

汉字花样显示

3140103541 张昌琳

概览

本程序介绍.....	1
原理	1
区位码.....	1
ZB 码	2
16x16 点阵汉字.....	2
程序结构.....	3
使用说明.....	3
显示结果展示.....	3
源代码	7

本程序介绍

- 1、采用拼音输入法
- 2、支持 ASCII 码和汉字混合显示
- 3、可以选择显示字库和显示模式
显示字库包括：hzk16 和 mhzk
显示模式包括：正常显示\放大一倍\斜体显示\镜像显示\倒立显示
- 4、依赖文件为 "Hzk16.字库", "mhzk16.字库", 和"汉字拼音五笔表.txt"放置于当前文件夹，若修改了文件路径，请修改程序中 PATH1，PATH2 的宏定义

原理

区位码

将 ASCII 码和汉字编码分区和位两个标记。

要显示一个汉字要先找到它的编码，先找到它的区，再找到它的位置，从而得到它的顺序，可以按顺序从字库中读出。

程序结构



使用说明

- 1、运行程序进入菜单页面，根据提示选择想进行的操作
 1. 选择显示字库
 2. 选择显示模式
- 2、输入欲显示汉字的拼音
- 3、在屏幕呈现的汉字选项中选择欲显示汉字对应编号
- 4、屏幕显示汉字

显示结果展示

模式选择界面展示：

```

请选择显示字库:
1. Hzk16
2. mhzk16
1
请选择显示模式:
1. 正常显示
2. 放大一倍
3. 斜体显示
4. 镜像显示
5. 倒立显示
3

```

拼音输入法展示如下（键入“a”会显示拼音中包含 a 的所有字）：

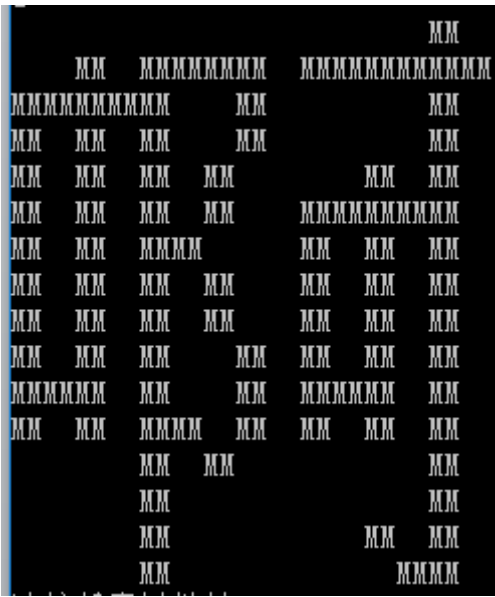
```

请输入拼音:
a
1啊 2阿 3吖 4腌 5锒 6挨 7哎 8唉 9哀 10埃
11皑 12癌 13藺 14矮 15艾 16碍 17爱 18隘 19捩 20暖
21媛 22瑗 23媛 24僇 25郃 26氮 27鞍 28安 29俺 30按
31暗 32岸 33胺 34案 35谖 36埤 37揩 38犴 39庵 40按
41铵 42鹁 43黯 44肮 45昂 46盎 47凹 48敖 49熬 50翱
51袄 52傲 53奥 54懊 55澳 56坳 57拗 58嗽 59舄 60廌
61遨 62媪 63鳌 64鳌 65熬 66犛 67熬 68熬 69麀 70芭
71捌 72扒 73叭 74吧 75芭 76八 77疤 78巴 79拔 80跋
81靶 82把 83耙 84坝 85霸 86罢 87爸 88菱 89菝 90芭
91霸 92钹 93耙 94白 95柏 96百 97摆 98佰 99败 100拜
101裨 102裨 103辮 104斑 105班 106搬 107扳 108般 109颀 110板
111版 112扮 113拌 114伴 115瓣 116半 117办 118绊 119阪 120坂
121钣 122瘢 123舩 124邦 125帮 126梆 127榜 128膀 129绑 130棒
131磅 132蚌 133镑 134傍 135谤 136菁 137浜 138胞 139包 140褒
141剥 142苞 143薄 144雹 145保 146堡 147饱 148宝 149抱 150报
151暴 152豹 153鲍 154爆 155葆 156孢 157煲 158鸫 159褓 160趵
161皤
请选择输入汉字
1

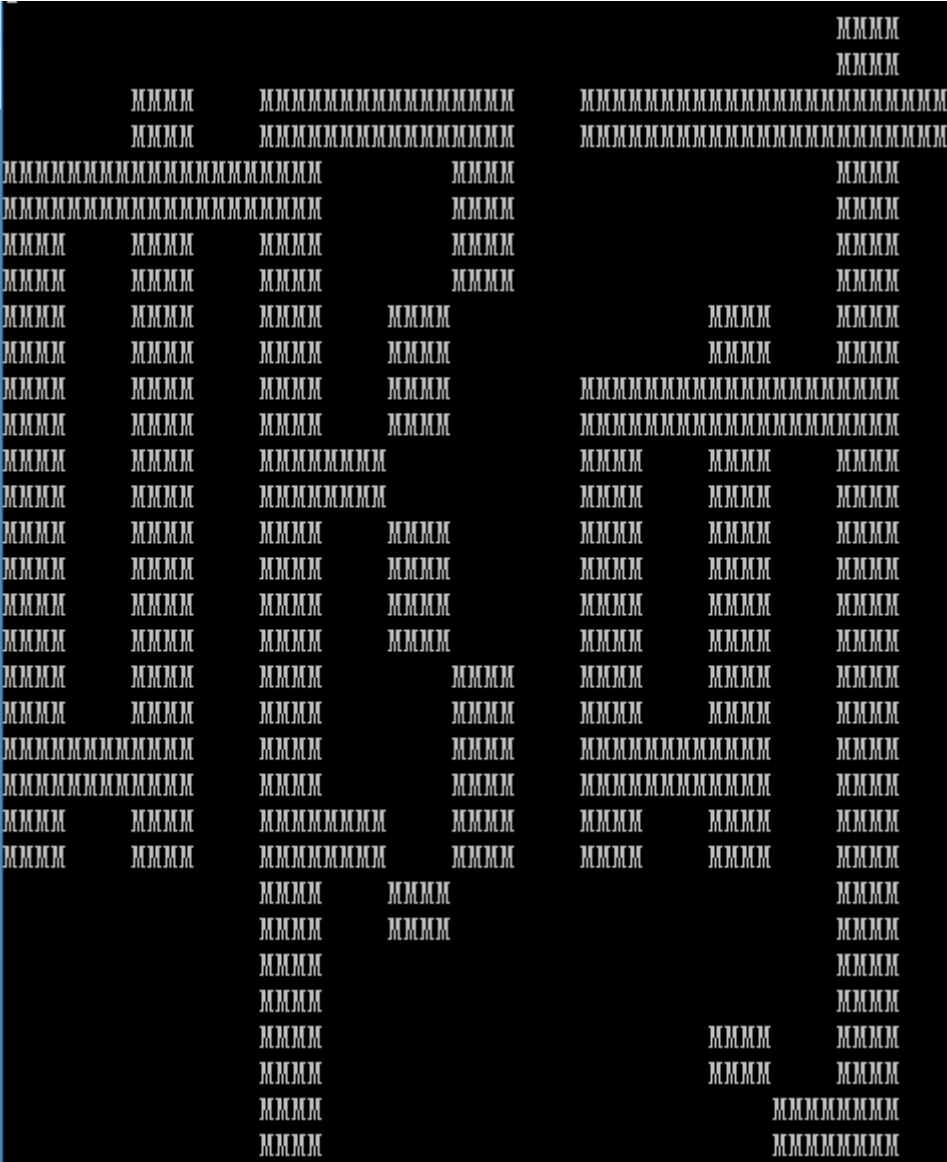
```

因为 GB 码和 ZB 码同一显示模式下显示的结果是一样的，只是编码方式和使用字库不一样，因此每个显示模式只贴一个图。

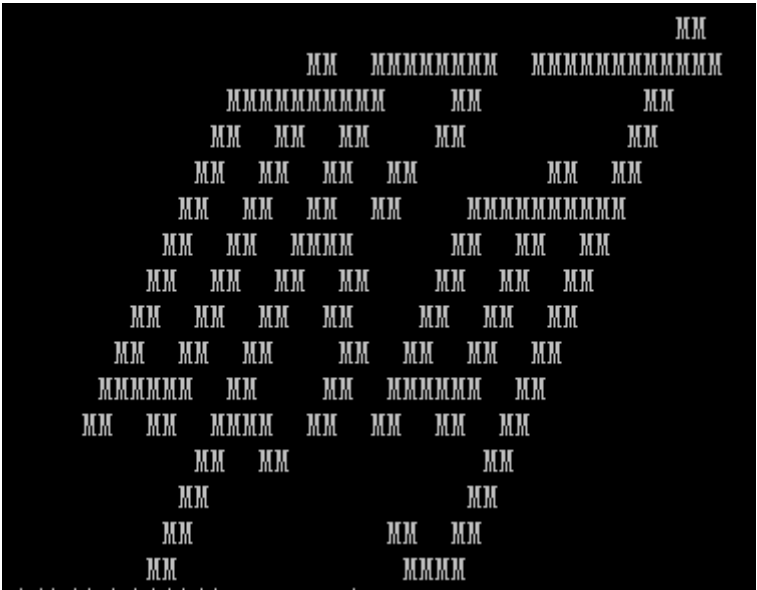
正常显示：



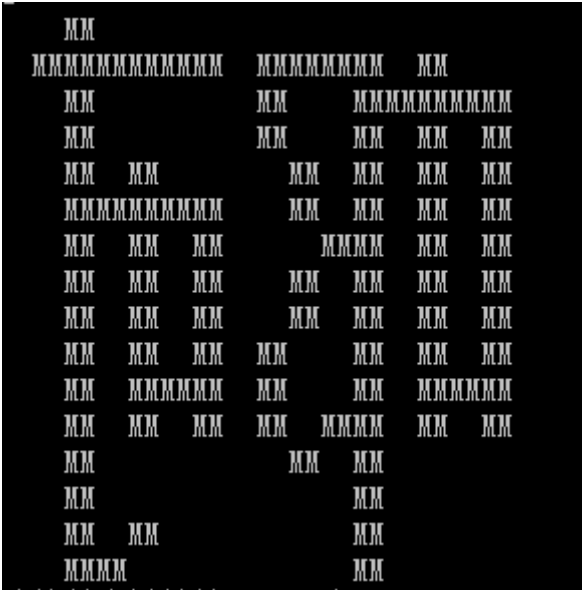
放大一倍：



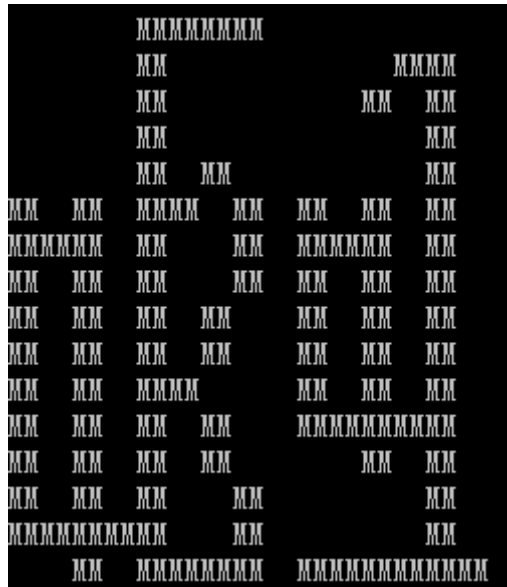
斜体显示:



镜像显示:



倒立显示



源代码

```
/*
说明：
1、采用拼音输入法
2、支持ASCII码和汉字混合显示
3、可以选择显示字库和显示模式
4、依赖文件为“Hzk16.字库”，“mhzk16.字库”，和“汉字拼音五笔表.txt”放置于当前文件夹，
若修改了文件路径，请修改程序中PATH1，PATH2的宏定义
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define PATH1 "Hzk16.字库"
#define PATH2 "mhzk16.字库"
#define PATH3 "汉字拼音五笔表.txt"
#define MAX_LINE 512
#define MAX_PINYIN 10

//为了方便变量都用全局的了
int q, w, adr;
int m, i, j, k, p;
char u[32];
short t[16];
```

```

char s[32];
FILE *fp;
char buf[MAX_LINE]; //缓冲区
char pinyin[MAX_PINYIN];
int len, flag, count, choice; //len为行字符个数
char *appear;

//拼音输入
void pinyinInput();
//GB码及Hzk字库读取点阵
void GBcode();
//GB码转ZB码及mhzk字库读取点阵
void GBtoZB();
//花式打印点阵函数
void printmodel(); //正常显示
void printmode2(); //放大一倍
void printmode3(); //斜体显示
void printmode4(); //镜像显示
void printmode5(); //倒立显示

int main() {
    int ziku, mode, breakflag=0;

    printf("请选择显示字库: \n1. Hzk16\n2. mhzk16\n");
    scanf("%d", &ziku);
    printf("请选择显示模式: \n1. 正常显示\n\
2. 放大一倍\n3. 斜体显示\n4. 镜像显示\n5. 倒立显示\n");
    scanf("%d", &mode);

    pinyinInput();
    GBcode();
    switch(ziku) {
        case 1:
            break;
        case 2:
            GBtoZB();
            break;
    }

    switch(mode) {
        case 1:
            printmodel();
            break;
        case 2:

```



```

        printmode2();
        break;
    case 3:
        printmode3();
        break;
    case 4:
        printmode4();
        break;
    case 5:
        printmode5();
        break;
}

return 0;
}

//拼音输入法
void pinyinInput() {
    printf("请输入拼音: \n");
    scanf("%s", pinyin);

    fp=fopen(PATH3, "rb");
    flag=0;
    count=0;
    while(fgets(buf, MAX_LINE, fp) != NULL) {
        len = strlen(buf);
        buf[len-1] = '\0'; //去掉换行符
        appear=strstr(buf, pinyin);
        if(appear!=NULL) {
            flag=1;
        }
        if(flag==1) {
            if(appear==NULL) {
                flag=-1;
            }
            else{
                count++;
                buf[2] = '\0';
                printf("%d%s ", count, buf);
                if(count%10==0) printf("\n");
            }
        }
        if(flag==-1) break;
    }
}

```

```

printf("\n");
fclose(fp);
if(flag==0) {
    printf("输入拼音不存在\n");
}else{
    printf("请选择输入汉字\n");
    scanf("%d",&choice);
    fp=fopen(PATH3,"rb");
    flag=0;
    count=0;
    while(fgets(buf,MAX_LINE,fp) != NULL) {
        len = strlen(buf);
        buf[len-1] = '\0'; //去掉换行符
        appear=strstr(buf, pinyin);
        if(appear!=NULL) {
            flag=1;
        }
        if(flag==1) {
            if(appear!=NULL) {
                count++;
                if(count==choice) {
                    s[0]=buf[0];
                    s[1]=buf[1];
                    s[2]='\0';
                    flag=-1;
                }
            }
        }
        if(flag==-1) break;
    }
}
fclose(fp);
}

```

//GB码及Hzk字库读取点阵

```

void GBcode() {
    q=(s[0]-0xA1)&0xFF;
    w=(s[1]-0xA1)&0xFF;
    adr=(q*94+w)*32;

    fp=fopen(PATH1,"rb");
    if(!fp) {
        printf("文件无法打开\n");
        exit(1);
    }
}

```

```

    }

    fseek(fp, adr, SEEK_SET);
    fread(u, 1, 32, fp);
    fclose(fp);
}

//国标GB到哲标ZB的转换及mhzk字库读取
void GBtoZB() {
    if((s[0]&0x80)==0) { //AScii
        t[0]=t[0]&0x0F;
        t[0]=t[0]&(s[0]);
    } else { //HanZi
        q =(s[0]-0xA1)&0xFF; //Qu
        w =(s[1]-0xA1)&0xFF; //Wei
        m=0x2000+q*94+w; //ZB码
        t[0]=((short)m);
    }

    //检测哪一类字符：符号、汉字、显示字符等。
    if((t[0]&0xE000)==0) { //AScii、其它符号
        adr=adr; //字库中位置
    } else if((t[0]&0x8000)==0) { //汉字字符
        adr=(t[0]-0x2000+256)*32; //前256个字符是ascii码
    }

    //mhzk字库读取
    fp=fopen(PATH2, "rb");
    if(!fp) {
        printf("文件无法打开2\n");
        exit(1);
    }

    fseek(fp, adr, SEEK_SET);
    fread(u, 1, 32, fp);
    fclose(fp);
}

//正常显示
void printmodel() {
    for(j=0; j<16; j++) {
        m=0x80;
        for(i=0; i<8; i++) {
            if(m&u[j*2]) {
                printf("MM");
            }
        }
    }
}

```

```

        else{
            printf("  ");
        }
        m>>=1;
    }
    m=0x80;
    for(i=0;i<8;i++){
        if(m&u[j*2+1]){
            printf("MM");
        }
        else{
            printf("  ");
        }
        m>>=1;
    }
    printf("\n");
}

return;
}

```

//放大一倍

```

void printmode2() {
    for(j=0;j<16;j++){
        m=0x80;
        for(i=0;i<8;i++){
            if(m&u[j*2]){
                printf("MMMM");
            }
            else{
                printf("    ");
            }
            m>>=1;
        }
        m=0x80;
        for(i=0;i<8;i++){
            if(m&u[j*2+1]){
                printf("MMMM");
            }
            else{
                printf("    ");
            }
            m>>=1;
        }
    }
}

```

```

printf("\n");
m=0x80;
for(i=0;i<8;i++){
    if(m&u[j*2]){
        printf("MMM");
    }
    else{
        printf(" ");
    }
    m>>=1;
}
m=0x80;
for(i=0;i<8;i++){
    if(m&u[j*2+1]){
        printf("MMM");
    }
    else{
        printf(" ");
    }
    m>>=1;
}
printf("\n");
}

return;
}

//斜体显示
void printmode3() {
    for(j=0;j<16;j++){
        for(p=0;p<16-j;p++){
            printf(" "); //这里就是产生斜体的trick啦
        }
        m=0x80;
        for(i=0;i<8;i++){
            if(m&u[j*2]){
                printf("MM");
            }
            else{
                printf(" ");
            }
            m>>=1;
        }
        m=0x80;
        for(i=0;i<8;i++){

```

```

        if(m&u[j*2+1]) {
            printf("MM");
        }
        else{
            printf("  ");
        }
        m>>=1;
    }
    printf("\n");
}

return;
}

```

//镜像显示

```

void printmode4() {
    for(j=0;j<16;j++) {
        m=0x01;
        for(i=0;i<8;i++) {
            if(m&u[j*2+1]) {
                printf("MM");
            }
            else{
                printf("  ");
            }
            m<<=1;
        }
        m=0x01;
        for(i=0;i<8;i++) {
            if(m&u[j*2]) {
                printf("MM");
            }
            else{
                printf("  ");
            }
            m<<=1;
        }
        printf("\n");
    }

    return;
}

```

//倒立显示

```
void printmode5() {  
    for(j=16; j>0; j--) {  
        m=0x80;  
        for(i=0; i<8; i++) {  
            if(m&u[j*2]) {  
                printf("MM");  
            }  
            else {  
                printf("  ");  
            }  
            m>>=1;  
        }  
        m=0x80;  
        for(i=0; i<8; i++) {  
            if(m&u[j*2+1]) {  
                printf("MM");  
            }  
            else {  
                printf("  ");  
            }  
            m>>=1;  
        }  
        printf("\n");  
    }  
  
    return;  
}
```