

实验 3



Xcode 开发环境

实验结构

- ▶▶ 在 Xcode 集成开发环境（IDE）中开发基本程序

实验目标

本实验结束后，你将能够：

- ▶▶ 下载和安装 Xcode IDE
- ▶▶ 在 Xcode 中创建回文应用

模块：为 iOS 平台开发基本应用

导论

Xcode IDE 被用于开发 iOS 应用和 Mac OS X 桌面应用。它很类似于微软 Visual Studio 和 Eclipse。你可以首先用它来创建项目，然后再在其中编辑代码和用户界面文件。

实验：在 XCODE 中使用开发环境

Xcode 是将代码（也就是文本指令）转化为能在 iOS 设备上运行的应用的工具。Xcode 可以从下列网址下载：

developer.apple.com/xcode/downloads.

背景

应用是在多种数据的基础上建立起来的。除了处理器上运行的原始二进制指令之外，应用还包括声音文件、图形文件、屏幕、背景、按钮、游戏令牌、字体文件和其它支持的信息。创建应用时，Xcode 会将这些元素合并到一个产品文件中。大部分构建过程是自动的。

本实验中，你将在 Xcode 中创建一个程序，检查字符串是不是回文。你需要执行下面这些任务：

1. 下载 Xcode IDE
2. 安装 Xcode IDE
3. 在 Xcode 中创建应用，检查字符串是不是回文

实验准备

开始实验之前，你需要有：

- 安装有 Xcode 的 Mac OS
- 苹果开发者帐户

实验：推荐解决方案

任务 1 解决方案：下载 Xcode

1. 打开网页浏览器，到如下链接：developer.apple.com/xcode/downloads。显示的网页如图 1 所示：



图 1: 下载 Xcode 的网页

2. 在网页上, 点击链接来下载最新版的 Xcode, 如图 2 所示:



图 2: 选择下载最新版的 Xcode

3. 在 **Register as an Apple Developer** 页面 (参见图 3), 填入你的应用开发角色和经历。

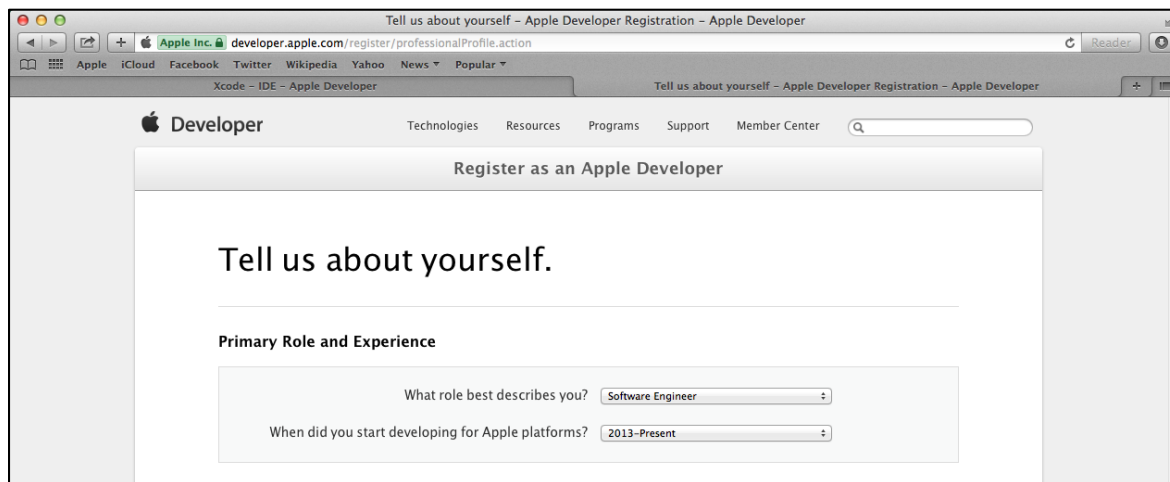
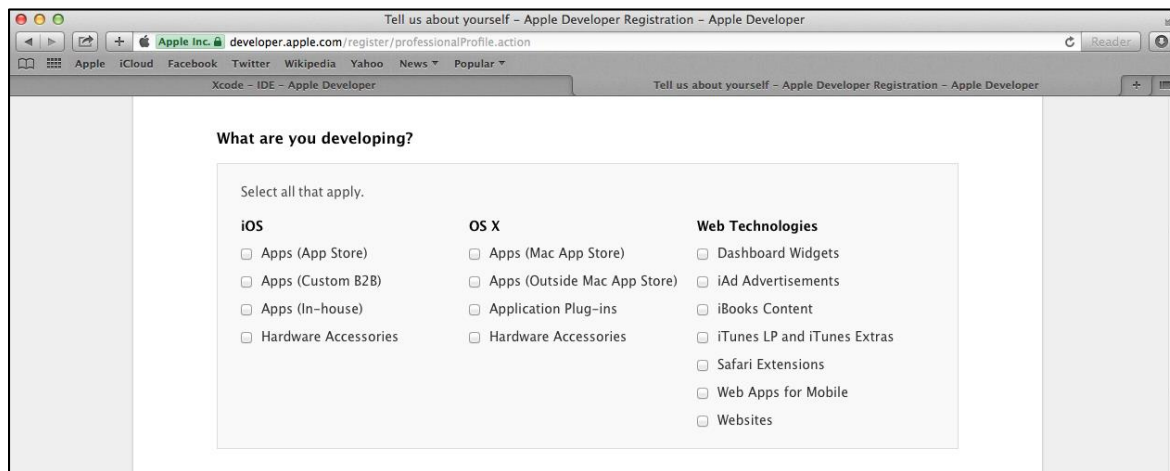


图 3: 填入个人细节, 下载 Xcode

4. 向下滚动并输入开发相关的操作系统细节信息 (参见图 4)。



模块：为 iOS 平台开发基本应用

图 4: 填入用于应用开发的细节信息

5. 向下滚动，选择你在其它平台上的工作经历（参见图 5）。

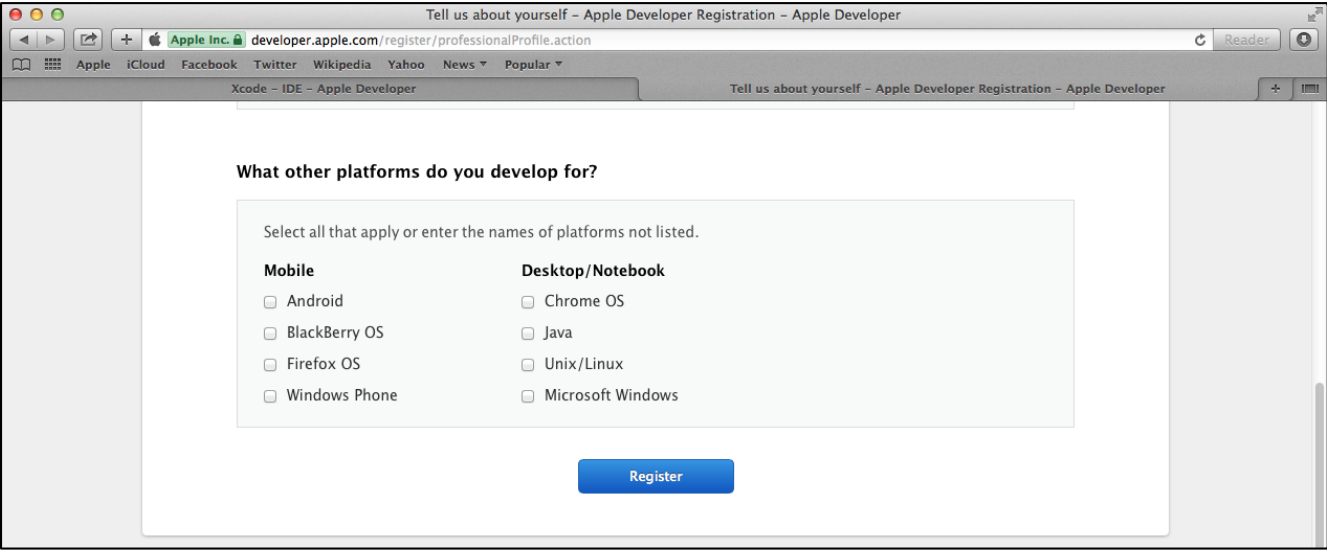


图 5: 选择你在其它平台上的工作经历

6. 继续向下滚动。阅读条款，确认 Registered Apple Developer Agreement（参见图 6）。

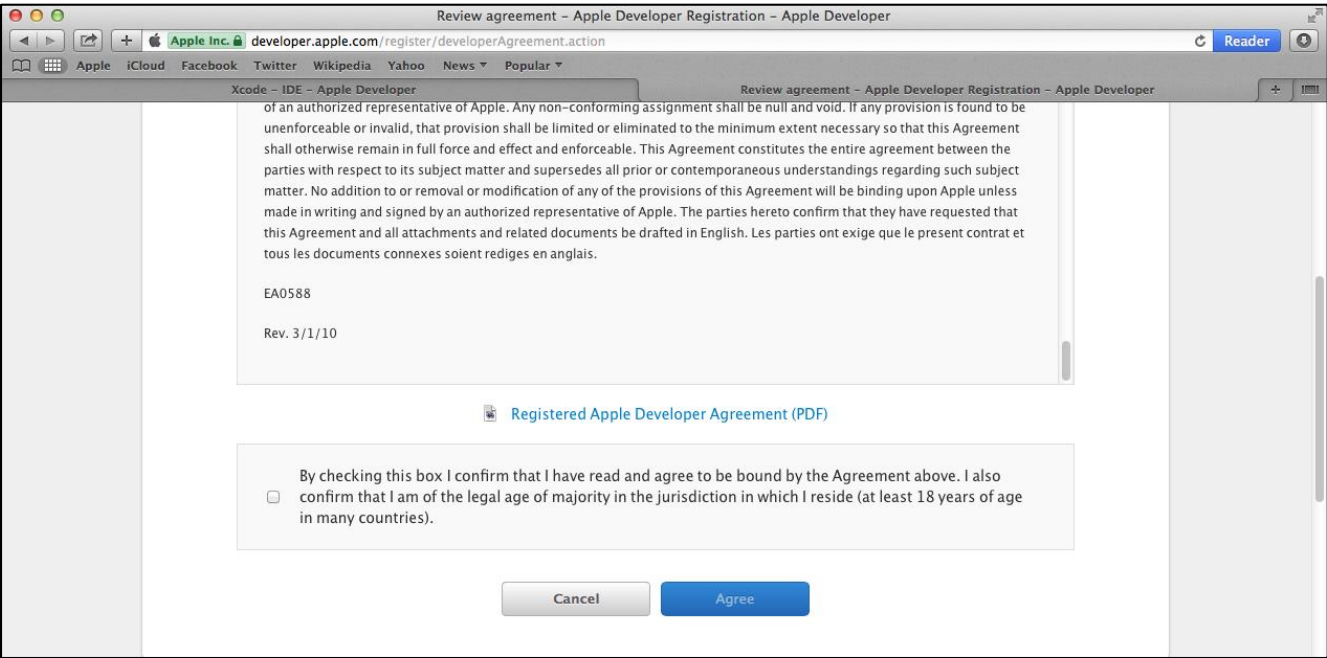


图 6: 确认 Registered Apple Developer Agreement

任务 2 解决方案: 搭建 Xcode 开发环境

1. 使用苹果开发者帐户登陆, 选择 **Apple Developer Forums** 选项, 如图 7 所示:

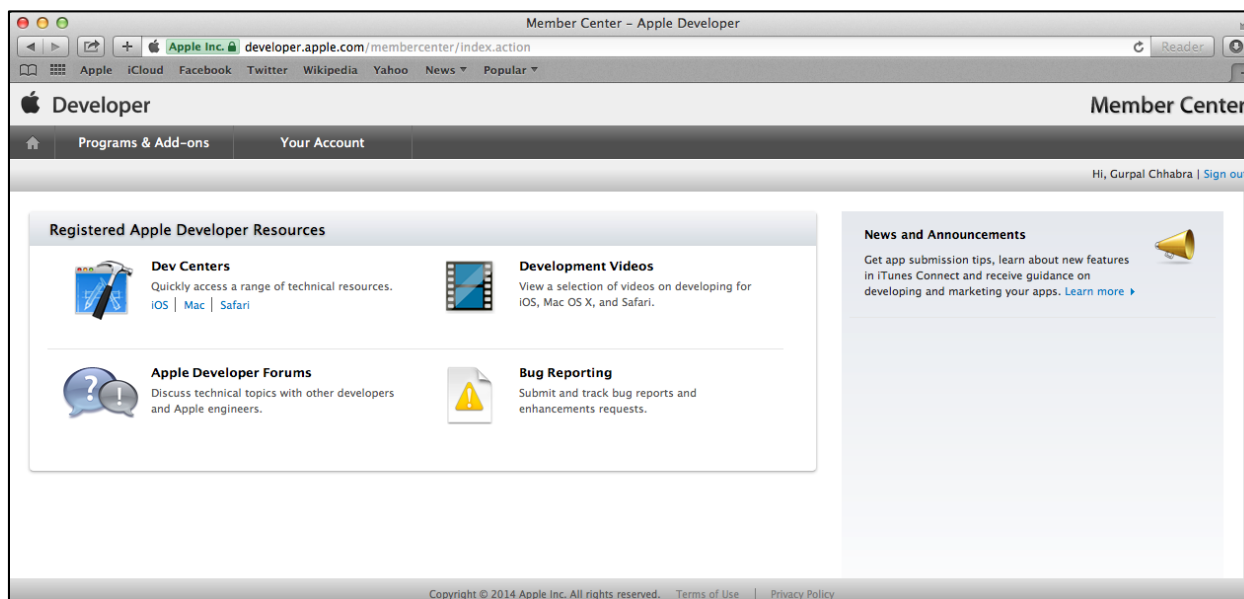


图 7: 选择 Apple Developer Forums

2. 要搭建 Xcode 开发环境, 将 **Xcode** 图标拖到 **Applications** 文件夹, 参见图 8:



图 8: 将 Xcode 图标拖到 Applications 文件夹

任务 3 解决方案：在 Xcode 上创建回文应用

1. Xcode 安装后，准备好创建应用。在 **Welcome to Xcode** 界面，选择 **Create a new Xcode project** 选项开始新项目，参见图 9：



图 9: 从欢迎页面开始创建新 Xcode 项目

2. 这会显示 **Choose a template for your new project** 页面。选择面板左侧 **OS X** 部分的 **Application** 选项，然后选择 **Command Line Tool** 模板，如图 10 所示。点 **Next** 继续下一步。

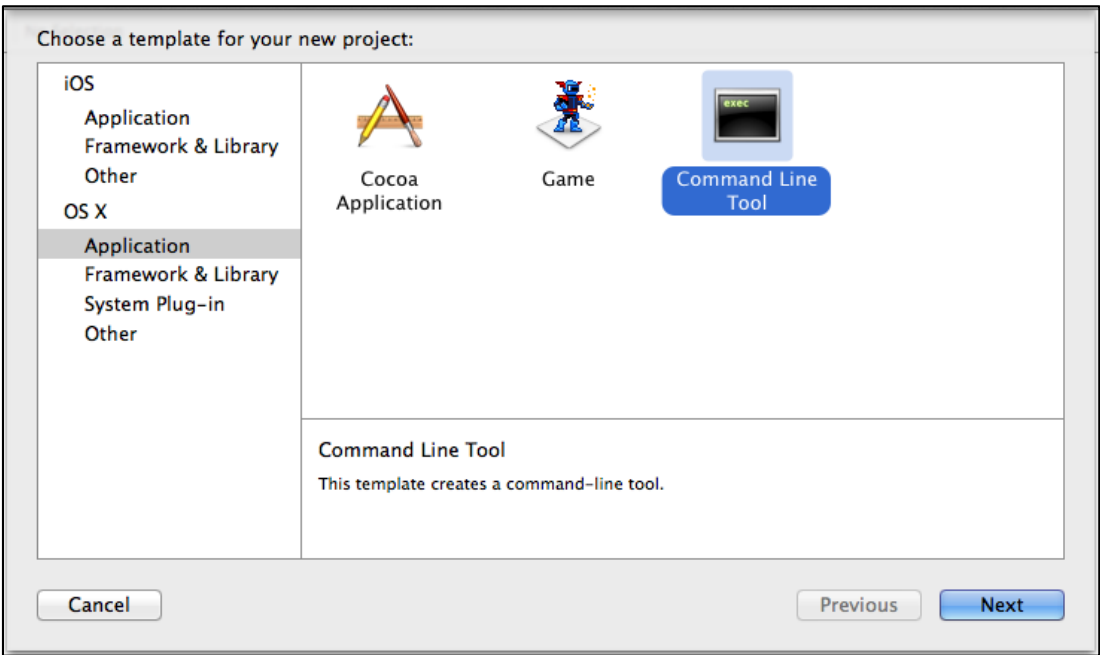


图 10: 选择 Command Line Tool 模板来运行 Objective-C 程序

3. 在 **Choose options for your new project** 界面（参见图 11），命名项目为 **Palindrome** 并点 **Next**。

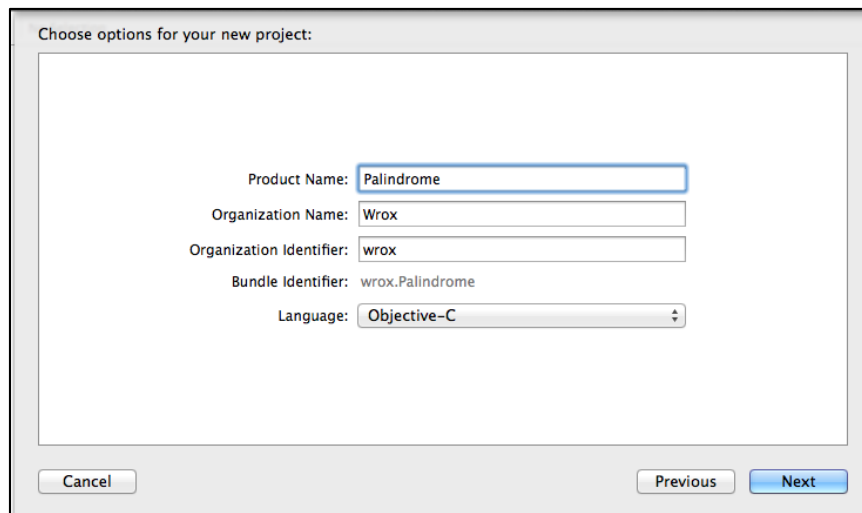


图 11: 命名新项目

4. 下面开始在 `main.m` 文件中创建和执行回文程序，如图 12 所示。相关源码提供在本实验最后。

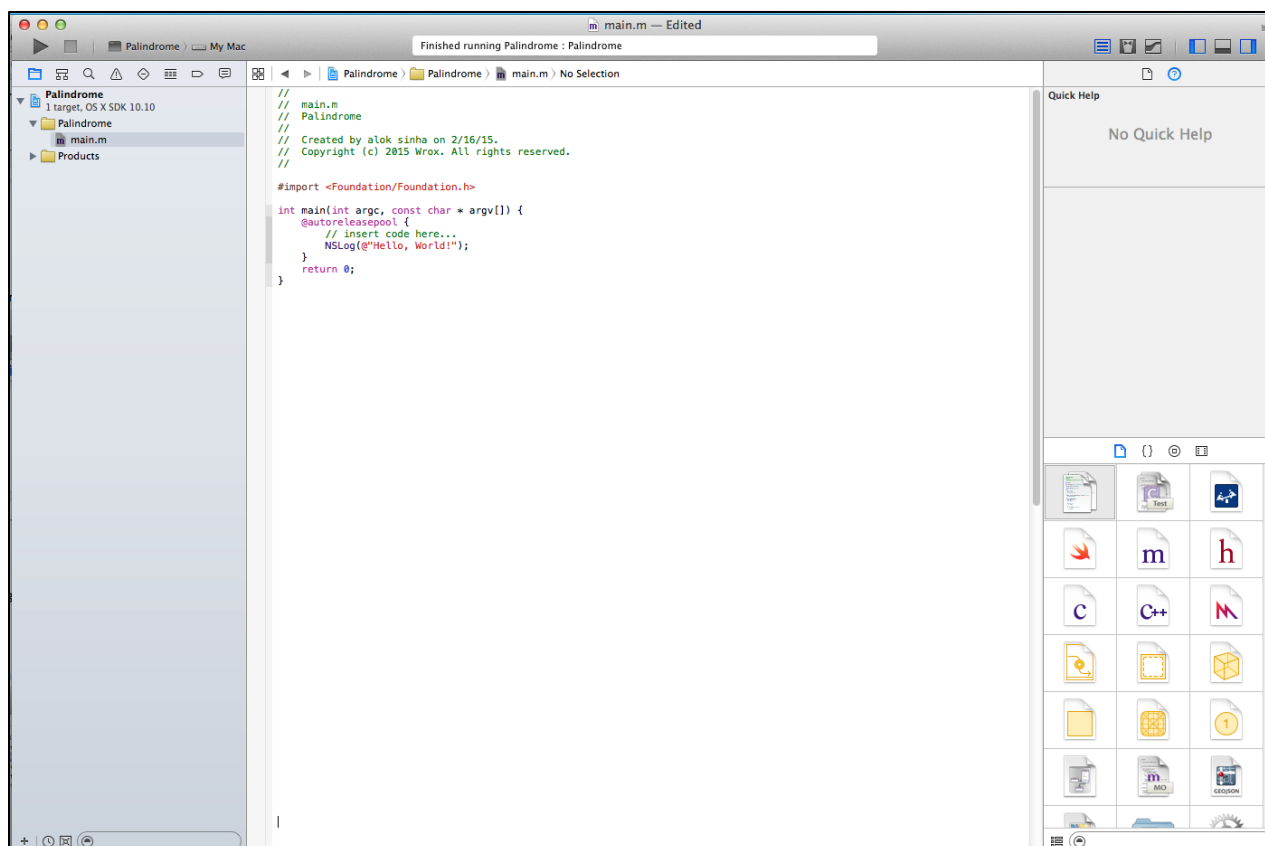


图 12: Main.m 文件

5. 要编译和运行程序，点屏幕左上角的 **Run** 按钮。图 13 中可以看到，这个按钮位于左上角红黄圆圈的正下方。

模块：为 iOS 平台开发基本应用

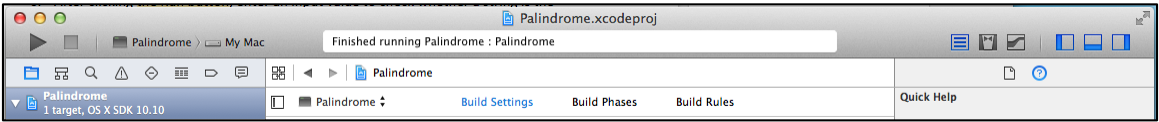


图 13: 点击按钮来编译和运行程序

6. 点 **Run** 按钮后，输入一个字符串，看它是否是回文。图 14 显示了使用“hello world”字符串的默认 main.m。

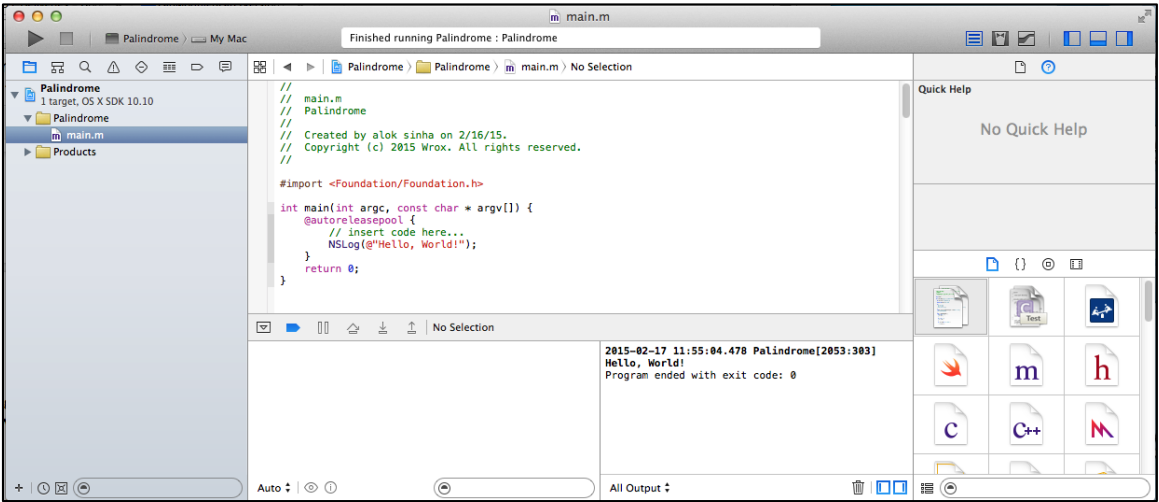


图 14: 使用“hello world”字符串的默认 main.m

7. 图 15 显示了输入另一个字符串后的输出：

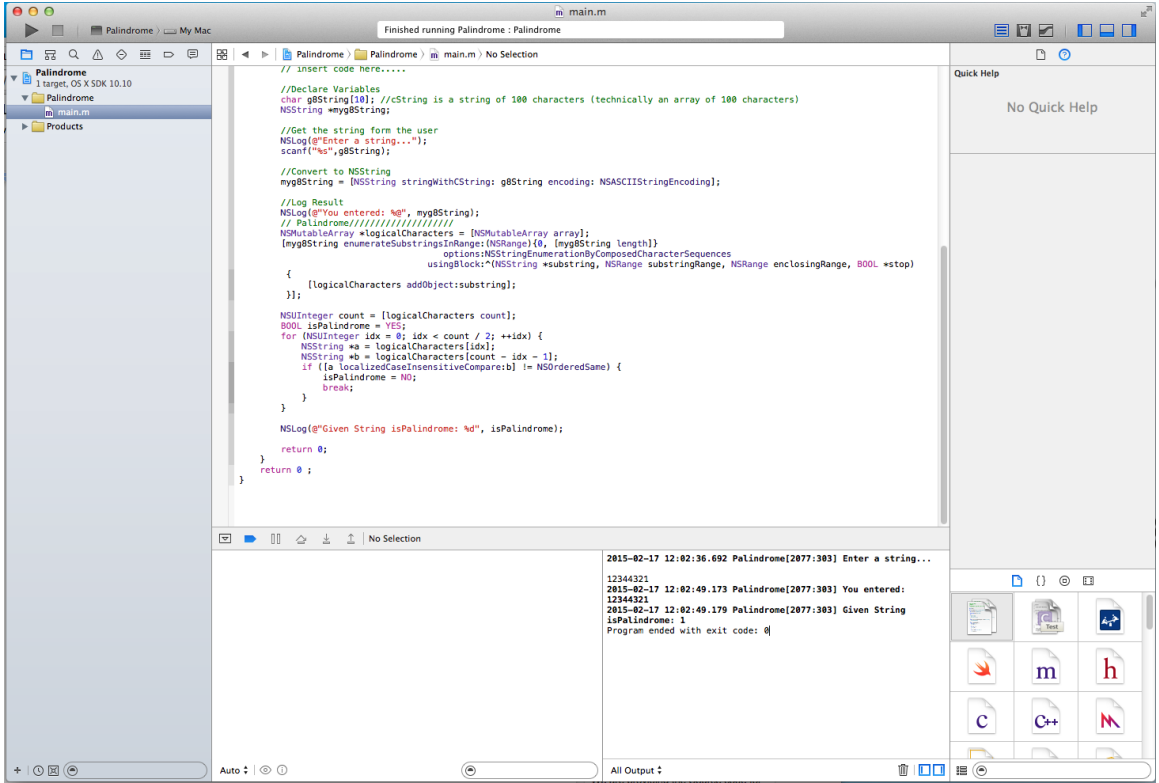


图 15: 用户定义字符串后的输出

源码

```

// main.m
// palindrome
//
// Created by Guest User on 10/7/14.
// Copyright (c) 2014 GlobusEight. All rights reserved.
//

#import <Foundation/Foundation.h>

int main()
{

    @autoreleasepool {

        // insert code here.....

        //Declare Variables
        char g8String[100]; //cString is a string of 100 characters (technically an array of 100
characters)
        NSString *myg8String;

        //Get the string form the user
        NSLog(@"Enter a string...");
        scanf("%s",g8String);

        //Convert to NSString
        myg8String = [NSString stringWithCString: g8String encoding:
NSASCIIStringEncoding];

        //Log Result
        NSLog(@"You entered: %@", myg8String);
        // Palindrome/////////////////////////////////
        NSMutableArray *logicalCharacters = [NSMutableArray array];
        [myg8String enumerateSubstringsInRange:NSMakeRange(0, [myg8String 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) {

```

模块：为 iOS 平台开发基本应用

```
        isPalindrome = NO;
        break;
    }
}

NSLog(@"Given String isPalindrome: %d", isPalindrome);

return 0;

}
return 0;
}
```