

---

## Práctica de PRO2 - Primavera 2019 - entrega definitiva X26293\_es

---

Este problema es el único canal disponible para realizar **la entrega definitiva** de la práctica y también **la entrega de los materiales para evaluar la competencia transversal “Treball en equip”**. Tened en cuenta que

- Esta es la entrega que dará lugar a la mayor parte de la nota de la práctica (90%); la nota de la entrega tendrá una parte procedente de corrección automática y otra procedente de corrección manual
- No superar ningún juego de pruebas conllevará un cero en la nota de la entrega
- En esta entrega pedimos más cosas, que detallamos a continuación **en negrita**, además del código de la práctica y el Makefile, que se pedían en la entrega provisional
- Recomendamos usar el problema de la entrega provisional para probar el código hasta tener totalmente acabada la práctica

### Entrada

Una secuencia de instrucciones y datos que siguen el formato del enunciado de la práctica y del juego de pruebas público.

### Salida

Una secuencia de resultados que siguen el formato del enunciado de la práctica y del juego de pruebas público.

### Observación

El Jutge prueba vuestro código mediante 4 juegos de pruebas:

- sample: el juego de pruebas público
- privat1: combinación de los juegos de pruebas de la entrega intermedia
- privat2: juego de pruebas que explora situaciones de las tres funcionalidades que no aparecen en la entrega intermedia (codificar, decodificar, añadir/actualizar idioma), sin controlar la eficiencia
- privat3: juego de pruebas que hace énfasis en la eficiencia de algunas de las funcionalidades que no aparecen en la entrega intermedia

En un fichero llamado `practica.tar` tenéis que entregar

- Los ficheros .hh y .cc de las clases y el programa principal **con los comentarios Doxygen correspondientes**
- El fichero Makefile, que usaremos para generar y probar el ejecutable

- Un fichero `html.zip`, obtenido comprimiendo la carpeta `html` del Doxygen generado a partir de los `.hh` y `.cc` anteriores. Dicha documentación no solo ha de incluir la parte pública de las clases (como en la entrega de la especificación) sino también la parte privada (atributos y métodos privados) y la implementación de todas las operaciones
- Un único integrante de cada equipo ha de entregar un fichero `.zip` con los materiales requeridos para evaluar la competencia “Treball en equip”, siguiendo las instrucciones contenidas en el documento correspondiente, disponible en el apartado “Pràctica” de la página web de la asignatura

Tened en cuenta las siguientes restricciones:

- El fichero que contiene el programa principal se ha de llamar `program.cc`
- El Makefile ha de generar un ejecutable llamado `program.exe`
- Es importante que uséis las opciones de compilación del Jutge de PRO2 (ved Documentation → Compilers → PRO2 a [www.jutge.org](http://www.jutge.org))
- No usar la opción `-D_GLIBCXX_DEBUG` o usarla de forma incorrecta podrá ser penalizado
- **Si no se entrega la documentación generada por el Doxygen, o ésta es incompleta (por ejemplo, si no se ven los elementos privados o la implementación de todas las operaciones), la nota de la corrección manual será cero. Comprobad que vuestra documentación es correcta antes de la entrega. Es necesario usar un Doxyfile basado en el de la sesión 10 de laboratorio para generar correctamente la documentación**

Producid el fichero `practica.tar` con la instrucción Linux

```
tar -cvf practica.tar fitxer1 fitxer2 fitxer3 ...
```

desde el directorio/carpeta donde tengáis los ficheros que vais a entregar. Incluid esta instrucción en vuestro Makefile, de forma que el `.tar` se pueda generar ejecutando `make practica.tar`. Con eso reduciréis el riesgo de error en sucesivas entregas. El Jutge no acepta `.tar` donde los ficheros estén dentro de carpetas. Recomendamos usar GNU tar para reducir el riesgo de que el fichero `practica.tar` sea incompatible con el Jutge. No es necesario incluir `BinTree.hh` en `practica.tar`.

### Ejemplo de entrada 1

```
2
```

```
idioma_corto
```

```
3
```

```
a 4
```

```
b 7
```

```
_ 5
```

```
idioma_medio
```

```
8
```

```
ç 2
```

```
e 7
```

```
f 3
```

```
g 8
```

```
_ 4
```

```
a 4
```

```
i 6
```

```
À 2
```

```
tabla_frec idioma_corto
```

```
tabla_frec idioma_medio
```

```
treecode idioma_corto
```

```
treecode idioma_medio
```

```
codigos idioma_corto todos
```

```
codigos idioma_medio todos
```

```
codigos idioma_corto a
codigos idioma_corto k
codigos idioma_medio À
codifica idioma_medio
efghifhgg_xxx_bbb_c

codifica idioma_medio
gafa_gafe_figa_Àfegia

decodifica idioma_medio
0110011101001001011111

decodifica idioma_medio
111011011111101101111011011011111100100111

decodifica idioma_corto
1

anadir/modificar idioma_corto
3
a 4
b 3
d 5

tabla_frec idioma_corto

treecode idioma_corto

codigos idioma_corto todos

tabla_frec basic2-1

treecode basic2-1

codigos basic2-1 todos

codigos basic2-1 5

codifica basic2-1
abddba

decodifica basic2-1
1101010011

anadir/modificar idioma_cortisimo
3
ç 2
_ 10
à 5

tabla_frec idioma_cortisimo

treecode idioma_cortisimo

codigos idioma_cortisimo todos

codifica idioma_cortisimo
ççç_ççà_ç

codifica idioma_cortisimo
x

decodifica idioma_cortisimo
101010

decodifica idioma_cortisimo
1001
fin
```

## Ejemplo de salida 1

Tabla de frecuencias de idioma\_corto:

\_ 5  
a 4  
b 7

Tabla de frecuencias de idioma\_medio:

\_ 4  
a 4  
e 7  
f 3  
g 8  
i 6  
À 2  
Ç 2

Treecode de idioma\_corto:

recorrido en preorden:

\_ab 16  
b 7  
\_a 9  
a 4  
\_ 5  
recorrido en inorden:  
b 7  
\_ab 16  
a 4  
\_a 9  
\_ 5

Treecode de idioma\_medio:

recorrido en preorden:

\_figaÀÇe 36  
aÀÇe 15  
e 7  
aÀÇ 8  
a 4  
ÀÇ 4  
À 2  
Ç 2  
\_fig 21  
g 8  
\_fi 13  
i 6  
\_f 7  
f 3  
\_ 4  
recorrido en inorden:  
e 7  
aÀÇe 15  
a 4  
aÀÇ 8  
À 2  
ÀÇ 4  
Ç 2  
\_figaÀÇe 36  
g 8  
\_fig 21  
i 6  
\_fi 13

f 3  
\_f 7  
\_ 4

Codigos de idioma\_corto:

\_ 11  
a 10  
b 0

Codigos de idioma\_medio:

\_ 1111  
a 010  
e 00  
f 1110  
g 10  
i 110  
À 0110  
Ç 0111

Codigo de a en idioma\_corto:

a 10

Codigo de k en idioma\_corto:

El idioma no existe o el caracter no esta en el idioma

Codigo de À en idioma\_medio:

À 0110

Codifica en idioma\_medio el texto:

efghifhg\_g\_xx\_bbb\_c

El texto no pertenece al idioma; primer caracter que fa

Codifica en idioma\_medio el texto:

gafa\_gafe\_figa\_Àfegia

1001011100101111100101110001111111011010010111101101110

Decodifica en idioma\_medio el texto:

0110011101001001011111

El texto no procede de una codificacion del idioma; ult

Decodifica en idioma\_medio el texto:

111011011111101101111011011011111100100111

fi\_fi\_Ài\_faÇ

Decodifica en idioma\_corto el texto:

1

El texto no procede de una codificacion del idioma; ult

Modificado idioma\_corto

Tabla de frecuencias de idioma\_corto:

\_ 5  
a 8  
b 10  
d 5

Treecode de idioma\_corto:

recorrido en preorden:  
\_dab 28  
b 10  
\_da 18

a 8	Tabla de frecuencias de idioma_cortisimo:
_d 10	_ 10
_ 5	à 5
d 5	ç 2
recorrido en inorden:	
b 10	Treecode de idioma_cortisimo:
_dab 28	recorrido en preorden:
a 8	_àç 17
_da 18	àç 7
_ 5	ç 2
_d 10	à 5
d 5	_ 10
Codigos de idioma_corto:	recorrido en inorden:
_ 110	ç 2
a 10	àç 7
b 0	à 5
d 111	_àç 17
	_ 10
Tabla de frecuencias de basic2-1:	Codigos de idioma_cortisimo:
El idioma no existe	_ 1
Treecode de basic2-1:	à 01
El idioma no existe	ç 00
Codigos de basic2-1:	Codifica en idioma_cortisimo el texto:
El idioma no existe	ççç_ççà_ç
Codigo de 5 en basic2-1:	0000001000001100
El idioma no existe o el caracter no esta	Codifica en idioma_cortisimo el texto:
Codifica en basic2-1 el texto:	xen el idioma
abddba	El texto no pertenece al idioma; primer caracter que fa
El idioma no existe	Decodifica en idioma_cortisimo el texto:
Decodifica en basic2-1 el texto:	101010
1101010011	El texto no procede de una codificacion del idioma; ult
El idioma no existe	Decodifica en idioma_cortisimo el texto:
Anadido idioma_cortisimo	1001
	—ç—

## Información del problema

Autoría: PR02

Generación: 2026-01-25T21:06:07.624Z

© Jutge.org, 2006–2026.  
<https://jutge.org>