

Argumentos en línea de comandos

1. Describa qué son, qué tipos las definen y qué contienen las variables `argc` y `argv` en la ejecución de un programa.
2. Dada la siguiente ejecución de un programa:

```
$ ./programa -s "HoLa Mundo!"
```

- a. ¿cuánto vale `argc`?
- b. Haga un diagrama que muestre el contenido de `argv`.

Argumentos posicionales

Atención

En los próximos ejercicios, la función de cada argumento estará dada por su ubicación en la invocación al programa.

1. Escriba un programa que imprima por pantalla todos los argumentos recibidos, uno por línea.
2. Escriba un programa que reciba una cadena de caracteres e imprima su longitud, a la `strlen()`.
3. Escriba un programa que reciba dos cadenas de caracteres e imprima el resultado de su comparación lexicográfica, como lo haría la función `strcmp()` o `strcasecmp()`.
4. Escriba un programa que reciba como argumentos dos números e imprima el resultado de la operación módulo entre ambos.
5. Escriba un programa que imprima de manera ordenada los argumentos recibidos:

```
$ ./g07e07 Ser o "no ser," "esa es" "la cuestión"
Ser
esa es
la cuestión
no ser,
o
```

Si el primer parámetro del programa es `-r`, entonces los ordena al revés:

```
$ ./g07e07 Ser o "no ser," "esa es" "la cuestión"
o
no ser,
la cuestión
esa es
Ser
```

6. El siguiente programa reemplazará caracteres, recibidos como argumentos, en líneas leídas por `stdin`. Se debe ejecutar del siguiente modo:

```
$ cat archivo.txt | ./reemplazar viejo nuevo
```

donde `viejo` y `nuevo` son los caracteres a reemplazar.

¿Cómo modificaría el programa para que pueda reemplazar más que un carácter, por ejemplo, reemplazar `x` por `equis`, o “American National Standards Institute” por “ANSI”, o “ISO” por “International Organization for Standardization”, etc.

7. El siguiente programa generará muestras aleatorias en un intervalo dado por el usuario. El programa debe recibir como argumentos el inicio del intervalo, x_i , el fin del intervalo, x_f , y la cantidad de muestras a generar, n , de forma tal que se ejecute de la siguiente manera:

Invocación

```
$ ./random x_i x_f n
```

```
$ ./g07e09 -3 2.65 10
2.0284
1.5767
0.4812
-1.1543
-0.7098
-0.1651
2.0144
0.2328
0.2302
1.3886
```

8. El siguiente programa procesará las líneas por `stdin` e imprimirá datos de las mismas. Escriba un programa que reciba como parámetros una *posición* y un *ancho* e imprima, de cada línea, los datos que comienzan en la posición *posición* y se componen de *ancho* caracteres. La ejecución del mismo es:

```
$ cat archivo.txt | ./extraer posición ancho
```

Este programa es de utilidad cuando se trabaja con datos almacenados en archivos formados por registros con campos de ancho fijo. El utilizar este tipo de formato permite

un procesamiento muy veloz de los datos, y archivos más pequeños que la gran mayoría de los formatos (sin compresión). Un ejemplo de un archivo de este estilo:

bibliografia.txt

```
KernighanRitchieThe C Programming LanguagePrentice-Hall19889780131103702
KernighanRitchieEl lenguaje de programacioPearson Educa19919789688802052
Aguilar ZahonerProgramacion en C: metodoMcGraw-Hill 20059788448198442
Deitel Deitel C: How to Program Pearson 20129780132990448
Bowman Algorithms and Data StructOxford Univer20049780195174809
Bowman HernándAlgoritmos y estructuras dOxford Univer19999789706134592
Kruse Tondo Data Structures and PrograPearson Educa20079788177584233
```

Los campos del documento son:

- **Autor 1:** posición: 1, ancho: 9
- **Autor 2:** posición: 10, ancho: 7
- **Título:** posición: 17, ancho: 26
- **Editorial:** posición: 43, ancho: 13
- **Año:** posición: 56, ancho: 4
- **ISBN:** posición: 60, ancho: 13

Ejemplo de ejecución

```
$ cat bibliografia.txt | ./g07e10 60 13
9780131103702
9789688802052
9788448198442
9780132990448
9780195174809
9789706134592
9788177584233
```

9. El siguiente programa trabajará con archivos con campos separados por delimitadores **DSV**. El más común de ellos es el que utiliza comas (,) como separadores, el **CSV**. El archivo anterior podría escribirse como:

bibliografia.csv

```
Kernighan,Ritchie,The C Programming Language,Prentice
Hall,1988,9780131103702
Kernighan,Ritchie,El lenguaje de programación C,Pearson
Educación,1991,9789688802052
Aguilar,Zahoner,Programación en C: metodología, algoritmos y estructuras
de datos,McGraw-Hill,2005,9788448198442
Deitel,Deitel.C: How to Program,Pearson,2012,9780132990448.
Bowman,,Algorithms and Data Structures: An Approach in C,Oxford University
Press,2004,9780195174809
Bowman,Hernández,Algoritmos y estructuras de datos: aproximación en
```

C,Oxford University Press,1999,9789706134592
Kruse,Tondo,Data Structures and Program Design in C,Pearson
Education,2007,9788177584233

En este caso, implemente un programa que reciba como argumento el número de campo e imprima el campo correspondiente. Los números de campo comienzan en 1, mientras que el 0 corresponde a la línea completa.

10. Escriba un programa `expr` que evalúe expresiones en [notación polaca inversa](#) recibidas como argumento. Por ejemplo:

```
$ ./expr 2 222 111 + "*"
666
```

evalúa la operación: $(111 + 222) \times 2$.

Argumentos no-posicionales

Atención

En los próximos ejercicios, la función de cada argumento estará dada por las banderas, *flags*, en la línea de comandos.

1. Modifique el programa del [ejercicio sobre randoms](#) para que reciba los argumentos utilizando las siguientes banderas:

- **-a** comienzo del intervalo
- **-b** fin del intervalo
- **-n** cantidad de muestras
- **-h** muestra un mensaje de ayuda, sin realizar ninguna otra función

Las banderas deben poder aparecer en cualquier orden, por lo que las siguientes invocaciones del programa son equivalentes:

```
$ ./g07e13 -a -3 -b 2.65 -n 10
$ ./g07e13 -b 2.65 -a -3 -n 10
$ ./g07e13 -n 10 -b 2.65 -a -3
2.0422
1.6900
-1.0428
-0.3802
-1.7582
-1.3972
-1.5386
-0.0568
-2.3125
```

```

-2.6120
$ ./g07e13 -h
NOMBRE
    g07e13, sampler - genera lotes de números pseudo-aleatorios

SINOPSIS
    g07e13 [-a A] [-b B] [-n N] [-h]

DESCRIPCIÓN
    Generar un lote de N números pseudo-aleatorios en el intervalo
    [A,B). La distribución muestreada es U[A,B).

OPCIONES
    h
        muestra esta ayuda.
    a A
        comienzo del intervalo. Valor por omisión: 0.0.
    b B
        fin del intervalo. Valor por omisión: 1.0.
    n N
        cantidad de muestras a tomar. Valor por omisión: 100.

AUTORES
    Ada Lovelace, Alan Turing

LICENCIA
    GNU General Public License v3.0

```

2. Modifique el programa extractor de campos de ancho fijo ([ver aquí](#)) de forma tal que los argumentos del mismo se puedan recibir en cualquier orden. Para ello, agregue las siguientes banderas:

- **-p** posición del campo
- **-a** ancho del campo
- **-h** muestra un mensaje de ayuda, sin realizar ninguna otra función

Ejemplo de ejecución

```

$ cat bibliografia.txt | ./g07e14 -p 60 -a 13
$ cat bibliografia.txt | ./g07e14 -a 13 -p 60
9780131103702
9789688802052
9788448198442
9780132990448
9780195174809
9789706134592
9788177584233
$ ./g07e14 -h
NOMBRE
    g07e14, extractor - extrae campos de ancho fijo

SINOPSIS
    g07e14 -p POSICIÓN -a ANCHO [-h]

```

DESCRIPCIÓN

Lee líneas de stdin y extrae porciones de texto de cada una de estas.

OPCIONES

h
muestra esta ayuda.
p POSICIÓN
posición en la que comienza un campo
a ANCHO
ancho del campo a extraer

AUTORES

Steve Jobs, nah, mentira, "nunca" escribió código en su vida.

LICENCIA

GNU General Public License v3.0

3. Sea la siguiente forma de onda

$$f(t) = e^{-\alpha t} A \cos(2\pi ft)$$

Escribir un programa con comandos en línea de órdenes que permita tabular la función dentro del intervalo [T1,T2) utilizando N puntos, aceptando la siguiente invocación:

```
$ sampler -alfa <a> -amp <A> -frec <f> -start <T1> -end <T2> -points <N>
```

Realizar las validaciones pertinentes.

Ejemplo de ejecución

```
$ ./g07e15 -amp 3 -frec .6 -end 6 -start 0 -alfa 0.5 -points 10
0      3
0.6    -1.41665
1.2    -0.308511
1.8    1.06884
2.4    -0.84013
3      0.206853
3.6    0.265715
4.2    -0.364472
4.8    0.198392
5.4    0.0126596
```

```
$ ./g07e15 -h
```

NOMBRE

g07e15, sampler - muestrea una senoidal con decaimiento exponencial

SINOPSIS

g07e15 -alfa a -amp A -frec f -start T1 -end T2 -points N

DESCRIPCIÓN

Dada la función:

$-at$

$$f(t) = e^{-at} A \cos(2\pi f t)$$

toma N muestras a intervalos de tiempo regulares entre T1 y T2. El span del intervalo entre muestras es (T2 - T1)/N

OPCIONES

h

muestra esta ayuda.

alfa a

tasa de decaimiento

amp A

amplitud de la función en tiempo 0

frec f

frecuencia del oscilador

start T1

valor inicial del intervalo

end T2

valor final del intervalo

points N

cantidad de puntos a imprimir

AUTORES

Bill Gatos, nah, mentira, en un mundo sin fronteras ¿para qué queremos puertasy ventanas?.

LICENCIA

GNU General Public License v3.0