

CECS2222 22 Computer Programming II

C. Talavera

Programa que entra la fecha de nacimiento y el programa determina si es una fecha mágica o no. Una fecha mágica es determinada mágica si la multiplicación del mes por el día es igual a los dos últimos dígitos del año.

```
#ifndef _MAGIC_DATE_H
#define _MAGIC_DATE_H
#include<iostream>
using namespace::std;
#include "Date.h";
class MagicDate{
private:
    Date date;
public:
    MagicDate(int month, int day, int year);
    int getYear() const;
    bool getMagicDay() const;
    void display() const;
};
#endif

+++++

#include<iostream>
#include<string>
using namespace::std;
#include "MagicDate.h"

MagicDate::MagicDate(int month, int day, int year) :date(month,day,year){}

int MagicDate::getYear() const{
    //cambio a do digito
    int twoDigitsYear;
    string yyyy;
    yyyy = to_string(date.getYear());
    char yyTemp[3];
    yyTemp[0] = yyyy[2];
    yyTemp[1] = yyyy[3];
    yyTemp[2] = '\0';
    twoDigitsYear = atoi(yyTemp);
    return twoDigitsYear;
}

bool MagicDate::getMagicDay() const{
    if (date.getMonth() *date.getDay() == getYear())
        return true;
    else
        return false;
}

void MagicDate::display() const{
    if (getMagicDay()){
        cout << "The "<<date<<" is a magic date\n";
    }
}
```

```

        else{
            cout << "The "<<date<<" is not magic date\n";
        }
    }

+++++

#ifndef _DATE_H
#define _DATE_H
class Date{
private:
    int month, day, year;
public:
    //Constructores
    Date();
    Date(int aMonth, int aDay, int aYear);
    //Copy Constructor
    Date(const Date &aDate);
    //Destructor
    ~Date();
    //Mutators
    void setMonth(int aMonth);
    void setDay(int aDay);
    void setYear(int aYear);
    //Accessor
    int getMonth() const;
    int getDay() const;
    int getYear() const;
    //Funcion que imprime la fecha
    void displayDate() const;
    void helpIncrement();
    bool endOfMonth(int testDay);
    bool leapYear(int testYear);
    Date &operator++(); // prefix increment operator
    Date operator++(int); // postfix increment operator
    bool operator==(const Date &obj);
    friend ostream &operator << (ostream &strm, const Date &obj);
    friend istream &operator >> (istream &strm, Date &obj);
    static const int days[]; // array of days per month
};
#endif
+++++
#include<iostream>
#include<string>
using namespace::std;
#include "MagicDate.h"

MagicDate::MagicDate(int month, int day, int year) :date(month,day,year){}

int MagicDate::getYear() const{
    //cambio a do digito
    int twoDigitsYear;
    string yyyy;
    yyyy = to_string(date.getYear());
    char yyTemp[3];
    yyTemp[0] = yyyy[2];
    yyTemp[1] = yyyy[3];
    yyTemp[2] = '\0';

```

```

        twoDigitsYear = atoi(yyTemp);
        return twoDigitsYear;
    }

    bool MagicDate::getMagicDay() const{
        if (date.getMonth() *date.getDay() == getYear())
            return true;
        else
            return false;
    }

    void MagicDate::display() const{
        if (getMagicDay()){
            cout << "The "<<date<<" is a magic date\n";
        }
        else{
            cout << "The  "<<date<<" is not magic date\n";
        }
    }
}
+++++

#include<iostream>
using namespace::std;
#include "MagicDate.h"
int main(){

    MagicDate myBirthday1(2,10,1920);
    myBirthday1.display();
    MagicDate myBirthday2(2,10,1921);
    myBirthday2.display();
    system("pause");
    return 0;
} //end main

/*The 2/10/1920 is a magic date
   The 2/10/1921 is not magic date
   Press any key to continue . . .*/

```