

Ejercicio 1 – final 2019

El INDEC dispone de un archivo SUELDOS2019.TXT donde cuenta con una muestra de sueldos de todas las provincias del país y de todos los meses del año.

Cada renglón del archivo tiene la información: **Código de provincia (1 a 24)**, **Mes (1 a 12)** y **sueldo**. Puede existir en el archivo más de un juego de datos para provincia y mes.

Se sabe que hay información de todas las provincias y de todos los meses.

Se desea:

- **Mostrar una matriz por pantalla con los sueldos promedio de cada provincia y mes.**
- **Comparando los sueldos totales de cada provincia de enero y diciembre, informar por cada provincia, cual fue el porcentaje de aumento por paritarias.**
- **Informar de que provincia y mes se cuenta con menor cantidad de datos.** Para este punto implementar una función que devuelva en el **nombre la cantidad de datos y devuelva por parámetro el código de provincia y mes.**

SOLO LIBRES:

Se dispone de un archivo PROVINCIAS con el código y nombre de cada una de las provincias.

Realizar un informe ordenado por promedios de sueldos de diciembre con el siguiente formato:

COD PROVINCIA NOMBRE SUELDO PROMEDIO.

Ejercicio 2 (20 puntos)

Escriba un programa C++ que lea el archivo **abono_futbol.txt**, el cual fue previamente creado por otro programa. En dicho archivo, se han guardado los siguientes datos de las contrataciones del servicio del corriente mes: **documento abonado | Monto abono mensual | día de contratación**. Cada uno de los datos viene en un renglón diferente.

Cómo primer dato, **deberá ingresarse la cantidad de días del mes actual.**

Se pide **generar un listado con los importes proporcionales a cobrar por los días del mes en curso (días del mes – día de contratación) mostrando el monto a cobrar.** El formato es el siguiente:

ABONOS PROPORCIONALES

DOCUMENTO DEL ABONADO MONTO A COBRAR

XXXXXXXXXXXX

XXXXX

Ejercicio 2 (20 puntos)

Se plantea un problema, en el cual se le dan todas las acciones del main y los prototipos de las funciones que el mismo utiliza.

Se desea llenar un vector de N elementos con números enteros aleatorios. Para agregar un elemento al vector, el número generado no tiene que estar ya almacenado, ni tampoco ser consecutivo (creciente o decreciente con el último número almacenado). El valor de N se ingresa al inicio.

Se desea además saber cuantos números fueron necesarios de generar para completar el vector.

En el desarrollo de las funciones, agregue las variables que considere necesarias.

```
#include <iostream>
#include <string>
```

```

using namespace std;

bool NoRepetido( int Vec[], int nTamanio, int nNumero );
void NoConsecutivo( int Vec[], int nTamanio, int nNumero , bool &IResultado);
int main(int argc, char *argv[]) {
    int N,nNro;
    bool IResul;
    int nVector[100];
    cout <<"Ingrese la cantidad de elementos: ";
    cin >> N;
    int nl=0,nGenerados=0;
    while (nl<N){
        nGenerados++;
        nNro=rand()%1000;
        if (NoRepetido( nVector, nl ,nNro ) ){
            NoConsecutivo( nVector, nl ,nNro , IResul);
            if (IResul) {
                nVector[ nl ] = nNro;
                nl++;
            }
        }
    }
    for (nl=0;nl<N;nl++){
        cout <<nVector[nl]<<" ";
    }
    cout << "Cantidad de elementos: "<<N<<endl;
    cout << "Cantidad de numeros generados: "<<nGenerados;
    return 0;
}

void NoConsecutivo( int Vec[], int nTamanio, int nNumero , bool &IResultado){
    // esta función verifica que el número que se quiere almacenar
    // no es consecutivo con el último almacenado.
    //OJO con la verificación del primer elemento
    ...
}

bool NoRepetido( int Vec[], int nTamanio, int nNumero ){
    // esta función verifica que el número que se quiere almacenar
    // no se encuentra en el vector.
}

```

Ejercicio 1 (35 puntos)

En un servicio odontológico de un hospital público, se llevan distintos registros para luego efectuar estadísticas.

Cuentan con un archivo denominado mapa_dental.txt donde cada registro contiene la siguiente información: CÓDIGO PIEZA DENTARIA (alfanumérico) - DESCRIPCIÓN.

Al principio del mes, ingresa los datos de los odontólogos que han trabajado en dicho período: NRO MATRÍCULA (1 a 9999) – APELLIDO – NOMBRE – DNI. Estos datos vienen sin orden, y el fin de los mismos está dado por NRO DE MATRÍCULA = 0.

Luego ingresa los datos de cada prestación efectuada, discriminadas por cada odontólogo, de la siguiente manera: NRO DE MATRÍCULA y TOTAL DE PRESTACIONES

y, por cada prestación: CÓDIGO PIEZA DENTARIA – DNI PACIENTE – EDAD – OBRA SOCIAL ('S' si posee, 'N' en caso contrario).

Se desea:

1) Obtener un listado ORDENADO de manera ASCENDENTE por código de pieza dentaria, con la siguiente información:

CÓDIGO PIEZA	DESCRIPCIÓN	CANTIDAD DE PRESTACIONES
XXXX	XXXXXXXXXXXXX	XX

2) Informar la cantidad total de pacientes atendidos y cuántos de ellos poseen obra social.

3) Generar un archivo odontologos.txt que contenga la siguiente información: NRO MATRÍCULA – APELLIDO – NOMBRE – DNI – TOTAL DE PRESTACIONES EN EL MES.

OBSERVACIÓN: Deberá AL MENOS, implementar UNA FUNCIÓN. Queda a su criterio la decisión de nombre, tipo y funcionalidad. Pero, NO SE PERMITE QUE LA FUNCIÓN EMITA INFORMES Y/O LISTADOS.