

## 第八次作业大程 Report

### 1. 源代码

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  int main(void)
4.  {
5.      FILE* fphzk = NULL;
6.      FILE* fp = fopen("output.txt", "w");
7.      int i, j, k, offset, operation;
8.      int flag;
9.      unsigned char buffer[32];
10.     unsigned char word[1000][3];
11.     unsigned char key[8] = {
12.         0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01
13.     };
14.     fphzk = fopen("hzk16", "rb");
15.     if (fphzk == NULL) {
16.         fprintf(stderr, "error hzk16\n");
17.         return 1;
18.     }
19.     //处理汉字
20.     printf("请输入汉字:\n");
21.     for (int m = 0;; m++) {
22.         fgets(word[m], 3, stdin);
23.         word[m][3] = 0;
24.         if (word[m][0] == '\n')
25.             break;
26.     }
27.     for (int m = 0;; m++)
28.     {
29.         if (word[m][0] == '\n')
30.             break;
31.         offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word[m][1] -
0xa0 - 1)) * 32;
32.         fseek(fphzk, offset, SEEK_SET);
33.         fread(buffer, 1, 32, fphzk);
34.         for (k = 0; k < 16; k++) {
35.             for (j = 0; j < 2; j++) {
36.                 for (i = 0; i < 8; i++) {
```

```

37.             flag = buffer[k * 2 + j] & key[i];
38.             printf("%s", flag ? "●" : "○");
39.         }
40.     }
41.     printf("\n");
42. }
43.     printf("\n");
44. }
45.
46. //选择模式
47. while (1)
48. {
49.     printf("请选择模式: \n0.退出\n1.文件输出\n2.放大\n3.斜体\n4.倒立\n5.变色\n6.动态显示\n");
50.     scanf("%d", &operation);
51.     system("cls");
52.     system("color 0F");
53.     switch (operation)
54.     {
55.     case 0:
56.         return 0;
57.     case 1:
58.         for (int m = 0;; m++)
59.         {
60.             if (word[m][0] == '\n')
61.                 break;
62.             offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
[m][1] - 0xa0 - 1)) * 32;
63.             fseek(fphzk, offset, SEEK_SET);
64.             fread(buffer, 1, 32, fphzk);
65.             for (k = 0; k < 16; k++)
66.             {
67.                 for (j = 0; j < 2; j++) {
68.                     for (i = 0; i < 8; i++) {
69.                         flag = buffer[k * 2 + j] & key[i];
70.                         fprintf(fp, "%s", flag ? "● " : "○ ");
71.                     }
72.                 }
73.                 fprintf(fp, "\n");
74.             }
75.             printf("\n");
76.         }
77.         break;
78.     case 2:

```

```

79.         for (int m = 0;; m++)
80.         {
81.             if (word[m][0] == '\n')
82.                 break;
83.             offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
[m][1] - 0xa0 - 1)) * 32;
84.             fseek(fphzk, offset, SEEK_SET);
85.             fread(buffer, 1, 32, fphzk);
86.             for (k = 0; k < 16; k++)
87.             {
88.                 for (j = 0; j < 2; j++) {
89.                     for (i = 0; i < 8; i++) {
90.                         flag = buffer[k * 2 + j] & key[i];
91.                         printf("%s", flag ? "●●" : "○○");
92.                     }
93.                 }
94.                 printf("\n");
95.                 for (j = 0; j < 2; j++) {
96.                     for (i = 0; i < 8; i++) {
97.                         flag = buffer[k * 2 + j] & key[i];
98.                         printf("%s", flag ? "●●" : "○○");
99.                     }
100.                }
101.                printf("\n");
102.            }
103.            printf("\n\n");
104.        }
105.        break;
106.    case 3:
107.        for (int m = 0;; m++)
108.        {
109.            if (word[m][0] == '\n')
110.                break;
111.            offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
[m][1] - 0xa0 - 1)) * 32;
112.            fseek(fphzk, offset, SEEK_SET);
113.            fread(buffer, 1, 32, fphzk);
114.            for (k = 0; k < 16; k++) {
115.                for (int n = 0; n < 16-k; n++)
116.                {
117.                    printf(" ");
118.                }
119.                for (j = 0; j < 2; j++) {
120.                    for (i = 0; i < 8; i++) {

```

```

121.                flag = buffer[k * 2 + j] & key[i];
122.                printf("%s", flag ? "●" : "○");
123.            }
124.        }
125.        printf("\n");
126.    }
127.    printf("\n");
128.}
129.    break;
130.    case 4:
131.        for (int m = 0;; m++)
132.        {
133.            if (word[m][0] == '\n')
134.                break;
135.            offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
[m][1] - 0xa0 - 1)) * 32;
136.            fseek(fphzk, offset, SEEK_SET);
137.            fread(buffer, 1, 32, fphzk);
138.            for (k = 15; k >= 0; k--) {
139.                for (j = 1; j >= 0; j--) {
140.                    for (i = 7; i >= 0; i--) {
141.                        flag = buffer[k * 2 + j] & key[i];
142.                        printf("%s", flag ? "●" : "○");
143.                    }
144.                }
145.                printf("\n");
146.            }
147.            printf("\n");
148.        }
149.        break;
150.    case 5:
151.        system("color 0C");
152.        for (int m = 0;; m++)
153.        {
154.            if (word[m][0] == '\n')
155.                break;
156.            offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
[m][1] - 0xa0 - 1)) * 32;
157.            fseek(fphzk, offset, SEEK_SET);
158.            fread(buffer, 1, 32, fphzk);
159.            for (k = 0; k < 16; k++) {
160.                for (j = 0; j < 2; j++) {
161.                    for (i = 0; i < 8; i++) {
162.                        flag = buffer[k * 2 + j] & key[i];

```

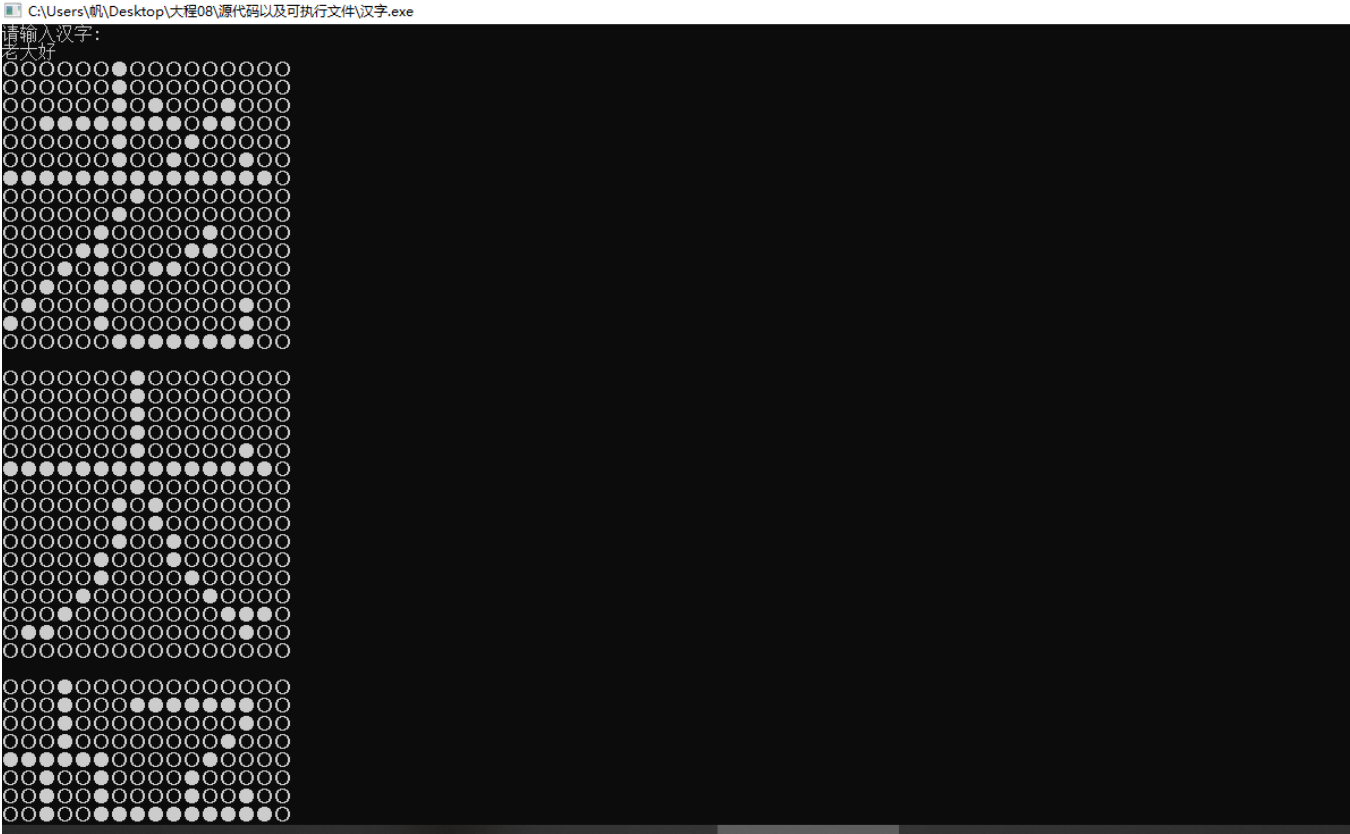
```

163.             printf("%s", flag ? "●" : "○");
164.         }
165.     }
166.     printf("\n");
167. }
168.     printf("\n");
169. }
170.     case 6:
171.         for (int n = 1; n <= 20; n++)
172.         {
173.             for (int m = 0;; m++)
174.             {
175.                 if (word[m][0] == '\n')
176.                     break;
177.                 offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (
word[m][1] - 0xa0 - 1)) * 32;
178.                 fseek(fphzk, offset, SEEK_SET);
179.                 fread(buffer, 1, 32, fphzk);
180.                 for (k = 0; k < 16; k++) {
181.                     for (j = 0; j < 2; j++) {
182.                         for (i = 0; i < 8; i++) {
183.                             flag = buffer[k * 2 + j] & key[i];
184.                             printf("%s", flag ? "●" : "○");
185.                         }
186.                     }
187.                     printf("\n");
188.                 }
189.                 printf("\n");
190.             }
191.             _sleep(100);
192.             system("cls");
193.             _sleep(100);
194.         }
195.
196.     default:
197.         break;
198. }
199. }
200. fclose(fphzk);
201. fphzk = NULL;
202. fclose(fp);
203. system("pause");
204. return 0;
205. }

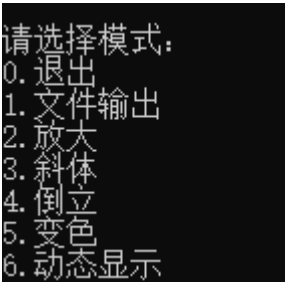
```

## 2. 结果展示

### 1) 基本功能：输入汉字（小于 1000 个）

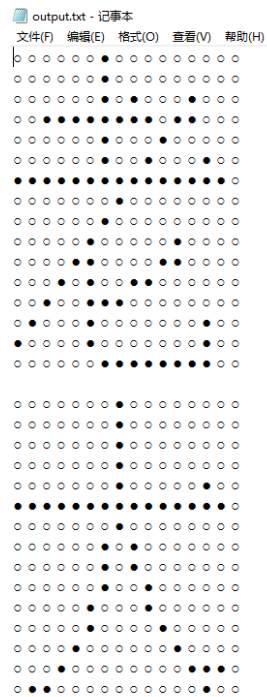


所有功能如下

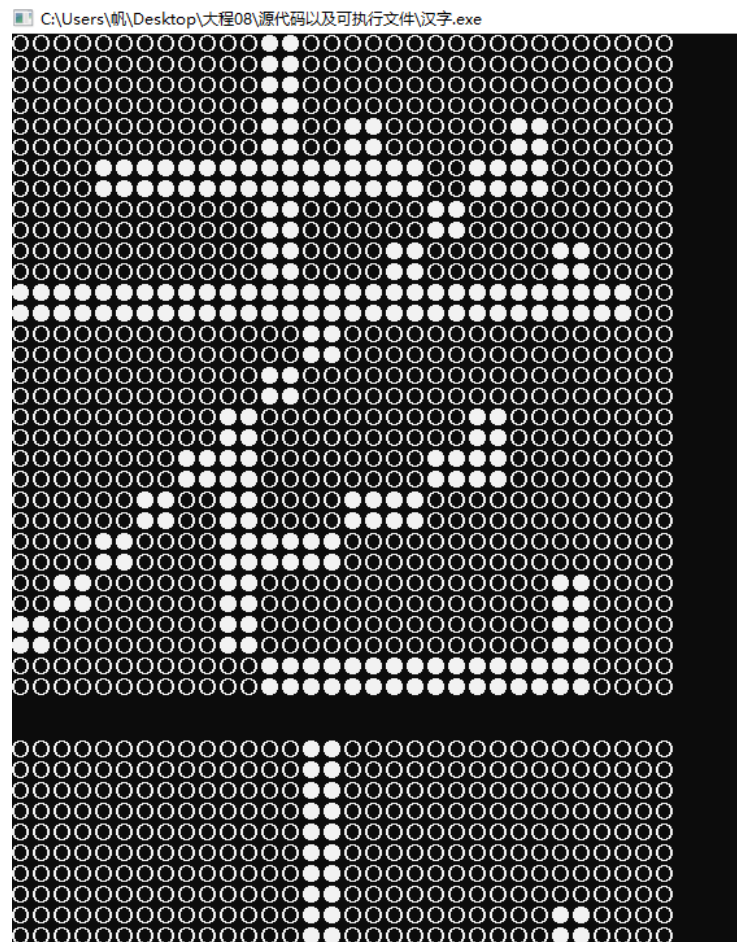


## 2) 功能扩展：文件输出

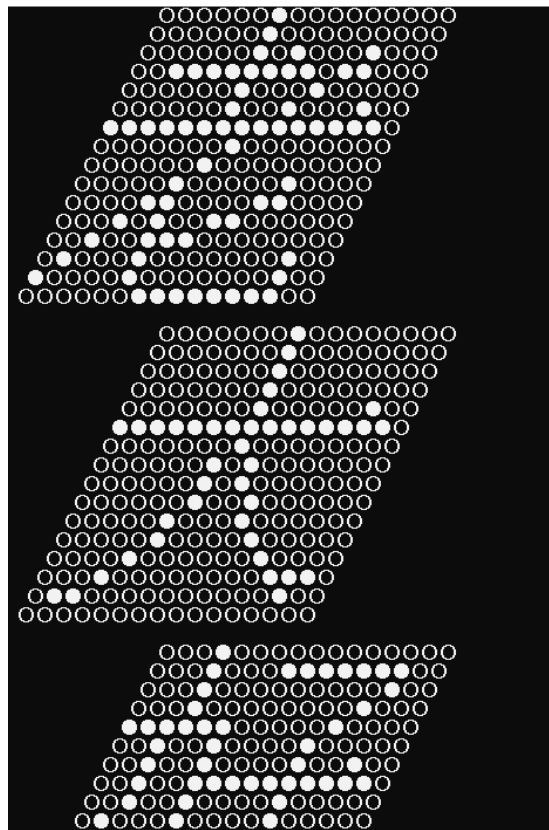
如图选择 1 后可以输出到 output.txt:



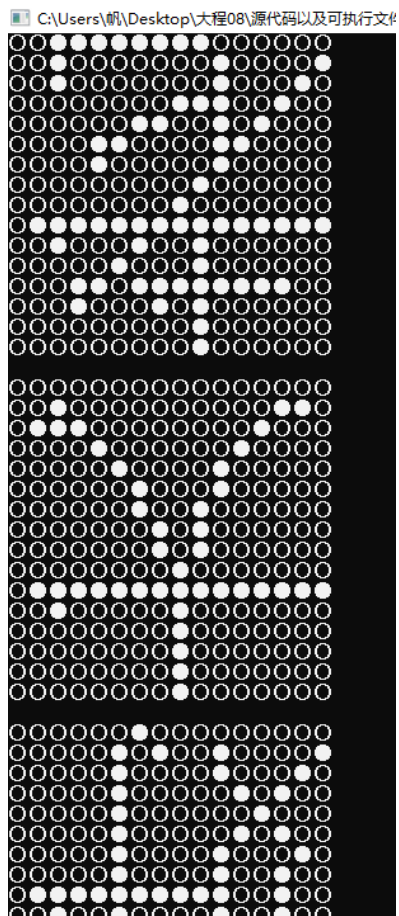
## 3) 功能扩展：放大



4) 功能扩展：斜体

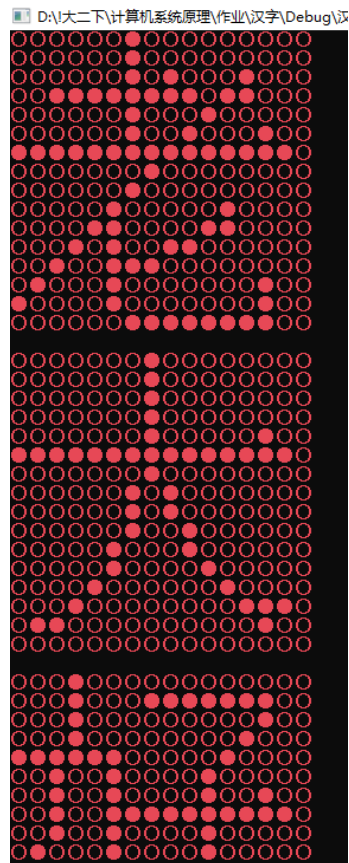


5) 功能扩展：倒立





#### 6) 功能扩展：变色



#### 7) 功能扩展：动态显示

这个功能需要通过在程序中具体体验才可以体会到，欢迎打开汉字显示.exe 进行实际操作。

## 3. 总结

此次进行汉字显示的大程收获颇多，了解、理解了汉字库以及汉字显示的实际过程。并通过多种方式进行汉字显示，收获了乐趣以及经验，对未来计算机系统原理课程充满了兴趣以及期待。