
Práctica de PRO2 - Otoño 2022 - entrega intermedia

X52595_es

Este problema corresponde a la entrega intermedia de la práctica. Tened en cuenta que

- esta no es la entrega definitiva de la práctica completa, ni siquiera es una entrega provisional de la práctica completa
- en esta entrega se ha de proporcionar todo el código necesario (clases, programa principal y, opcionalmente, Makefile) para poder usar todas las funcionalidades de la práctica excepto "codificar_permutacion_guardado", "codificar_permutacion" y "decodificar_permutacion".
- en esta entrega no pedimos carpetas generadas mediante `doxygen`, pero los ficheros de las clases pueden llevar todos los comentarios `doxygen` que queráis; de hecho recomendamos que incluyáis las especificaciones de todas las operaciones y que sean lo más definitivas posible
- esta entrega no tendrá nota manual

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 vuestras entregas mediante 4 juegos de pruebas:

- sample: el juego de pruebas público
- privat1: situaciones que no aparecen en el público
- privat2: muchos alfabetos, mensajes y operaciones de codificar/decodificar
- privat3: mensajes grandes

En un fichero llamado `practica.tar` tenéis que entregar los ficheros `.hh` y `.cc` de las clases y el programa principal. Opcionalmente, podéis entregar el Makefile que estáis usando para generar y probar el ejecutable.

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 definidas en el menú Documentación → Compilers → PRO2 de `www.jutge.org` para funcionar en igualdad de condiciones con el Judge
- no usar la opción `-D_GLIBCXX_DEBUG` o usarla de forma incorrecta podrá ser penalizado

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. Os convendrá incluir en vuestro Makefile una regla con esta instrucción, 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.

Ejemplo de entrada 1

| | |
|--|--|
| 6 | Contrato_frances_1 full_interv |
| abc | Assembler1 full_interv |
| abc | switch_to_pm: / cli / lgdt [gdt_descriptor] / mov eax, |
| base_adn | listar_alfabetos |
| GATC | listar_mensajes |
| base_arn | nuevo_mensaje |
| ACGU | Contrato_frances_2 full_no_interv |
| minusculas | Le contrat d'ouverture d'un livret A prevu a l'article |
| abcdefghijklmnopqrstuvwxyz | nuevo_mensaje |
| full_interv | RNA_string_2 base_arn |
| !"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKL | nm |
| | RNA_string_2 base_arn |
| | nm |
| full_no_interv | nm |
| ABCDEFGHIJKLMN | nm |
| OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{ }~ | nm |
| 8 | nm |
| Hell minusculas | nm |
| dont worry if there is a hell below we are | nm |
| | nm |
| Resistire full_interv | nm |
| Cuando pierda todas las partidas / Cuando | nm |
| Qualsevol_nit full_interv | nm |
| Fa una nit clara i tranquila, hi ha la lluna | nm |
| | nm |
| DNA_string_1 base_adn | nm |
| ATGGCATTAAATGGCATTAAATGGCATTAA | nm |
| | nm |
| DNA_string_2 base_adn | nm |
| CATTAAATGGCATTACATTAAATGGCATTAA | nm |
| | nm |
| RNA_string_1 base_arn | nm |
| AAGAUGCCCGUAAGAUGCCCGUAAGAUGCCGU | nm |

```
codificar_sustitucion_guardado
Hell
curtis mayfield
```

```
csg
Resistire
Duo Dinamico
```

```
csg
DNA_string_2
GGGG
```

```
codificar_sustitucion
minusculas
esto es una clave
humanidad yo te amo porque estas todo el tiempo metiendo el secreto de la vida en tus pantalones
```

```
cs
mayusculas
CLAVE
YO LA AME Y ELLA NO ME AMO YO LE DI MI ALMA Y ELLA SE BURLO
```

```
decodificar_sustitucion
minusculas
esto es una clave
lleommv xmynbdeueqfscnvhtymervlsuxsvgnprpsbmogkmmzxmwfndwkteebtptidhwsz dmhxn dpktowdgtasecngrs
```

```
ds
mayusculas
CLAVE
Z FEBLMZD KEFPCKNIDOP VQOKYIDNP YMBXIUENXAUBBFLFEBCEUFWBLI
```

```
borra_alfabeto
mayusculas
```

```
ba
base_adn
```

```
borra_mensaje
DNA_string_1
```

```
borra_alfabeto
base_adn
```

```
la
```

```
lm
fin
```

Ejemplo de salida 1

```
#listar_alfabetos
1. abc
especial [a..c] 0 mensajes
2. base_adn
normal "GATC" 2 mensajes
3. base_arn
normal "ACGU" 1 mensajes
4. full_interv
especial [ ..~] 4 mensajes
5. full_no_interv
normal "ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz " 1 mensajes
#listar_mensajes
1. Assembler1
full_interv
"switch_to_pm: / cli / lgdt [gdt_descriptor] / mov eax,
2. Contrato_frances_1
full_interv
"Le contrat d'ouverture d'un livret A prevu a l'article
3. DNA_string_1
base_adn
"ATGGCATTAAATGGCATTAAATGGCATTAA"
4. DNA_string_2
base_adn
"CATTAAATGGCATTACATTAAATGGCATTAA"
5. Hell
minusculas
"dont worry if there is a hell below we are all going t
6. Qualsevelunit
full_interv
"Fa una nit clara i tranquila, hi ha la lluna que fa ll
7. RNA_string_1
base_arn
"AAGAUGCCGUAAGAUGCCGUAAGAUGCCGU"
8. Resistire
full_interv
"Cuando pierda todas las partidas / Cuando duerma con l
#nuevo_mensaje Contrato_frances_2 full_no_interv
9
#nuevo_mensaje RNA_string_2 base_arn
10
#nm RNA_string_2 base_arn
error: ya existe un mensaje con ese identificador
#nm acb acb
error: el alfabeto no existe
#nuevo_alfabeto bca
7
#na abc
error: el alfabeto ya existe
#borra_mensaje DNA_string_2
9
#bm DNA_string_2
error: el mensaje no existe
#codificar_sustitucion_guardado Hell "curtis mayfield"
"fhdlhnnncrveqjkwjyhxxh rlaxmmpwcdybgdrvq ywmdlontxgqeftl
#csg Resistire "Duo Dinamico"
"gkQn)YnRWOVT&udo)KbaZKW05Wbt.NPUmxc3:W^d4iSWS\QQDY_nDV
#csg DNA_string_2 "GGGG"
```

| | |
|---|---|
| error: el mensaje no existe | 6. minusculas |
| #codificar_sustitucion minusculas "esto es normal" | normal "abcdefghijklmnopqrstuvwxyz " 1 mensajes |
| "lleomnv xmynbdeueqfscnvhvtymervlsuxsvgndprbrrmoqkzmzxmwfrrndwkteehtptidhwsz dmhxn dpktowdgtasecngrr" | 1. Assembler1 |
| #cs mayusculas "CLAVE" | full_interv |
| error: el alfabeto no existe | eswhacklove: / cli / lgdt [gdt_descriptor] / mov eax, |
| #decodificar_sustitucion minusculas "esto es un secreto" | 2. Contrato_frances_1 |
| "humanidad yo te amo porque estas todo el tiempo metido en el secreto de la vida en tus pantalones" | full_interv |
| #ds mayusculas "CLAVE" | 3. Contrato_frances_2 |
| error: el alfabeto no existe | full_no_interv |
| #borra_alfabeto mayusculas | "Le contrat d'ouverture d'un livret A prevu a l'article |
| error: el alfabeto no existe | 4. Hell |
| #ba base_adn | minusculas |
| error: hay mensajes guardados con el alfabeto | "dont worry if there is a hell below we are all going to |
| #borra_mensaje DNA_string_1 | 5. Qualsevol_nit |
| 8 | full_interv |
| #borra_alfabeto base_adn | "Fa una nit clara i tranquila, hi ha la lluna que fa ll |
| 6 | 6. RNA_string_1 |
| #la | base_arn |
| 1. abc | "AAGAUGCCGUAAGAUGCCGUAAGAUGCCGU" |
| especial [a..c] 0 mensajes | 7. RNA_string_2 |
| 2. base_arn | base_arn |
| normal "ACGU" 2 mensajes | "CCGUAAGAUGCCCCGUAAGAUGCC" |
| 3. bca | 8. Resistire |
| normal "bca" 0 mensajes | full_interv |
| 4. full_interv | "Cuando pierda todas las partidas / Cuando duerma con l |
| especial [..~] 4 mensajes | normal "abcdefghijklmnopqrstuvwxyz{ }~ !"#\$%&'()*+,-./0123456789 |
| 5. full_no_interv | |
| normal "ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{ }~ !"#\$%&'()*+,-./0123456789 | |

Información del problema

Autoría: PRO2

Generación: 2026-01-25T21:15:43.526Z

© Jutge.org, 2006–2026.

<https://jutge.org>