

# Certamen 1 Introducción al análisis de datos.

orlando soto

June 17, 2022

## 1 Programación básica en c++

Corrija las siguientes lineas de código con errores:

### 1.1

---

```
int N = 10;
int arr[N];
arr[5] = 1.2;
arr[2] = arr[5]*arr[2];
```

---

### 1.2

---

```
#define NR 10
#define NC 10
float **matrix;
matrix = new float[NR];
for (int k = 0; k<NR;k++) matrix[k] = new float[NC];
```

---

### 1.3

---

```
int N = 10;
double *arr = new double[N];
for (int k = N; k>=0;k--) arr[k] =0.;
```

---

### 1.4

---

```
#include <iostream>
class A{
    int a;
};
class B: public A{
```

```

public:
    int b;
};
void main(){
A obj_a;
B obj_b;
obj_b.a = 12;
obj_a.a = 10;
obj_b.b = 5;
std::cout<<"a from A: "<< obj_a.a<<std::endl;
std::cout<<"a from B: "<< obj_b.a<<std::endl;
std::cout<<"b from B: "<< obj_b.b<<std::endl;
}

```

---

## 2 Clases

Suponga que se quiere diseñar un sistema de adquisición de datos que involucra 4 dispositivos. Todos ellos se comunican usando comandos SCPI sobre el estandar RS232 con un computador. Se dispone de una clase llamada Serial que se encarga de realizar la comunicación. La clase Serial tiene los siguientes métodos:

```

Serial::Serial(int id = 0);
std::string Serial::Read(int n = -1);
int Serial::Write(std::string msg);

```

---

El id del constructor es un número que sirve para identificar el dispositivo. Los 4 dispositivos a controlar son: Medidor de corriente (**DMM**), data logger (**DL**), un switch (**SW**) y un generador de señales (**SG**). El programa principal que utilizará los dispositivos es el siguiente:

```

// including sleep() for windows and unix. unsigned sleep(unsigned seconds)
#ifdef _WIN32
#include <Windows.h>
#else
#include <unistd.h>
#endif
#include "DMM.h"
#include "DL.h"
#include "SW.h"
#include "SG.h"
#include <fstream>
#include <iostream>
#define MIN_IDC 0.4
#define MAX_IDC 1.5
#define NEVPERSIG 1000
int main(){
    unsigned long nev = 0;
    std::ofstream ofile("datos_DAQ.txt");
    DMM *dmm = new DMM();
    DL *dl = new DL();

```

```

SW *sw = new SW();
SG *sg = new SG();
float idc = -1;
while (true){
    idc = dmm->ReadCurrent();
    if( !(MIN_IDC < idc && idc < MAX_IDC) ){
        SW -> TurnOff();
        sleep(30);
        SW->TurnOn();
        sleep(5);
        continue;
    }
    DL -> GetData();
    ofile<<"event: "<<nev<<std::endl;
    ofile<<"I DC: "<<idc<<std::endl;
    ofile<<"Data: "<<DL->Data<<std::endl;
    nev++;
    if (!(nev % NEVPERSIG) && nev)
        SG->SetNext();
}
return 0;
}

```

---

## 2.1

Explique brevemente que realiza la rutina.

## 2.2

Crear la clase e implementar los métodos que se muestran en el programa principal (mínimo), para el instrumento asignado (ver sorteo). Los comandos SCPI que reciben los instrumentos se pueden ver en la tabla 2.2.

instrument	SCPI comm.	example of action or return message
<b>DMM</b>	":INSTR:IDC?"	"1.02E00"
SG	":INSTR:SET:SIN"	Set the sinusoidal function output
SG	":INSTR:SET:STEP"	Set the step function output
SW	":INSTR:ON"	Turn on the switch
SW	":INSTR:OFF"	Turn off the switch
DL	":INSTR:DATA?"	"0.785 0.3345 1.452 0.945 1.100 0.564 0.112 5.123"

Table 1: scpi commands for the instruments

## 3 Complejidad algoritmos

Suponga que se tiene un algoritmo que se puede descomponer de la siguiente forma:

$$T(n) = c_0 T(n/2) + c_1, \quad T(1) = t_1, \quad (1)$$

en donde  $c_0$ ,  $c_1$  y  $t_1$  son constantes.

Resuelva la ecuación recursiva y obtenga la expresión de  $T(n)$  en función de  $n$ ,  $c_0$ ,  $c_1$  y  $t_1$ , luego calcule el orden de complejidad del algoritmo para  $c_0 = 1$  y  $c_0 = 2$ . *Hint: recuerde la expresión para una suma geométrica y la regla de L'Hôpital.*