

# 实验 2



## 创建基本回文应用

### 实验结构

- ▶▶ 在 Objective-C 开发环境中创建一个基本的回文程序

### 实验目标

本实验结束后，你将能够：

- ▶▶ 搭建 Objective-C 开发环境
- ▶▶ 在 Objective-C 开发环境下创建基本的回文程序

## 导论

iOS 应用开发需要 Objective-C 的知识。本实验中，你将创建一个 Objective-C 开发环境上运行的应用。你将使用 iOS terminal 来开发、编译和执行程序。

## 实验：创建一个基本的回文应用

Objective-C 是一个编译语言。也就是说，它会被编译为原始机器码，而不是在运行时被解释。顾名思义，它基于 C 编程语言，其语法和概念类似于基于 C 的语言。

## 背景

回文是正反向读起来都一样的文字、短语、数字或其它字符串序列。著名的例子包括 “A man, a plan, a canal, Panama”、“race car”、“taco cat”。本实验中，你将创建一个检查字符串是不是回文的程序。

你需要执行下面这些任务：

1. 安装用于 Objective-C 的集成开发环境（IDE）。为此，你将使用 EditRocket IDE。你也可以使用 iOS terminal。
2. 写程序检查，看字符串是不是回文。

## 实验准备

---

开始本实验之前，你需要确保：

- 安装了 EditRocket
- 具有 C 的基本知识，包括数据类型和格式
- 具有 IDE 的知识，包括如何使用及在哪保存所需数据

## 实验：推荐解决方案

### 任务 1 解决方案：为 Objective-C 搭建环境

---

市面上有很多 IDE，例如 [EditRocket](#)、[Jetbrains](#) 和 [Xcode](#)，可以用于创建、编译和执行 Objective-C 程序。本实验中，你将使用 iOS 命令行终端 terminal 来执行 Objective-C 程序。

1. 在系统上打开 iOS 命令行终端 terminal
2. 键入超级用户 (su)。这将询问一个密码，如图 1 所示：

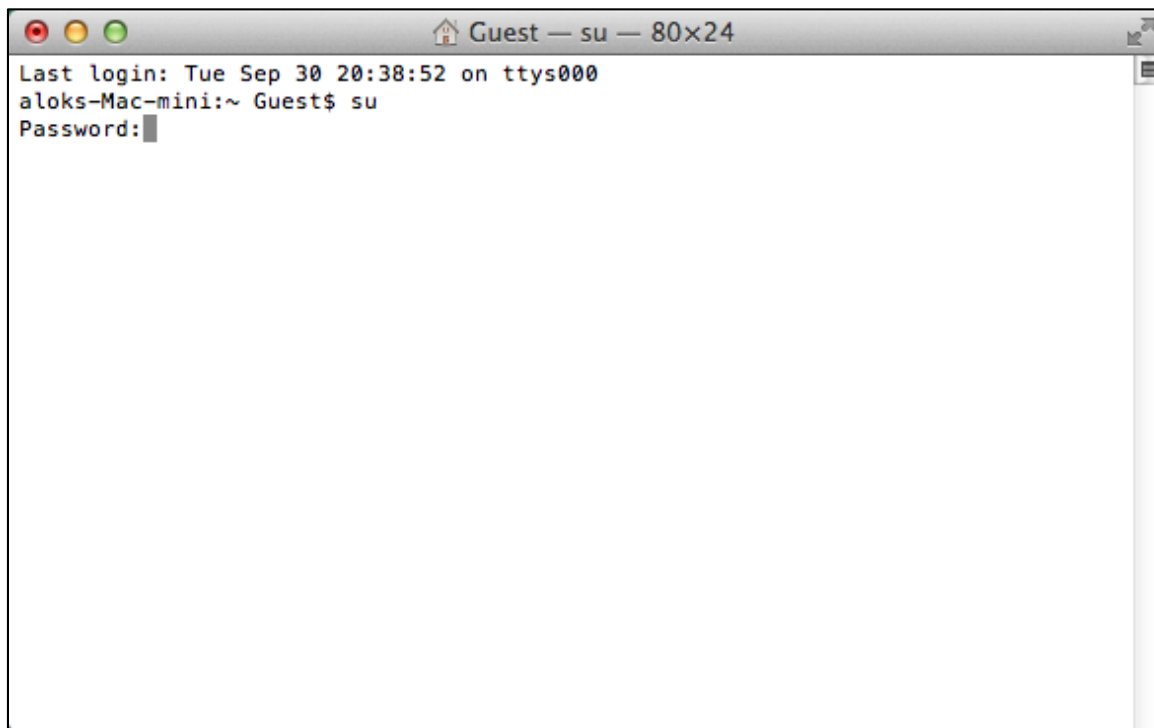


图 1：在 iOS 命令行终端 Terminal 中键入 su 和密码

## 任务 2 解决方案：写程序来检查字符串是不是回文

---

1. 你需要使用 vim 在命令行终端 terminal 写一个程序，如图 2 所示：

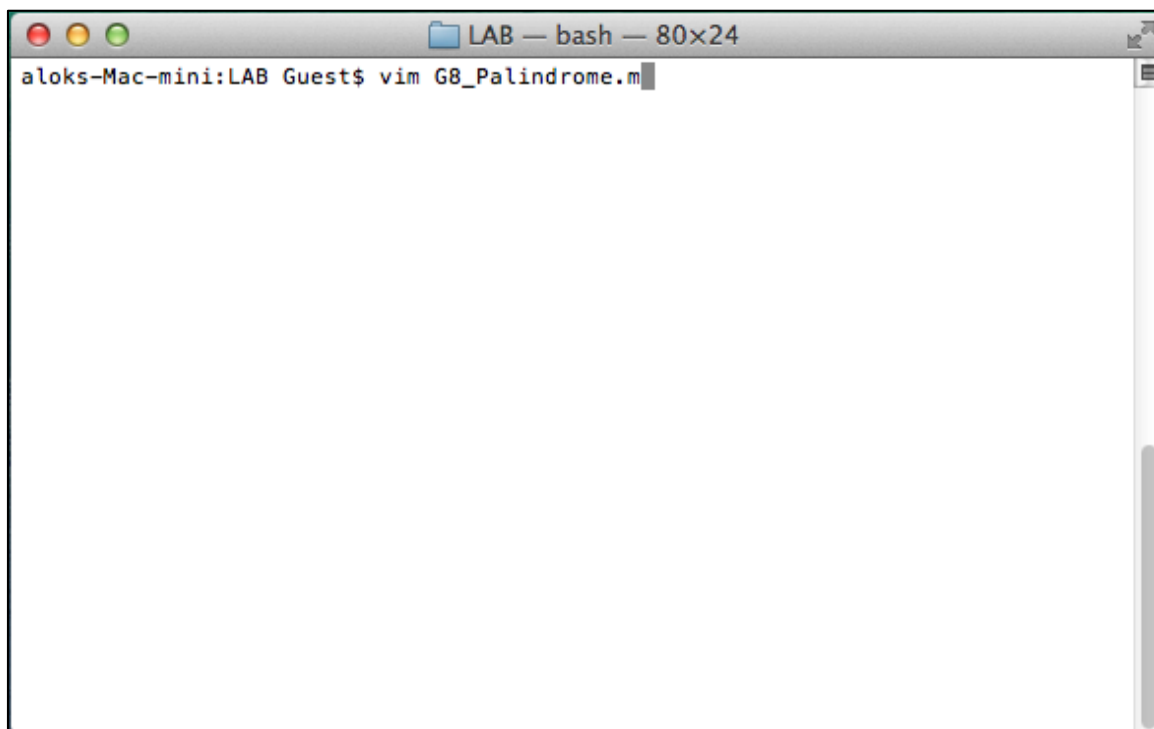


图 2：在命令行终端 Terminal 上使用 Vim

模块：为 iOS 平台开发基本应用

2. Vim 会打开一个空页面，让你来写程序，如图 3 所示：

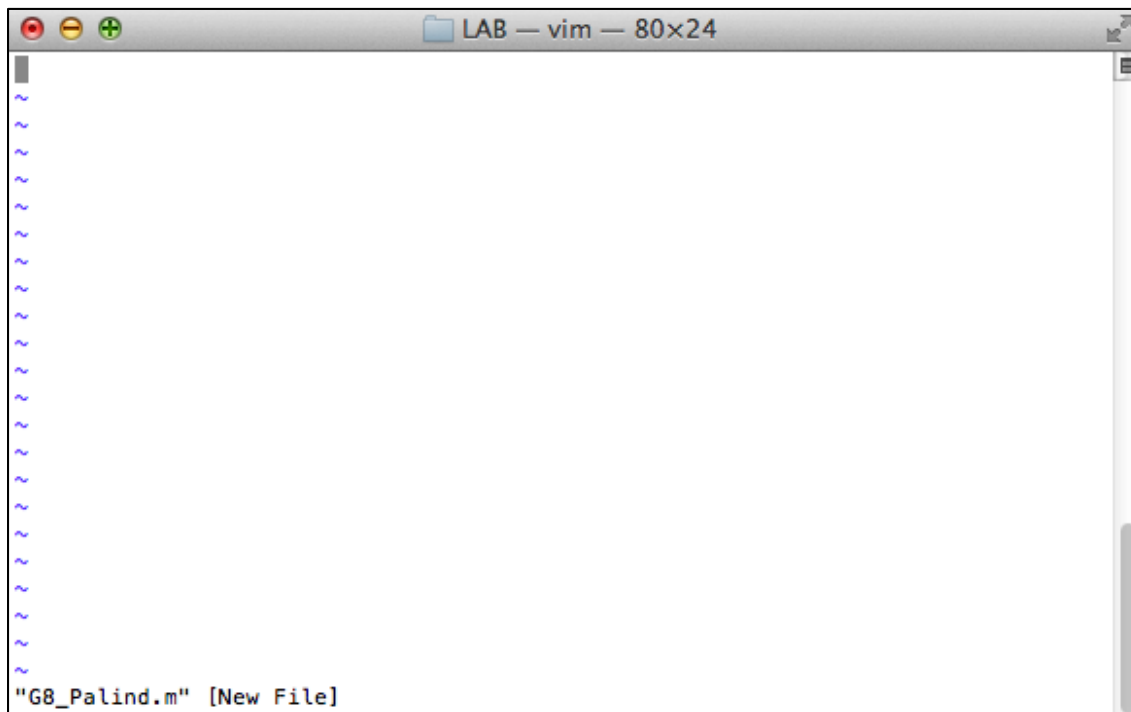


图 3：写程序的空页面

3. 下面写一个程序来测试字符串是不是回文，如图 4 所示。写代码的方式有很多种，源码我们会在本实验最后提供。

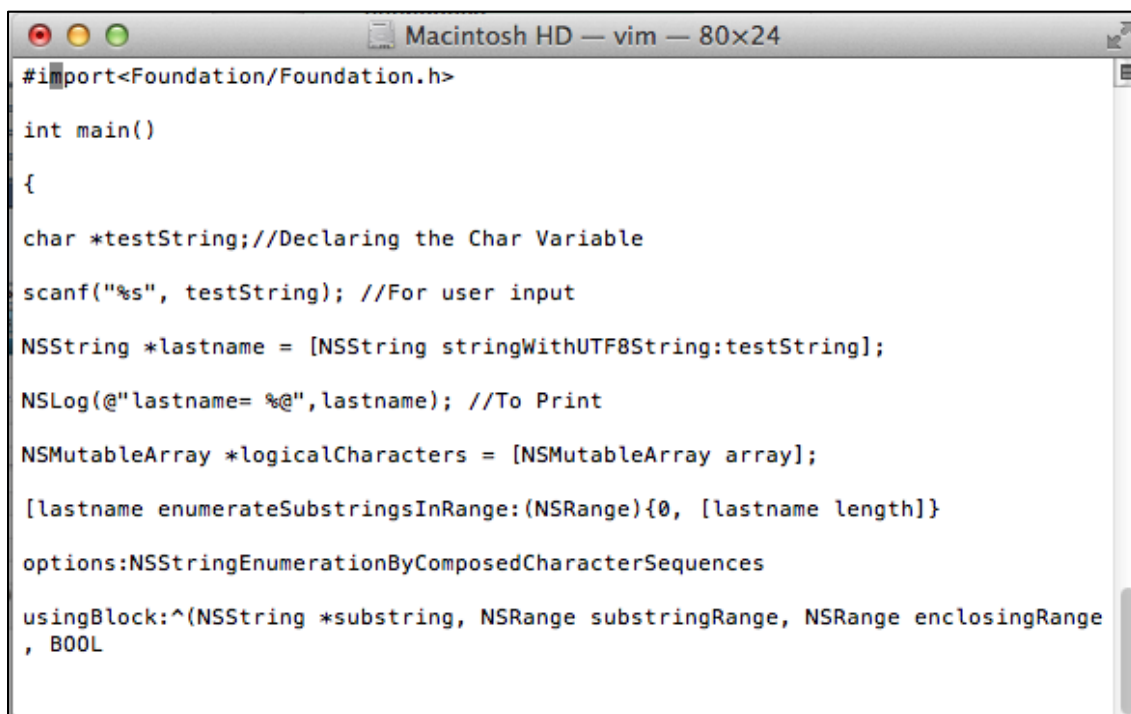


图 4：写程序来测试字符串是不是回文

4. 程序写完之后需要保存（参见图 5），为此你需要：

1. 按 键盘上的<esc>按键
2. 键入 “:wq”并按回车

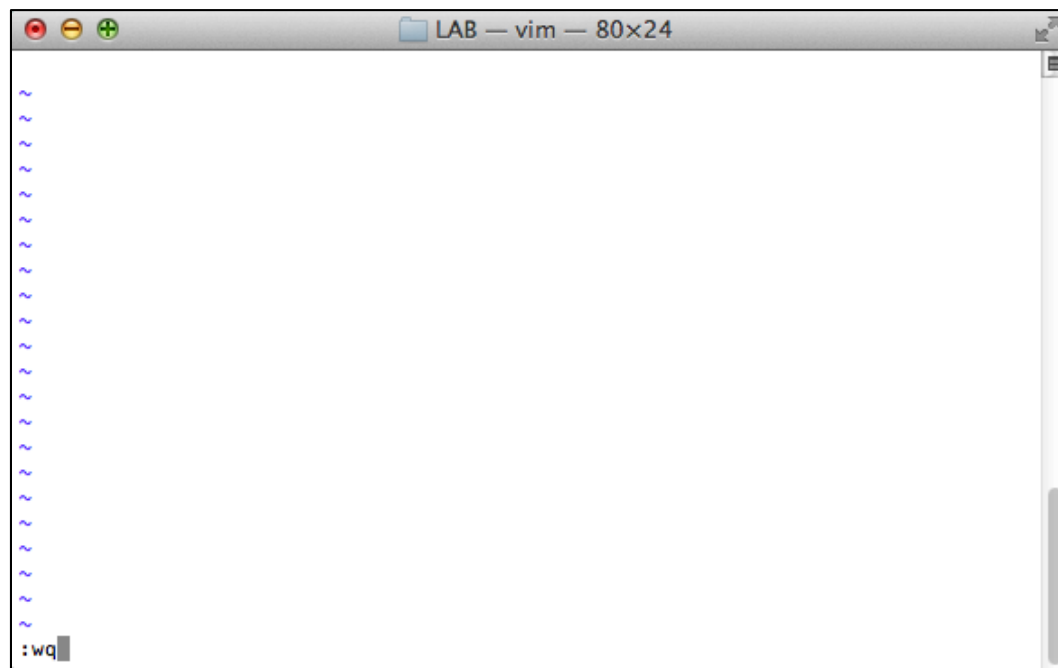


图 5：保存程序

5. 保存程序后，你需要编译程序，做法是在命令行终端 **terminal** 上键入 `gcc -framework Foundation G8_Palindrome.m -o G8_Palindrome`，如图 6 所示：

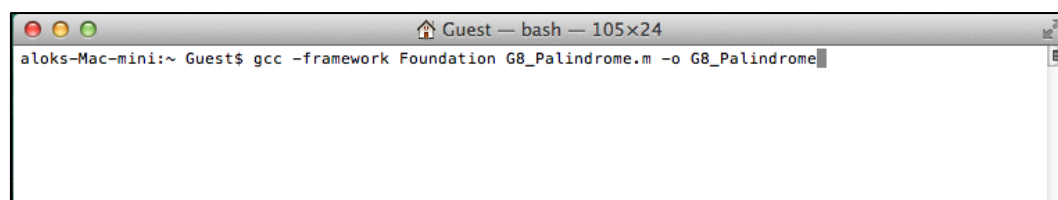


图 6：编译程序

6. 程序编译后需要执行（参见图 7）。

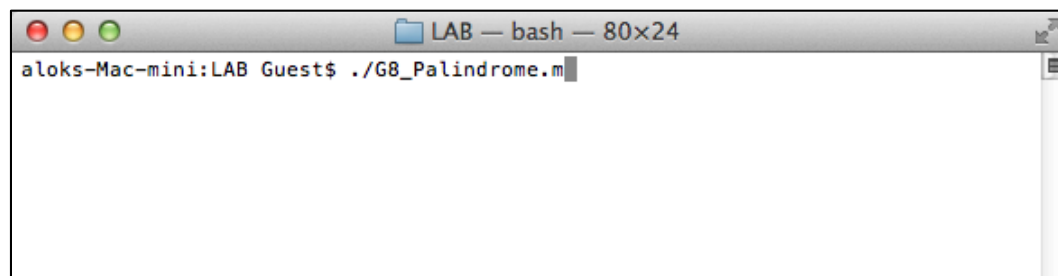


图 7：执行程序

## 源码

解决方案的源码如下所示，自己写不出来的学员可以参考。这对于实验而言非常有用，我们的目标是创建一个检查字符串是不是回文的应用。

```
#import<Foundation/Foundation.h>
int main()
{

    char *testString;//Declaring the Char Variable

    scanf("%s", testString); //For user input

    NSString *lastname = [NSString stringWithUTF8String:testString];
    NSLog(@"lastname= %@",lastname); //To Print
    NSMutableArray *logicalCharacters = [NSMutableArray array];
    [lastnameenumerateSubstringsInRange:(NSRange){0, [lastname length]}
    options:NSStringEnumerationByComposedCharacterSequences
    usingBlock:^(NSString *substring, NSRangesubstringRange, NSRangeenclosingRange, BOOL
    *stop)
    {
        [logicalCharactersaddObject:substring];
    }
    ];
    NSUInteger count = [logicalCharacters count];
    BOOL isPalindrome = YES;
    for (NSUIntegeridx = 0; idx< count / 2; ++idx) {
        NSString *a = logicalCharacters[idx];
        NSString *b = logicalCharacters[count - idx - 1];
        if ([a localizedCaseInsensitiveCompare:b] != NSOrderedSame) {
            isPalindrome = NO;
            break;
        }
    }
    NSLog(@"isPalindrome: %d", isPalindrome);
    return 0;
}
```