



TERCER PARCIAL

16 de mayo de 2020

Indicaciones generales

- Este es un examen **individual** con una duración de **120 minutos: de 8:00 a 10:00**.
- 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 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 `plantillaP1.cpp`, `plantillaP2.cpp` y `plantillaP3.cpp` para dar su solución.
- **Entrega:** tres archivos con extensión `‘.cpp’` (C++) según el caso, conteniendo el código solución, basado en las plantillas. Nombre los archivos como `p1.cpp`, `p2.cpp` y `p3.cpp`.

Importante: no use acentos ni deje espacios en los nombres de los archivos que cree.

1. [30 ptos.] Escriba una función en C++ llamada `tempValidation` que reciba de manera indefinida valores de temperatura de pacientes registrados por un médico a través del teclado (entrada estándar). La función debe capturar valores hasta que el médico ingrese el valor de 0, en tal caso la función deberá retornar el promedio de las temperaturas ingresadas hasta el momento. Cada valor ingresado por el médico debe irse imprimiendo en consola.

Implemente una función `main()` que invoque a la función `tempValidation`, reciba el valor retornado por dicha función y lo imprima en consola. Utilice la plantilla adjunta para resolver el ejercicio. A continuación se presenta un ejemplo de la consola al ejecutar el programa (función `main`):

```
1 | Este programa captura valores de temperatura y retorna el
   | promedio
2 | Ingrese los valores de temperatura, presione '0' para terminar:
3 | 25
4 | El valor ingresado fue: 25
```



```
5 | 26
6 | El valor ingresado fue: 26
7 | 23
8 | El valor ingresado fue: 23
9 | 28
10 | El valor ingresado fue: 28
11 | 34
12 | El valor ingresado fue: 34
13 | 39
14 | El valor ingresado fue: 39
15 | 40
16 | El valor ingresado fue: 40
17 | 0
18 | El valor ingresado fue: 0
19 | El promedio de las temperaturas ingresadas fue: 30.7143
```

2. [30 ptos.] Implemente en C++ la función

```
1 || void greatest_prime_number(int n, int &p);
```

que calcula p , el mayor número primo menor que n . (Ejemplo: si $n=10$, entonces, $p=7$)

Use el archivo `plantillaP2.cpp`. Use la función `main` que solicita un valor de n , invoca la función implementada `greatest_prime_number` e imprime el mayor número primo menor que n (por ejemplo, si $n=10$, entonces $p=7$).

Nota: Puede crear una función adicional que evalúe si un número es primo.

3. [40 ptos.] La plantilla que acompaña a este punto define las variables x y y una tras de otra, de manera que estén contiguas en memoria, así como el prototipo no implementado de una función `foo`. Partiendo de ésta, realice las siguientes acciones:
- a) [10 ptos.] Cree un apuntador que apunte a la variable x
 - b) [15 ptos.] Dentro de la función `foo`, sume 1 al valor de x , al cuál está apuntando p
 - c) [15 ptos.] Dentro de la función `foo`, incremente el apuntador en una posición (8 Bytes) de manera que apunte ahora a y , y divida el valor guardado en y entre dos.