高精度n进制转m进制(n,m≤62)

1.支持n，m在62以内任意进制互相转换。n，m需要输入或设置

2.这个需要字符串没有正负号，没有开头多余的0

3.这个代码假定大写字母A - Z表示10 - 35，小写字母a - z表示36 - 61。考试时应该按题意进行修改

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>

int get_num(char c){ //假定A ~ Z表示10 ~ 35， a ~ z表示36 ~ 61
    if(c >= '0' && c <= '9') return c - '0';
    if(c >= 'A' && c <= 'Z') return c - 'A' + 10;
    return c - 'a' + 36;
}
char get_char(int x){
    if(x >= 0 && x <= 9) return x + '0';
    if(x >= 10 && x <= 35) return x - 10 + 'A';
    return x - 36 + 'a';
}
char s[100005];
int a[100005], ans[100005];
int n, m;

void Trans(){
    int len, Top_bit = 1, Quotient, i, j;
    len = strlen(s);
    for(i = 0; i < len; i++) a[i] = get_num(s[i]);
    memset(ans, 0, sizeof(ans));
    for(i = 0; i < len; i++){
        Quotient = a[i];
        for(j = 0; j < Top_bit; j++){
            ans[j] *= n;
            ans[j] += Quotient;
            Quotient = ans[j] / m;
            ans[j] %= m;
        }
        while(Quotient != 0){
            ans[Top_bit] = Quotient;
            Quotient = ans[Top_bit] / m;
            ans[Top_bit] %= m;
            Top_bit++;
        }
    }
    for(i = Top_bit - 1; i >= 0; i--) putchar(get_char(ans[i]));
}
int main(){
    scanf("%d%d%s", &n, &m, s);
    Trans();
    return 0;
}
```