

Año
2020

Número
1216

Código de alumno

Práctica

Araoz Andrade, Ivan Alexander

Apellidos y nombres del alumno (letra de imprenta)

Ivan Araoz

Firma del alumno

Curso: TP

Práctica N°: 1

Horario de práctica: P434

Fecha: 12/09/2022

Nombre del profesor: E. Gómez

Nota

19

G. F. V.

Firma del jefe de práctica

Nombre y apellido:
(iniciales)

INDICACIONES

1. Llene todos los datos que se solicitan en la carátula, tanto los personales como los del curso.
2. Utilice las zonas señaladas del cuadernillo para presentar su trabajo en limpio. Queda terminantemente prohibido el uso de hojas sueltas.
3. Presente su trabajo final con la mayor claridad posible. No desglose ninguna hoja de este cuadernillo. Indique de una manera adecuada si desea que no se tome en cuenta alguna parte de su desarrollo.
4. Presente su trabajo final con la mayor pulcritud posible. Esto incluye lo siguiente:
 - cuidar el orden, la redacción, la claridad de expresión, la corrección gramatical, la ortografía y la puntuación en su desarrollo;
 - escribir con letra legible, dejando márgenes y espacios que permitan una lectura fácil;
 - evitar borrones, manchas o roturas;
 - no usar corrector líquido;
 - realizar los dibujos, gráficos o cuadros requeridos con la mayor exactitud y definición posibles.
5. No seguir estas indicaciones influirá negativamente en su calificación.
6. Al recibir esta práctica calificada, tome nota de las sugerencias que se le dan en la contracarátula del cuadernillo.

Zona exclusiva para
cálculos y desarrollos
(borrador)

```
#include <iostream>
#include <fstream>
#include <romanip>
using namespace std;
#include "Funciones.hpp"

int main() {

    ifstream archRegistros ("Precipitaciones.txt", ios::in);
    if (not archRegistros.is_open()) {
        cout << "ERROR: No se pudo abrir el archivo Registros.txt" << endl;
        exit(1);
    }

    ofstream archReporte ("Reporte.txt", ios::out);
    if (not archReporte.is_open()) {
        cout << "ERROR: No se pudo abrir el archivo Reporte.txt" << endl;
        exit(1);
    }

    // llamada a la función principal
    leerImprimirArchivos (archRegistros, archReporte);
    archRegistros.close();
    archReporte.close();
}
```


Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

```
#ifndef FUNCIONES_HPP
#define FUNCIONES_HPP

#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

void leerImprimirArchivos (ifstream &, ofstream &);
void leerImprimirEstacion (ifstream &, ofstream &, int);
void imprimirCabecera (ifstream &, ofstream &, int);
void imprimirCabeceraReporte (ofstream &);
void imprimirReporteResumen (ifstream &, ofstream &);
void imprimirDatosEstacion (ifstream &, ofstream &);
void leerImprimirDato (ifstream &, ofstream &, char);
char cambiarMayus (char);
void leerImprimirReporte (ifstream &, ofstream &);
void imprimirResumen (ofstream &, int, int, int, int, int, int,
double, double, int);
void calcularTiempoLlovido (int, int, int, int, int, int, int &,
int &, int &);
void imprimirReporte (ofstream &, int, int, int, int, int,
int, double);
```

Presente aquí su trabajo

Zona exclusiva para cálculos y desarrollos (borrador)

100-
26
24 12
37

Recuerde que esto es fundamental para el diseño descendente.

Esta línea debe imprimirse solamente una vez.

```
#include <Funciones.hpp>
// el orden de las funciones (modulación) no será perfecto
void leerImprimirArchivos (ifstream &archRegistros,
                           ofstream &archReporte) {
    int numEstacion;
    while (true) {
        archRegistros >> numEstacion;
        if (archRegistros.eof()) break; // llamamos a la función:
        leerImprimirEstacion (archRegistros, archReporte, numEstacion);
    }
}

void leerImprimirEstacion (ifstream &archRegistros, ofstream
                           &archReporte, int numEstacion) {
    // llamamos la función para imprimir hasta Region
    imprimirCabecera (archRegistros, archReporte, numEstacion);
    // llamamos la función que imprime la cabecera del reporte
    imprimirCabeceraReporte (archReporte);
    // llamamos la función que contiene el resto del reporte
    imprimirReporteResumen (archRegistros, archReporte);
}

void imprimirCabecera (ifstream &archRegistros, ofstream
                       &archReporte, int numEstacion) {
    archReporte << setw(37) << " " << "INFORME DE PRECIPITACIONES"
    << endl;
    archReporte << setw(2) << " " << "ESTACION No. " <<
    numEstacion << endl << endl;
    imprimirDatosEstacion (archRegistros, archReporte);
}

void imprimirDatosEstacion (ifstream &archRegistros,
                             ofstream &archReporte) {
    archReporte << setw(2) << " " << "NOMBRE:"
    leerImprimirDato (archRegistros, archReporte, ',');
    archReporte << setw(2) << " " << "CIUDAD:";
    leerImprimirDato (archRegistros, archReporte, ',');
    archReporte << setw(2) << " " << "REGION:";
    leerImprimirDato (archRegistros, archReporte, ':');
}

void leerImprimirDato (ifstream &archRegistros, ofstream
                       &archReporte, char parada) {
    char caracter = 'a';
    while (true) {
        caracter = archRegistros.get();
        if (caracter == parada) { archReporte.put ('\n'); break; }
        if (caracter != ' ')
            caracter = cambiarMayus (caracter);
        archReporte.put (caracter);
    }
}
```


Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

May Min
x

```

char cambiarMayus(char c) {
    char car = c;
    if (c >= 'a' and c <= 'z') {
        car = c - ('a' - 'A');
    }
    return car;
}

```

```

void imprimirReporteResumen(ifstream &archRegistros,
    ofstream &archReporte) {
    //aquí van las declaraciones de abajo

```

```

    while (true) {
        verificar = archRegistros.get();
        if (verificar == '\n') break;
        archRegistros.putback(verificar);
        leerImprimirReporte(archRegistros, archReporte);
    }
    imprimirResumen(archReporte, ddI, ddF, mmI, mmF, yyyyI, yyyyF,
        totalLlovido, mayorLluvia, contador);
}

```

```

void leerImprimirReporte(ifstream &archRegistros, ofstream
    &archReporte) {

```

```

    int dd, mm, yyyy, ddI, mmI, yyyyI, ddF, mmF, yyyyF, contador=1;
    double cantLlovida, totalLlovido = 0.0, mayorLluvia = 0.0;

```

```

    char descartar; //declaraciones arriba del while

```

```

    int ddM, mmM, yyyyM;

```

```

    archRegistros >> dd >> descartar >> mm >> descartar >> yyyy;

```

```

    if (contador == 1) {

```

```

        ddI = dd;
        mmI = mm;
        yyyyI = yyyy;
    }

```

```

    else {

```

```

        ddF = dd;
        mmF = mm;
        yyyyF = yyyy;
    }

```

Estas declaraciones deberian ir arriba del while

```

    int hora1, minuto1, segundo1, hora2, minuto2, segundo2;

```

```

    archRegistros >> hora1 >> descartar >> minuto1 >> descartar
        >> segundo1;

```

```

    archRegistros >> hora2 >> descartar >> minuto2 >> descartar
        >> segundo2;

```

```

    //... int hora, minuto, segundo;

```

```

    calcularTiempoLlovido(hora1, hora2, minuto1, minuto2,
        segundo1, segundo2, hora, minuto, segundo);

```

```

    archRegistros >> cantLlovida;

```

```

    totalLlovido += cantLlovida;

```

```

    imprimirReporte(archReporte, dd, mm, yyyy, hora, minuto,
        segundo, cantLlovida);
}

```


Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

```
void imprimirResumen(ofstream & archReporte, int ddI,
    int ddF, int mmI, int mmF, int yyyyI, int yyyyF,
    double totalLlovido, double mayorLluvia, int contador) {
    archReporte << endl << "RESUMEN" << endl << "Fecha inicial:";
    archReporte.fill('0');
    archReporte << setw(2) << ddI << '/' << setw(2) << mmI
        << '/' << yyyyI;
    archReporte.fill(' ');
    archReporte << setw(3) << " " << "Fecha final:";
    archReporte.fill('0');
    archReporte << setw(2) << ddF << '/' << setw(2) << mmF
        << '/' << yyyyF << endl; archReporte.fill(' ');
    archReporte << "Cantidad de días registrados: " << contador
        << endl;
    archReporte << "Tiempo total de lluvia: " << totalLlovido
        << endl; Falta el tiempo acumulado.
    archReporte << "Total llovido en el periodo: " << totalLlovido
        << "mm" << endl;
    archReporte << "La mayor precipitación: " << mayorLluvia
        << endl; Debe imprimir también el día relacionado a la mayor precipitación.
}
```

```
void imprimirReporte(ofstream & archReporte, int dd,
    int mm, int yyyy, int hora, int minuto, int
    segundo, double cantLlovida) {
    archReporte.fill('0');
    archReporte << setw(2) << dd << '/' << setw(2) << mm
        << '/' << yyyy << " ";
    archReporte << setw(2) << hora << ':' << setw(2) << minuto
        << ':' << setw(2) << segundo;
    archReporte.fill(' ');
    archReporte << " " << cantLlovida << "mm" << endl;
}
```

Falta implementar el módulo
calcularTiempoLlovido(...). Debería
calcular también la cantidad acumulada
de tiempo (en el módulo mencionado
e en leer Imprimir Reporte).