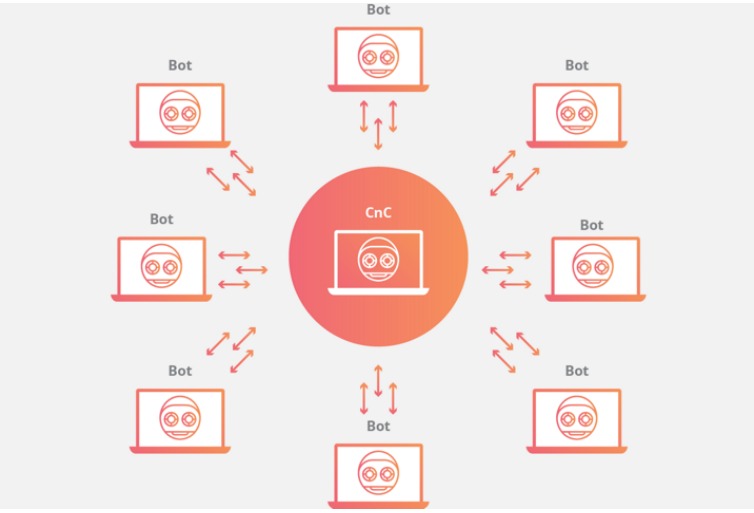
Tercer Parcial

30 de noviembre de 2020

Indicaciones generales

- Este es un examen **individual** con una duración de **110 minutos: de 3:00 pm a 4:50 pm**.
- En `e-aulas` puede acceder a las diapositivas y a la sección correspondiente a este parcial.
- Solamente será posible tener acceso a `e-aulas.urosario.edu.co` y a los sitios web correspondientes a la documentación de Python y C++ dispuestos por el profesor.
- Celulares y otros dispositivos electrónicos no deben ser utilizados durante el examen.
- El estudiante solo podrá disponer de hojas en blanco como borrador de apuntes (opcional).
- El estudiante puede tener una hoja manuscrita de resumen (opcional). Esta hoja debe estar marcada con nombre completo.
- Como este parcial se realizará virtualmente, solo se considerarán las respuestas al parcial de aquellos estudiantes que se conecten a través de la plataforma, **enciendan su cámara y micrófono** durante la realización del parcial, y suban el parcial a e-aulas antes de retirarse del aula virtual.
- La actividad en `e-aulas` se cerrará a la hora en punto acordada. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de `e-aulas` será calificado tal como está. Si ningún tipo de material es entregado por este medio, la nota de la evaluación será 0.0.
  Se aconseja subir a e-aulas versiones parciales de la solución a la actividad.
- **Cualquier incumplimiento de lo anterior conlleva la anulación del examen.**
- Las respuestas deben estar totalmente justificadas.
- Use las plantillas p1_plantilla.py, y pY_plantilla.cpp con `Y = 2,3`, para dar su solución.
- **Entrega**: un archivo con extensión `'.py'` (Python) y dos archivos con la extensión `'.cpp'` según el caso, conteniendo el código solución, basado en las plantillas. Nombre los archivos como `p1.py`, y `pZ.cpp` con `Z = 2,3`.
  Importante: no use acentos ni deje espacios en los nombres de los archivos que cree.

1. [60 ptos.]Una *botnet* es una red (net) de robot's (bots) que pueden realizar diferentes tipos de ataques cibernéticos. Los bots que hacen parte de una botnet fueron en algún momento dispositivos inofensivos, sin embargo después de ser infectados por un virus o troyano, se convirtieron en dispositivos que siguen instruciones de un *Servidor de comando y control (CnC)* que los controla remotamente y que generalmente es administrado por un hacker. Adicionalmente, existen diferentes tipos de bots dependiendo de las *capacidades de ataque* que tengan, por ejemplo: clicker, spammer, destroyer, etc. Una ilustración de una botnet se muestra a continuación:



Implemente una clase *commandControlServer* que represente el *Servidor de comando y control*, la cual debe tener los atributos y métodos que se indican abajo. Igualmente, implemente una superclase *bot*, una subclase *botSpammer*, *botClicker* y *superbot* con los atributos y métodos que se indican abajo. Finalmente implemente una clase *botnet* que contenga al menos: 1 objeto *commandControlServer*, y un número indefinido de objetos de tipo *botSpammer*, *botClicker* y *superbot*.

```
#Clase commandControlServer
clase commandControlServer
        Atributos de la clase commandControlServer:
        ubicacion: De tipo string. Almacena la ciudad donde el
            commandControlServer esta ubicado.
        organizacion: De tipo string. Almacena la organizacion propietaria
             del commandControlServer.
        Metodos de la clase commandControlServer:
        __init__: construye un objeto commandControlServer con los
            atributos inicialmente vacios.
        __str__: retorna todos los atributos definidos para el
            commandControlServer.
```

```
#Clase bot como superclase
clase bot
        Atributos de la clase bot:
        tipo: De tipo string. Almacena el tipo de dispositivo, el cual
            puede ser: smartphone, tablet, pc o server.
        ram: De tipo entero. Almacena el valor de la memoria ram del
            dispositivo en Mb, por ejemplo: 1024.
        hd: De tipo entero. Almacena el valor del disco duro del
            dispositivo en Gb, por ejemplo: 80.
        Metodos de la clase bot:
        __init__: construye un objeto bot con 3 argumentos de entrada:
            tipo, ram y hd.
        __str__: retorna todos los atributos definidos para el bot.
```

```
#Clase botSpammer que hereda de la clase bot
clase botSpammer
        Atributos de la clase botSpammer:
```

```
        identificador: De tipo int. Contiene un entero que identifica el bot.
        categoria: De tipo string. Contiene la capacidad de ataque del bot, que en este caso es: spammer
        Metodos de la clase botSpammer:
        __init__: inicializa los atributos de un objeto de tipo botSpammer con 4 argumentos de entrada: identificador, tipo, ram y hd.
        __str__: retorna todos los atributos definidos para la clase botSpammer.
        generateSpam: retorna el texto "Send fake emails!!"
```

```
#Clase botClicker que hereda de la clase bot
clase botClicker
        Atributos de la clase botClicker:
        identificador: De tipo int. Contiene un entero que identifica el bot.
        categoria: De tipo string. Contiene la capacidad de ataque del bot, que en este caso es: clicker
        Metodos de la clase botClicker:
        __init__: inicializa los atributos de un objeto de tipo botClicker con 4 argumentos de entrada: identificador, tipo, ram y hd.
        __str__: retorna todos los atributos definidos para la clase botClicker.
        generateClicks: retorna el texto "Make a click on a Pay-per-Click website!!"
```

```
#Clase superbot que hereda de la clase botSpammer y de la clase botClicker
clase superbot
        Atributos de la clase superbot:
        categoria: De tipo string. Contiene la capacidad de ataque del bot, que en este caso es: super
        Metodos de la clase botClicker:
        __init__: inicializa los atributos de un objeto de tipo superbot con 4 argumentos de entrada: identificador, tipo, ram y hd.
        __str__: retorna todos los atributos definidos para la clase superbot.
        generateFullAttack: retorna el texto "Spamming and clicking!!"
```

```
#Clase botnet que implementa una coleccion de objetos
clase botnet
        Atributos de la clase botnet:
        equipos_botnet: Lista de objetos que pertenecen a la botnet
        Metodos de la clase botnet:
        __init__: construye la botnet con una lista equipos_botnet inicialmente vacia.
        __str__: retorna str con los atributos de cada uno de los objetos de la botnet.
        addDevice: agrega una lista de dispositivos a la lista equipos_botnet.
        removeDevice: remueve un dispositivo de la lista equipos_botnet.
```

Valide su implementación creando objetos de tipo commandControlServer, botSpammer, botClicker, superbot y botnet, y probando los métodos de cada uno de ellos como se indica en la plantilla adjunta usable para resolver el ejercicio.

El resultado de la ejecución del ejercicio debería ser el siguiente:

```
Creacion del servidor CnC:
s1: Servidor CnC ubicado en Kiev de Anonymous
Creacion del bot z1:
z1: Dispositivo 1 de tipo smartphone con ram: 1024, hd: 8, de la categoria spammer
z1: Send fake emails!!
Creacion del bot z2:
z2: Dispositivo 2 de tipo tablet con ram: 2048, hd: 10, de la categoria clicker
z2: Make a click on a Pay-per-Click website!!
Creacion del bot z3:
z3: Dispositivo 3 de tipo pc con ram: 6144, hd: 60, de la categoria super
z3: Make a click on a Pay-per-Click website!!
z3: Send fake emails!!
z3: Spamming and clicking!!
Botnet b1:
Servidor CnC ubicado en Kiev de Anonymous
Dispositivo 1 de tipo smartphone con ram: 1024, hd: 8, de la categoria spammer
Dispositivo 2 de tipo tablet con ram: 2048, hd: 10, de la categoria clicker
Dispositivo 3 de tipo pc con ram: 6144, hd: 60, de la categoria super
Botnet b1:
Servidor CnC ubicado en Kiev de Anonymous
Dispositivo 1 de tipo smartphone con ram: 1024, hd: 8, de la categoria spammer
Dispositivo 2 de tipo tablet con ram: 2048, hd: 10, de la categoria clicker
```

2. [20 ptos.]Escriba en C++ la función double sum(int inf, int sup) que recibe dos enteros inf y sup y, en caso que inf < sup calcula el promedio de todos los enteros desde inf hasta sup y lo retorna. En caso contrario retorna el número −1.e10. Note que el valor de retorno es double.

   Utilice la plantilla adjunta para resolver el ejercicio.

3. [20 ptos.]El siguiente código debe realizar la suma del contenido de las variables a y b.

```
int a = 3;
int b = 5;
int sum;

/*Code here*/

/**********/

cout << "sum = " << sum << endl;
```

   Complete el código de manera que se realiza la suma, pero en lugar de hacerlo directamente (es decir a+b), defina apuntadores a las variables a y b y realice la suma usando dichos apuntadores.

   Use la plantilla adjunta para contestar el punto.