**任务1：英雄出世**

炎热的夏天午后，小明正在百无聊赖地写c语言程序。忽然，电脑屏幕一阵抖动，浮现下面18×18个看似杂乱无章的数字：

32, 32, 32, 32, 32, 32, 32, 32, 32, 95, 45, 126, 126, 126, 126, 126, 45, 95,

32, 32, 32, 32, 32, 32, 32, 32, 10, 32, 32, 32, 32, 32, 32, 32, 32, 32,

-93, -34, 45, 45, 45, -93, -34, 45, 32, 32, 32, 92, 32, 32, 32, 32, 32, 10,

32, 32, 32, 32, 32, 32, 32, 32, 124, -93, -83, 32, 32, -93, -83, 32, 96, 92,

32, 32, 32, 124, 32, 32, 32, 32, 10, 32, 32, 32, 32, 32, 32, 32, 124, 32,

-93, -64, -95, -95, -93, -64, 32, 32, 32, 124, 47, 126, 92, 32, 32, 32, 32, 10,

32, 32, 32, 32, 44, 126, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,

32, 32, 32, 124, 32, 32, 32, 32, 10, 32, 32, 32, 32, 124, 32, 32, 32, 32,

79, 32, 32, 32, 32, 32, 32, 32, 32, 32, 95, 45, 126, 32, 32, 32, 32, 10,

32, 32, 32, 32, 32, 126, 45, 95, 95, 95, 95, 95, 95, 95, 95, 95, 45, 45,

126, 32, 32, 126, 92, 32, 32, 32, 10, 32, 32, 32, 47, 32, 32, 44, 32, 32,

32, 46, 32, 32, 32, 32, 32, 46, 32, 32, 32, 92, 32, 32, 92, 32, 32, 10,

32, 32, 47, 95, 44, 39, 124, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,

32, 32, 124, 96, 95, 95, 39, 32, 10, 32, 32, 32, 32, 32, 32, 124, 32, 32,

32, 124, 96, 46, 94, 46, 39, 124, 32, 32, 32, 124, 32, 32, 32, 32, 32, 10,

32, 32, 32, 32, 32, 32, 124, 32, 32, 32, 44, 40, 32, 32, 41, 47, 32, 92,

32, 32, 92, 95, 32, 32, 32, 32, 10, 32, 32, 32, 32, 32, 47, 95, 95, -93,

-81, 32, 32, 96, 39, 32, 32, 32, 32, -93, -36, 95, 95, 92, 32, 32, 32, 10

同时，天空飘下一张小纸条，上面写着一段神谕：“宽宏的数组，将容纳所见的一切；万能的指针，将把所见的一切化为字符；而后，救世主即将诞生！”。

聪明的你，请帮帮惊慌失措的小明，一起揭开救世主的神秘面纱吧……

*任务1的要求*：

在main函数中，利用数组与指针操作，将救世主的模样正确显示在屏幕上。

**任务2：解救神兽**

就在救世主面世的一刹那，电闪雷鸣……小明穿越了……成了救世主……

救世主当然不可能是孤独的，他有一只强大的助力兼小宠物——神兽。但是，在上古大战中，神兽被邪恶势力封印在一个叫“pet.txt”的文本文件里了。小明回想起救世主诞生的过程，神兽似乎也可以用类似的方式来唤醒，然而事情真有这么简单吗？小明打开文件后愣住了……

此时，冥冥中似乎有个惹人嫌的家伙夹杂着中英文唠叨什么“16210”，这对小明会有帮助吗？

另外，在救世主的记忆里，想容纳完整的神兽，似乎需要22×46大小的空间，这点也许对保留神兽很重要。

*任务2的要求*：

实现函数void showFile(char \* filename, char \*arr, int h, int w)读取pet.txt文件，其中参数fileName的实参应为文件路径名，arr为二维数组首元素的地址，h为高，w为宽，利用在main函数定义的二维数组存放数据，利用指针操作，将神兽的模样正确显示在屏幕上。应在main函数调用showFile函数。

**任务3：驯兽高手**

费尽周折将神兽解救出来之后，刚从长期封印中苏醒过来的神兽似乎迷失了方向，昏头昏脑地往背离救世主的远方走去。小明哪里舍得前功尽弃，于是口里念念有词，手上频频点指，终于使得神兽调头归来。你知道小明是怎么做到的吗？

*任务3的要求*：

实现函数void mirrorTrans(char \* arr, int h, int w)，操作存储神兽的数组，使之水平旋转180度（镜像大法），并把结果显示在屏幕上，其中arr的实参应为二维数组首元素的地址，h为高，w为宽。应在main函数调用mirrorTrans函数。注意：旋转后神兽的形态不应有明显的变化！

**任务4：枪林弹雨**

随着神兽的封印解开，与神兽同时被封印在文本文件devil.txt的邪恶势力也破茧而出，随即对神兽展开了猛烈的攻击。各种攻击覆盖了神兽所在的22×46的范围。救世主空有百试百灵的治疗术，但是呆头呆脑的神兽却从来不会喊疼，所以救世主需要一个辅助能力，对22×46范围内的任何一个坐标，能马上判断出神兽是否被击中。注意，擦破皮（正好在边缘）也算被击中。你能帮助救世主得到这个能力吗？

已知神兽不会贴近22×46这个范围的四条边沿（即首行、末行、首列、末列都属于不会被击中的位置，显示神兽后也能观察到，这点也许有利于你开发该能力）。

为了证明你正确地开发了该能力，请画出效果图，即把范围内所有不会被击中的位置标注为’\*’号，而神兽原本占据的位置显示为空格。

*任务4的要求*：

（1）在main函数调用任务2实现的showFile函数，显示邪恶势力devil的真面目，devil占据空间的大小为16×54；

（2）实现函数void coverHit(char \* arr, int h, int w)画一个示意图，用’\*’号标出22×46范围内所有发生未击中情况的位置点，其中arr为存放神兽的二维数组的首元素的地址，h为高，w为宽。当然，需要在main函数调用coverHit。

**任务5：终极封印**

现在，拯救世界的担子落到了小明的肩上。经过分析，小明发现原来devil.txt的封印方式太逊了，因为文件太冗余了。小明需要掌握一门终极封印术，把devil再次封印。封印术的关键在于：（1）封印文件仍应该能够描述devil的全部信息；（2）封印文件越小，封印效果越好。为此，小明踏上了茫茫的修炼之路，究竟是修炼人族的ASCII码（Easy）？还是矮人族的折叠术（Normal）？还是精灵族的赫夫曼树（Difficult，可查阅数据结构相关介绍）？还是小明自创的独门封印绝技（Amazing!）？决定权在小明手里，让我们拭目以待……

*任务5的要求：*

（1）实现函数void sealDevil(char \* arr, char \* filename) 对devil进行封印，其中arr表示存放devil的二维数组首元素的地址，filename表示新的封印文件的路径名，建议的文件名为“NewSeal”。

（2）实现函数void unsealDevil(char \* filename)读取新的封印文件NewSeal，显示解封的结果（即再次把devil完整显示在屏幕上）以证明你的封印过程无误。