USA COMPUTING OLYMPIAD

Palindromic Squares

Russ Cox

We generate all the squares from 1 to 300 and check to see which are palindromes.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <ctype.h>
#include <math.h>

/* is string s a palindrome? */
int
ispal(char *s)
{
    char *t;

    t = s+strlen(s)-1;
    for(t=s+strlen(s)-1; s<t; s++, t--)
        if(*s != *t)
            return 0;

    return 1;
}

/* put the base b representation of n into s: 0 is represented by "" */

void
numbconv(char *s, int n, int b)
{
    int len;

    if(n == 0) {
        strcpy(s, "");
        return;
    }

    /* figure out first n-1 digits */
    numbconv(s, n/b, b);

    /* add last digit */
    len = strlen(s);
    s[len] = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"[n%b];
    s[len+1] = '\0';
}

void
main(void)
{
    char s[20];
    char t[20];
    int i, base;
    FILE *fin, *fout;

    fin = fopen("palsquare.in", "r");
    fout = fopen("palsquare.out", "w");
    assert(fin != NULL && fout != NULL);
    fscanf(fin, "%d", &base);
    for(i=1; i <= 300; i++) {
        numbconv(s, i*i, base);
        if(ispal(s)) {
            numbconv(t, i, base);
            fprintf(fout, "%s %s\n", t, s);
        }
    }
    exit(0);
}
```

USACO Gateway | Comment or Question