**汕 头 大 学**

**《程序设计基础》综合实践报告**

**学院： 数学与计算机学院 系：计算机系 专业: 数据科学与大数据技术 年级： 2024**

**课程： 程序设计基础 任课教师：** **郑麟**  **学号：** 2024612001

**姓名：** 彭俊宏

**一、简介**

本程序是一个简单的万年历应用，具备显示指定年份和月份的日历、添加事件以及查询特定日期事件的功能。旨在为用户提供方便的日历查看与事件管理工具，帮助用户记录和查看重要事件。通过简洁的交互界面，用户可以轻松操作，提高时间管理的效率，解决日常生活中对日期查看和事件记录的需求。

**二、功能**

**1.显示日历**

* 用户在主菜单选择 “显示日历”，然后输入年份和月份，程序将打印出对应年份和月份的日历，展示每月的日期分布以及星期信息，如图所示。

图片包含 文本

描述已自动生成

**2.添加事件**

* 选择 “添加事件” 后，按提示依次输入事件的年份、月份、日期和事件描述，成功添加后会有提示信息，如图所示。

图片包含 文本

描述已自动生成

**3.查询事件**

* 选择 “查询事件”，输入要查询的年份、月份和日期，若存在该日期的事件，则显示事件描述，否则提示未找到，如图所示。

图片包含 文本

描述已自动生成

图片包含 文本

描述已自动生成

**三、设计过程**

1. **数据结构设计**
   * 定义Event结构体来存储事件信息，包括年份、月份、日期和事件描述，方便对事件进行统一管理。使用events数组存储所有事件，event\_count记录当前事件数量。
2. **功能函数实现**
   * **判断闰年**：is\_leap\_year函数通过判断年份是否满足闰年条件（能被 4 整除但不能被 100 整除，或者能被 400 整除）来确定是否为闰年，为计算 2 月天数提供依据。

图片包含 日程表

描述已自动生成

* + **获取每月天数**：get\_days\_in\_month函数根据月份和是否为闰年计算该月的天数，通过数组预先定义每月天数，并在 2 月时根据闰年情况调整。

图片包含 文本

描述已自动生成

* + **计算每月第一天星期几**：getFirstDayOfWeek函数采用基姆拉尔森计算公式计算每月第一天是星期几，为日历排版提供起始位置。同时需要注意的是闰年的二月份是29天，在最后需要用一个if-else区分，因为该公式默认二月份为28天。

文本

描述已自动生成

* + **打印日历**：print\_calendar函数先获取每月天数和第一天星期几，然后按格式打印日历，通过循环输出空格保证日期与星期对应整齐。

文本

描述已自动生成

* + **添加事件**：add\_event函数检查事件列表是否已满，若未满则接收用户输入的事件信息，存储到events数组并更新event\_count，同时注意缓冲区处理以确保正确读取事件描述。

文本

描述已自动生成

* + **查询事件**：query\_event函数根据用户输入的日期，在events数组中遍历查找匹配事件，若找到则输出事件描述，否则提示未找到。

图形用户界面, 文本, 应用程序

描述已自动生成

1. **主菜单设计**：menu函数通过循环显示主菜单，根据用户选择调用相应功能函数，实现程序的交互操作，直到用户选择退出。

**四、小结**

1. **功能特性总结**
   * 本程序实现了万年历的基本功能，包括准确显示日历、方便地添加和查询事件，界面简洁，操作流程清晰，基本满足用户日常对日期和事件管理的需求。
2. **实践感想**
   * 通过本次实践，我深入理解了程序设计的流程，从需求分析到功能设计与实现，每个环节都紧密相连。在实现过程中，遇到了如日期计算、缓冲区处理等问题，但通过查阅资料和不断调试得以解决，提升了自己的问题解决能力。同时也认识到程序设计中细节的重要性，如输入格式验证、界面友好性等，这些都影响着用户体验。未来，我将进一步优化程序，增加更多实用功能，提高程序的稳定性和扩展性。

**五、附录（源代码）**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_EVENTS 100 // 最大事件数量

typedef struct {

int year;

int month;

int day;

char description[100];

} Event;

Event events[MAX\_EVENTS];

int event\_count = 0;

// 判断是否为闰年

int is\_leap\_year(int year) {

return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

}

// 获取某月的天数

int get\_days\_in\_month(int year, int month) {

int days\_in\_month[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if (month == 2 && is\_leap\_year(year)) {

return 29;

}

return days\_in\_month[month - 1];

}

//计算每个月的第一天是星期几，打印该月的日历

int getFirstDayOfWeek(int year, int month) {

if (month == 1 || month == 2) {

month += 12;

year--;

}

int c = year / 100;

int y = year % 100;

int d = 1; // 第一天

int w = (d + 2\*month + 3\*(month+1)/5 + y + y/4 - y/100 + y/400 + 1) % 7;

return w;

}

//打印某年的某月的日历

void print\_calendar(int year, int month) {

int days = get\_days\_in\_month(year, month);

int start\_day =0;

if (month == 2 && is\_leap\_year(year)) {

start\_day = getFirstDayOfWeek(year,month)+1;

}

else

{

start\_day = getFirstDayOfWeek(year,month);

}

printf(" %d年%d月\n", year, month);

printf("日 一 二 三 四 五 六\n");

for (int i = 0; i < start\_day; i++) {

printf(" ");

}

for (int day = 1; day <= days; day++) {

printf("%2d ", day);

if ((start\_day + day) % 7 == 0) {

printf("\n");

}

}

printf("\n");

}

// 添加事件

void add\_event() {

if (event\_count >= MAX\_EVENTS) {

printf("事件列表已满，无法添加更多事件。\n");

return;

}

Event e;

printf("请输入事件的年份：");

scanf("%d", &e.year);

printf("请输入事件的月份：");

scanf("%d", &e.month);

printf("请输入事件的日期：");

scanf("%d", &e.day);

printf("请输入事件描述：");

getchar(); // 清除缓冲区

fgets(e.description, 100, stdin);

e.description[strcspn(e.description, "\n")] = '\0'; // 移除换行符

events[event\_count++] = e;

printf("事件已添加成功！\n");

}

// 查询事件

void query\_event() {

int year, month, day;

printf("请输入查询的年份：");

scanf("%d", &year);

printf("请输入查询的月份：");

scanf("%d", &month);

printf("请输入查询的日期：");

scanf("%d", &day);

int found = 0;

for (int i = 0; i < event\_count; i++) {

if (events[i].year == year && events[i].month == month && events[i].day == day) {

printf("事件：%s\n", events[i].description);

found = 1;

}

}

if (!found) {

printf("未找到任何事件。\n");

}

}

// 主菜单

void menu() {

int choice;

while (1) {

printf("\n===== 万年历 =====\n");

printf("1. 显示日历\n");

printf("2. 添加事件\n");

printf("3. 查询事件\n");

printf("4. 退出\n");

printf("请输入你的选择：");

scanf("%d", &choice);

switch (choice) {

case 1: {

int year, month;

printf("请输入年份：");

scanf("%d", &year);

printf("请输入月份：");

scanf("%d", &month);

print\_calendar(year, month);

break;

}

case 2:

add\_event();

break;

case 3:

query\_event();

break;

case 4:

printf("再见！\n");

exit(0);

default:

printf("无效选择，请重新输入。\n");

}

}

}

int main() {

menu();

return 0;

}