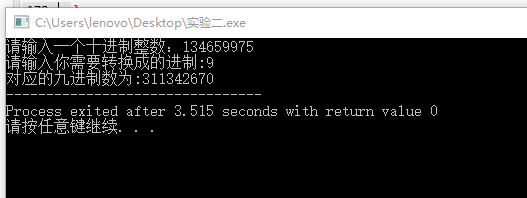
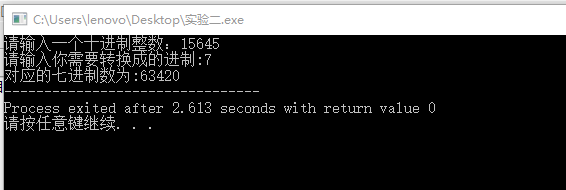
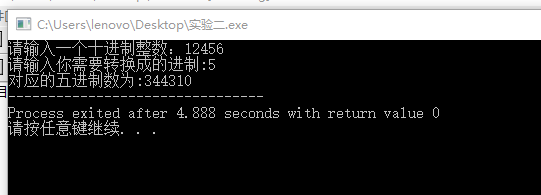
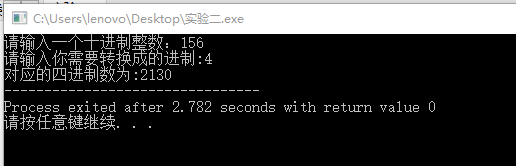
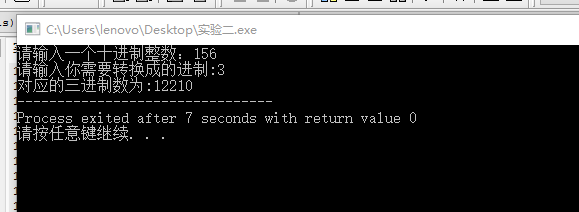
在用户输入过整数后，程序会显示：

请输入你需要转换成的进制:

用户输入需要转化的进制类型即可显示对应的进制。

1. **测试结果**





1. **附录**

#include <stdio.h>

#define Max 100

int Two(int i){

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){ //转化为二进制

b=i%2;

a[c]=b;

c++;

i=i/2;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的二进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int Three(int i){

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){ //转化为三进制

b=i%3;

a[c]=b;

c++;

i=i/3;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的三进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int Four(int i){

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){ //转化为四进制

b=i%4;

a[c]=b;

c++;

i=i/4;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的四进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int Five(int i){

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){ //转化为五进制

b=i%5;

a[c]=b;

c++;

i=i/5;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的五进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int Six(int i){

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){ //转化为六进制

b=i%6;

a[c]=b;

c++;

i=i/6;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的六进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int Seven(int i){

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){ //转化为七进制

b=i%7;

a[c]=b;

c++;

i=i/7;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的七进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int Eight(int i){ //转化为八进制

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){

b=i%8;

a[c]=b;

c++;

i=i/8;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的八进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int Nine(int i){

int a[Max]; //用于存放转换进制后的数

int b; //判断条件的数

int c = 0; //该数储存在数组中的位置

while(i != 0){ //转化为九进制

b=i%9;

a[c]=b;

c++;

i=i/9;

}

c--; //因为c代表存入数据的长度而下表范围是0~c-1

printf("对应的九进制数为:");

for(;c>0;c--){

printf("%d",a[c]);

}

return 0;

}

int main(){

int i = 0; // 输入的十进制整数

int x = 0;

printf("请输入一个十进制整数：");

scanf("%d",&i);

printf("请输入你需要转换成的进制:");

scanf("%d",&x);

switch(x){

case 2:{

printf("%d",Two(i));

break;

}

case 3:{

printf("%d",Three(i));

break;

}

case 4:{

printf("%d",Four(i));

break;

}

case 5:{

printf("%d",Five(i));

break;

}

case 6:{

printf("%d",Six(i));

break;

}

case 7:{

printf("%d",Seven(i));

break;

}

case 8:{

printf("%d",Eight(i));

break;

}

case 9:{

printf("%d",Nine(i));

break;

}

}

return 0;

}