

CCI34-A Programación Avanzada

Taller N°1

Semestre 1, Año 2017

1. Escriba un programa en C++ que, haciendo uso de la función `sizeof`, escriba en la salida de pantalla el tamaño en bytes de los tipos numéricos, `short` (signed y unsigned), `int` (signed y unsigned), `long` (signed y unsigned), `long long` (signed y unsigned), `float` (signed y unsigned) y `double` (signed y unsigned).

Ejemplo:

```
cout << "El tipo int usa " << sizeof(int) << " bytes" << endl;
```

2. Escriba un programa en C++ que, para los tipos de numéricos `short` (signed y unsigned), `int` (signed y unsigned), `long` (signed y unsigned), `long long` (signed y unsigned), `float` (signed y unsigned) y `double` (signed y unsigned), escriba el valor mínimo y máximo posible.

Nota: Incluir y trabajar con la directiva `<climits>`.

3. En el ejercicio 1 se le pidió buscar los tamaños de cada tipo numérico, en el ejercicio 2 se le pidió buscar los valores extremos de cada tipo. A partir de ello reflexione:
 - (a) ¿Que pasaría si usamos el tipo `char` en el programa del ejercicio 1? Verifique su suposición escribiendo la línea correspondiente.
 - (b) ¿Tiene sentido pedir el tamaño de un `unsigned char`? Compruebe el resultado con su programa.
 - (c) ¿Cuales serían los límites para los dos casos anteriores?
4. Escriba un programa en C++ que, a partir de un menú de selección por pantalla, permita realizar alguna de las siguientes operaciones:

- Operación #1: Altura de un triángulo isóceles de lados a y b . La altura se calcula como $h = \sqrt{a^2 - \frac{b^2}{4}}$
- Operación #2: $\sum_{i=1}^n (2i - 1)^3$
- Operación #3: $x^3 + 2x^2 + \sqrt[3]{2x - 1}$
- Operación #4: $\frac{a+b}{c-d}$

Importante:

- El menú debe ser implementado con un **switch** y el programa sólo debe finalizar hasta que el usuario ingrese la opción 0.
 - Cada operación debe ser definida en su propia función.
 - Los parámetros de cada función deben ser solicitados e ingresados por teclado.
 - Debe validar divisiones por 0 y el ingreso de valores no numéricos.
5. Escriba un programa en C++ que calcule si un número x , entregado por teclado, es primo o no. Calcule además sus divisores.
- Nota: Un número se considera primo cuando sólo es divisible por si mismo y 1. Debe utilizar el operador %.*
6. Escriba un programa en C++ que localice un caracter dentro de una cadena de texto y retorne su posición en relación a un inicio dado. La declaración o prototipo de la función es la siguiente:

```
int chrpos(char* cadena, char caracter, int offset = 0);
```

Dónde **cadena** es el texto original, **caracter** es la letra, número o signo a buscar y **offset** la posición desde donde se comenzará a buscar. Por defecto se busca desde la posición 0. Si se especifica una posición negativa, se debe buscar de atrás para adelante. El valor devuelto es la posición **relativa a offset**. Si la letra no se encuentra, debe retornar -1 .

Ejemplo:

```
char cadena[] = "este es un ejemplo";
cout << chrpos(cadena, 's'); << endl; // Imprime 1
cout << chrpos(cadena, 's', 1); << endl; // Imprime 0
cout << chrpos(cadena, 'n', 3); << endl; // Imprime 6
cout << chrpos(cadena, 'o', -1); << endl; // Imprime 0
cout << chrpos(cadena, 'j', -1); << endl; // Imprime 5
cout << chrpos(cadena, 'n', -3); << endl; // Imprime 6
cout << chrpos(cadena, 'x'); << endl; // Imprime -1 (no encontrado)
```

7. Escriba un programa en C++ que traspase **n** números aleatorios al archivo `.txt nombreFichero` y luego imprima el contenido del mismo por pantalla. Las variables **n** y **nombreFichero** deben ser obtenidas desde los argumentos del main de la siguiente forma:

```
ejercicio7.exe 10 archivo.txt
```

El orden de los parámetros podría variar, por ende deberá validar que el primer parámetro sea un número válido mayor a 0 y el segundo parámetro contenga la extensión `.txt`. Imprimir mensaje de error en caso contrario.

Nota: Revisar la documentación sobre generación de números aleatorios y cómo pasar y leer parámetros desde la función main. Recuerde que para transformar una variable del tipo `char` a `int` debe usar la función `atoi`.*

Consideraciones Finales

1. Plazo de entrega máximo: **lunes 17 abril hasta las 23:59**.
2. Los programas deben ser entregados en una carpeta comprimida con su nombre y apellido. Por cada pregunta debe existir un archivo `.cpp` (ej: `ejercicio1.cpp`, `ejercicio2.cpp`, etc.). Sólo los archivos con extensión `.cpp` deben existir dentro de la carpeta. Ningún otro archivo será aceptado.
3. Las respuestas a las preguntas abiertas del ejercicio 3 deberá agregarlas al principio del código como comentario. Ej:

```
/*  
(a) respuesta a la pregunta a.  
(b) respuesta a la pregunta b.  
(c) respuesta a la pregunta c.  
*/  
#include <iostream>  
int main() { }
```

4. El archivo comprimido debe ser enviado por correo a rdelafue@unap.cl. El asunto del correo debe ser exactamente el siguiente: [CCI34-A] Taller 1 - Apellidos, Nombre

Ej: [CCI34-A] Taller 1 - De la Fuente Orellana, Roberto

5. Documente y formatee bien su código fuente. ¡Respete la indentación o sangrado de los bloques!



Universidad
ARTURO PRAT
del Estado de Chile

Facultad de Ingeniería y Arquitectura
Ingeniería Civil en Computación e Informática

6. **Se descontarán décimas por no respetar las reglas o por entregas atrasadas.**
7. Dudas o consultas a rdelafue@unap.cl
8. Material de apoyo en Aula Virtual y en <http://www.cplusplus.com>
9. ¡A estudiar, practicar y buena suerte!