CALENDER

**#include <stdio.h>**

**#include <stdlib.h>**

**int isLeapYear( int year );**

**int leapYears( int year );**

**int todayOf( int y, int m, int d);**

**long days( int y, int m, int d);**

**void calendar(int y, int m);**

**int getDayNumber(int d,int m,int y);**

**char \*getName(int day);**

**void flush()**

**{**

**int c;**

**while ((c = getchar()) != '\n' && c != EOF);**

**}**

**typedef struct {**

**int day;**

**int month;**

**int year;**

**char note[255];**

**} Note;**

**int main(int argc, char\* argv[]){**

**int year,month, day;**

**char choice;**

**Note note;**

**FILE \*fp;**

**fp = fopen("note.bin", "r");**

**if (fp == NULL) {**

**fp = fopen("note.bin", "w");**

**}**

**fclose(fp);**

**while(1) {**

**printf("1. Find the day\n");**

**printf("2. Print calendar of a month\n");**

**printf("3. Add Note\n");**

**printf("4. Exit\n");**

**printf("Enter your choice: ");**

**scanf("\n%c", &choice);**

**switch(choice) {**

**case '1':**

**printf("Enter the day, month and year: ");**

**scanf("%d %d %d", &day, &month, &year);**

**printf("The day is : %s\n", getName(getDayNumber(day, month, year)));**

**break;**

**case '2':**

**printf("Enter the month and year: ");**

**scanf("%d %d", &month, &year);**

**printf("Please enter 's' to see the notes\n Press any other key to continue\n");**

**calendar(year, month);**

**break;**

**case '3':**

**printf("Enter the day, month and year: ");**

**scanf("%d %d %d", &note.day, &note.month, &note.year);**

**flush();**

**printf("Enter the note: ");**

**fgets(note.note, 255, stdin);**

**fp = fopen("note.bin", "a+");**

**if (fp == NULL) {**

**printf("File note.bin can not be opened\n");**

**exit(1);**

**}**

**fwrite(&note, sizeof(Note), 1, fp);**

**printf("Note added sucessfully\n");**

**fclose(fp);**

**break;**

**case '4':**

**printf("Bye!!");**

**exit(0);**

**break;**

**default:**

**printf("Not a valid option\n");**

**break;**

**}**

**}**

**return 0;**

**}**

**int isLeapYear( int y ){**

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

**}**

**int leapYears( int y ){**

**return y/4 - y/100 + y/400;**

**}**

**int todayOf( int y, int m, int d) {**

**static int DayOfMonth[] =**

**{ -1,0,31,59,90,120,151,181,212,243,273,304,334};**

**return DayOfMonth[m] + d + ((m>2 && isLeapYear(y))? 1 : 0);**

**}**

**long days( int y, int m, int d){**

**int lastYear;**

**lastYear = y - 1;**

**return 365L \* lastYear + leapYears(lastYear) + todayOf(y,m,d);**

**}**

**void calendar(int y, int m){**

**FILE \*fp;**

**Note\* notes, note;**

**int len, j, hasNote = 0;**

**char choice;**

**const char \*NameOfMonth[] = { NULL/\*dummp\*/,**

**"January", "February", "March", "April", "May", "June",**

**"July", "August", "September", "October", "November", "December"**

**};**

**char Week[] = "Su Mo Tu We Th Fr Sa";**

**int DayOfMonth[] =**

**{ -1,31,28,31,30,31,30,31,31,30,31,30,31 };**

**int weekOfTopDay;**

**int i,day;**

**weekOfTopDay = days(y, m, 1) % 7;**

**fp = fopen("note.bin", "rb");**

**if (fp == NULL) {**

**printf("Couldn't read notes\n");**

**}**

**len = 0;**

**while(fread(&note, sizeof(Note), 1, fp)) {**

**if (note.year == y && note.month == m) {**

**len++;**

**}**

**}**

**rewind(fp);**

**j = 0;**

**notes = (Note\*) malloc (sizeof(Note) \* len);**

**while(fread(&note, sizeof(Note), 1, fp)) {**

**if (note.year == y && note.month == m) {**

**notes[j] = note;**

**j++;**

**}**

**}**

**fclose(fp);**

**if(isLeapYear(y))**

**DayOfMonth[2] = 29;**

**printf("\n %s %d\n%s\n", NameOfMonth[m], y, Week);**

**for(i=0;i<weekOfTopDay;i++)**

**printf(" ");**

**for(i=weekOfTopDay,day=1;day <= DayOfMonth[m];i++,day++){**

**hasNote = 0;**

**for (j = 0; j < len; j++) {**

**if (notes[j].day == day) {**

**printf("|%2d| ",day);**

**hasNote = 1;**

**break;**

**}**

**}**

**if (hasNote == 0) {**

**printf("%2d ",day);**

**}**

**if(i % 7 == 6)**

**printf("\n");**

**}**

**printf("\n");**

**scanf("\n%c", &choice);**

**if (choice == 's') {**

**printf("Here are list of notes for %d %d\n", m, y);**

**for (j = 0; j < len; j++) {**

**printf("%d: %s\n", notes[j].day, notes[j].note);**

**}**

**} else {**

**return;**

**}**

**}**

**int getDayNumber(int d, int m, int y){ static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};**

**y -= m < 3;**

**return (y + y/4 - y/100 + y/400 + t[m-1] + d) % 7;**

**}**

**char \*getName(int day){ switch(day){**

**case 0 :return("Sunday");**

**case 1 :return("Monday");**

**case 2 :return("Tuesday");**

**case 3 :return("Wednesday");**

**case 4 :return("Thursday");**

**case 5 :return("Friday");**

**case 6 :return("Saturday");**

**default:return("Error: Invalid Argument Passed");**

**}**

**}**

**INPUT:**

Enter the date: 29

Enter the month: September

Enter the year: 2020

**OUTPUT:**

Tuesday