

Tarea 3

Profesor: Diego Arroyuelo

Ayudantes: Fernanda Weiss, Manuel Calquín

`fernanda.weiss.13@sansano.utfsm.cl`,

`manuel.calquin.14@sansano.utfsm.cl`

Fecha de entrega: 22 de diciembre, 2017

Plazo máximo de entrega: 5 horas.

Reglas del Juego

La presente tarea debe hacerse en grupos de 3 personas. Toda excepción a esta regla debe ser conversada con los ayudantes **ANTES** de comenzar la tarea. No se permiten de ninguna manera grupos de más de 3 personas. Las tareas deben compilarse en los computadores que se encuentran en el laboratorio B-032 (LDS). Deben usarse los lenguajes de programación C o C++. Se recomienda compilar en el terminal usando `gcc archivo.c -o output -Wall` (en el caso de lenguaje C) o `g++ archivo.cpp -o output -Wall` (en el caso de C++). Si usa C++, está permitido emplear estructuras de datos de la `std`, en caso de ser necesarias.

Subsecuencia Palindrómica de Largo Máximo

Dado un string $s[1..n]$ de largo n , una subsecuencia de s es otro string formado por caracteres de s , en el mismo orden en que aparecen en el string. Por ejemplo, si $s = \text{"mesopotamia"}$, algunas posibles subsecuencias son `"sptai"`, `"sopota"`, `"eooaia"`, `"mesopotamia"`, etc. Diseñe un algoritmo que permita encontrar el largo de la subsecuencia palindrómica más larga de un string dado. Use programación dinámica para obtener un tiempo de ejecución $\Theta(n^2)$. Para $s = \text{"mesopotamia"}$, algunas subsecuencias palindrómicas son `"m"`, `"mem"`, `"msm"`, `"mom"`, `"mpm"`, `"mtm"`, `"mam"`, `"opo"`, `"ama"`, `"aia"`, `"aa"`, y `"moom"`, entre otras. La respuesta para este ejemplo debería ser 5, que corresponde a la subsecuencia `"mopom"`.

Formato de Entrada

La entrada de datos se hará a través de la entrada standard (`stdin`). Cada línea de la entrada contiene un único string, que será entregado como entrada a su algoritmo. La entrada es finalizada con `EOF`.

Hint: para probar su programa de una mejor manera, ingrese los datos de entrada con el formato indicado en un archivo de texto (por ejemplo, el archivo `input.dat`). Luego, ejecute su programa desde la terminal, redirigiendo la entrada standard como a continuación:

```
./tarea2 < input.dat
```

De esta manera evita tener que entrar los datos manualmente cada vez que prueba su programa, y evita errores.

Formato de Salida

La salida de datos se hará a través de la salida standard (`stdout`). Por cada string dado en la entrada, se debe mostrar una línea que contenga el largo de la subsecuencia palindrómica más larga correspondiente.

Bonus por Eficiencia

Se dará un bonus de 25 puntos a las 5 tareas que ejecuten más rápido. Dicho bonus se podrá usar en cualquiera de las tareas del curso.

Entrega de la Tarea

La entrega de la tarea debe realizarse enviando un archivo comprimido llamado

`tarea3-apellido1-apellido2-apellido3.tar.xz`

(reemplazando sus apellidos según corresponda) en el sitio Aula del curso, a más tardar el día 22 de diciembre de 2017, a las 23:55:00 hrs (Chile Continental), el cual contenga:

- Los archivos con los códigos fuentes necesarios para el funcionamiento de la tarea. Los archivos deben compilar!
- `nombres.txt`: Nombre, ROL y qué programó cada integrante del grupo.
- `README.txt`: Instrucciones de compilación u observaciones, en caso de ser necesarias.
- `Makefile`: Reglas de compilación para compilar su tarea con el comando `make`. Se exige el uso del flag `-Wall`.

Restricciones y Consideraciones

- Por cada día de atraso (o fracción) en la entrega de la tarea se descontarán 10 puntos en la nota.
- El plazo máximo de entrega es 5 horas después de la entrega original.
- Pueden programar la tarea en C o C++ según ustedes consideren conveniente.
- Las tareas deben compilar en los computadores que se encuentran en el laboratorio B-032. **Las tareas que no compilen no serán revisadas y serán calificadas con nota 0.**
- Por cada *Warning* en la compilación se descontarán 5 puntos.
- Si se detecta **COPIA** la nota automáticamente será 0 (CERO), para todos los grupos involucrados. El incidente será reportado al Jefe de Carrera (JLML).
- La prolijidad, orden y legibilidad del código fuente es obligatoria. Habrá descuentos si alguno de estos ítems no se cumple.