



Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

Explicación de la práctica 2

Entrada/Salida y preprocesador

Seminario de Lenguajes opción C

Facultad de Informática
Universidad Nacional de La Plata

2016



Indice

Explicación de
la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-
escritura de
caracteres
Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

① Entrada/Salida

Definiciones

stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

② Preprocesador

Conceptos

Stringification y concatenación

¿Preguntas?



Entrada/Salida: Definiciones

Explicación de la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones

stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- **Periféricos:** Unidades de almacenamiento y dispositivos que permiten comunicación con el exterior.
- **Periféricos de entrada:** Permiten leer datos.
 - Teclado, mouse, etc...
 - Webcam.
- **Periféricos de salida:** Permiten escribir datos.
 - Monitor.
 - Impresora.
- **Periféricos de entrada/salida:** Permiten ambas operaciones.
 - Disco, pendrive, etc...
 - Placa de red, placa de sonido, etc...



Entrada/Salida: Definiciones

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones

stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

- **Periféricos:** Unidades de almacenamiento y dispositivos que permiten comunicación con el exterior.
- **Periféricos de entrada:** Permiten leer datos.
 - Teclado, mouse, etc...
 - Webcam.
- **Periféricos de salida:** Permiten escribir datos.
 - Monitor.
 - Impresora.
- **Periféricos de entrada/salida:** Permiten ambas operaciones.
 - Disco, pendrive, etc...
 - Placa de red, placa de sonido, etc...



Entrada/Salida: Definiciones

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones

stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos

Stringification y concatenación

¿Preguntas?

- **Periféricos:** Unidades de almacenamiento y dispositivos que permiten comunicación con el exterior.
- **Periféricos de entrada:** Permiten leer datos.
 - Teclado, mouse, etc...
 - Webcam.
- **Periféricos de salida:** Permiten escribir datos.
 - Monitor.
 - Impresora.
- **Periféricos de entrada/salida:** Permiten ambas operaciones.
 - Disco, pendrive, etc...
 - Placa de red, placa de sonido, etc...



Entrada/Salida: Definiciones

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones

stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos

Stringification y concatenación

¿Preguntas?

- **Periféricos:** Unidades de almacenamiento y dispositivos que permiten comunicación con el exterior.
- **Periféricos de entrada:** Permiten leer datos.
 - Teclado, mouse, etc...
 - Webcam.
- **Periféricos de salida:** Permiten escribir datos.
 - Monitor.
 - Impresora.
- **Periféricos de entrada/salida:** Permiten ambas operaciones.
 - Disco, pendrive, etc...
 - Placa de red, placa de sonido, etc...



stdin, stdout, stderr

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones **stdin**, **stdout**, **stderr**

Lectura- escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos Stringification y concatenación ¿Preguntas?

Los procesos (programas en ejecución) normalmente cuentan con 3 “archivos” abiertos.

- **stdin**: “Entrada estándar”, normalmente el teclado.
- **stdout**: “Salida estándar”, normalmente el monitor.
- **stderr**: “Error estándar”, normalmente la salida estándar.

Están definidos en `stdio.h`

```
#include <stdio.h>
FILE *stdin;
FILE *stdout;
FILE *stderr;
```



stdin, stdout, stderr

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, **stdout**,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

Los procesos (programas en ejecución) normalmente cuentan con 3 “archivos” abiertos.

- **stdin**: “Entrada estándar”, normalmente el teclado.
- **stdout**: “Salida estándar”, normalmente el monitor.
- **stderr**: “Error estándar”, normalmente la salida estándar.

Están definidos en `stdio.h`

```
#include <stdio.h>
FILE *stdin;
FILE *stdout;
FILE *stderr;
```




stdin, stdout, stderr

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos Stringification y concatenación ¿Preguntas?

Los procesos (programas en ejecución) normalmente cuentan con 3 “archivos” abiertos.

- **stdin:** “Entrada estándar”, normalmente el teclado.
- **stdout:** “Salida estándar”, normalmente el monitor.
- **stderr:** “Error estándar”, normalmente la salida estándar.

Están definidos en `stdio.h`

```
#include <stdio.h>
FILE *stdin;
FILE *stdout;
FILE *stderr;
```



stdin, stdout, stderr

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos Stringification y concatenación ¿Preguntas?

Los procesos (programas en ejecución) normalmente cuentan con 3 “archivos” abiertos.

- **stdin:** “Entrada estándar”, normalmente el teclado.
- **stdout:** “Salida estándar”, normalmente el monitor.
- **stderr:** “Error estándar”, normalmente la salida estándar.

Están definidos en `stdio.h`

```
#include <stdio.h>
FILE *stdin;
FILE *stdout;
FILE *stderr;
```



stdin, stdout, stderr

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, **stdout**,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

Los procesos (programas en ejecución) normalmente cuentan con 3 “archivos” abiertos.

- **stdin**: “Entrada estándar”, normalmente el teclado.
- **stdout**: “Salida estándar”, normalmente el monitor.
- **stderr**: “Error estándar”, normalmente la salida estándar.

Están definidos en `stdio.h`

```
#include <stdio.h>
FILE *stdin;
FILE *stdout;
FILE *stderr;
```



Explicación de la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
stdin, **stdout**,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Macro definida en `stdio.h`.
- Significado.
- Uso:
 - Con archivos “normales”.
 - Con la terminal y `stdin`.
 - Combinaciones de teclas en GNU/Linux y Windows.
- Es algún valor fuera del rango del tipo `unsigned char`.
- Se puede representar en una variable de tipo `int`.



Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, **stdout**,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Macro definida en `stdio.h`.
- Significado.
- Uso:
 - Con archivos “normales”.
 - Con la terminal y `stdin`.
 - Combinaciones de teclas en GNU/Linux y Windows.
- Es algún valor fuera del rango del tipo `unsigned char`.
- Se puede representar en una variable de tipo `int`.



Explicación de la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
stdin, **stdout**,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringificación y
concatenación
¿Preguntas?

- Macro definida en `stdio.h`.
- Significado.
- Uso:
 - Con archivos “normales”.
 - Con la terminal y `stdin`.
 - Combinaciones de teclas en GNU/Linux y Windows.
- Es algún valor fuera del rango del tipo `unsigned char`.
- Se puede representar en una variable de tipo `int`.



EOF

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, **stdout**,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Macro definida en `stdio.h`.
- Significado.
- Uso:
 - Con archivos “normales”.
 - Con la terminal y `stdin`.
 - Combinaciones de teclas en GNU/Linux y Windows.
- Es algún valor fuera del rango del tipo `unsigned char`.
- Se puede representar en una variable de tipo `int`.



EOF

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
`stdin`, `stdout`,
`stderr`

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Macro definida en `stdio.h`.
- Significado.
- Uso:
 - Con archivos “normales”.
 - Con la terminal y `stdin`.
 - Combinaciones de teclas en GNU/Linux y Windows.
- Es algún valor fuera del rango del tipo `unsigned char`.
- Se puede representar en una variable de tipo `int`.



Explicación de la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
`stdin`, `stdout`,
`stderr`

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Macro definida en `stdio.h`.
- Significado.
- Uso:
 - Con archivos “normales”.
 - Con la terminal y `stdin`.
 - Combinaciones de teclas en GNU/Linux y Windows.
- Es algún valor fuera del rango del tipo `unsigned char`.
- Se puede representar en una variable de tipo `int`.



Explicación de la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
`stdin`, `stdout`,
`stderr`

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Macro definida en `stdio.h`.
- Significado.
- Uso:
 - Con archivos “normales”.
 - Con la terminal y `stdin`.
 - Combinaciones de teclas en GNU/Linux y Windows.
- Es algún valor fuera del rango del tipo `unsigned char`.
- Se puede representar en una variable de tipo `int`.



Lectura-escritura de caracteres

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

Entrada:

- **int getchar():** Lee 1 carácter desde stdin, retorna un int (para no truncar el valor de EOF).
- **int getc(FILE *stream):** Idem getchar(), pero lee desde cualquier archivo.

`getchar() = getc(stdin)`

- **char *fgets(char *s, int tam, FILE *stream):** Lee hasta el fin de línea o de archivo, s debe estar alocado. Lee a lo sumo tam - 1 caracteres.

`man fgets`

Salida:

- **int putchar(int c)**
- **int putc(int c, FILE *stream)**
- **int puts(const char *s)**

Valores de retorno



Lectura-escritura de caracteres

Explicación de
la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

Entrada:

- **int getchar():** Lee 1 carácter desde stdin, retorna un int (para no truncar el valor de EOF).
- **int getc(FILE *stream):** Idem getchar(), pero lee desde cualquier archivo.

`getchar() = getc(stdin)`

- **char *fgets(char *s, int tam, FILE *stream):** Lee hasta el fin de línea o de archivo, s debe estar alocado. Lee a lo sumo tam - 1 caracteres.

`man fgets`

Salida:

- `int putchar(int c)`
- `int putc(int c, FILE *stream)`
- `int puts(const char *s)`

Valores de retorno



Lectura-escritura de caracteres

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

Entrada:

- **int getchar():** Lee 1 carácter desde stdin, retorna un int (para no truncar el valor de EOF).
- **int getc(FILE *stream):** Idem getchar(), pero lee desde cualquier archivo.

`getchar() = getc(stdin)`

- **char *fgets(char *s, int tam, FILE *stream):** Lee hasta el fin de línea o de archivo, s debe estar alocado. Lee a lo sumo tam - 1 caracteres.

`man fgets`

Salida:

- `int putchar(int c)`
- `int putc(int c, FILE *stream)`
- `int puts(const char *s)`

Valores de retorno



Lectura-escritura de caracteres

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

Entrada:

- **int getchar():** Lee 1 carácter desde stdin, retorna un int (para no truncar el valor de EOF).
- **int getc(FILE *stream):** Idem getchar(), pero lee desde cualquier archivo.

`getchar() = getc(stdin)`

- **char *fgets(char *s, int tam, FILE *stream):** Lee hasta el fin de línea o de archivo, s debe estar alocado. Lee a lo sumo tam - 1 caracteres.

`man fgets`

Salida:

- `int putchar(int c)`
- `int putc(int c, FILE *stream)`
- `int puts(const char *s)`

Valores de retorno



Caracteres especiales

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- `'\n'`: LF
- `'\r'`: CR
- `'\t'`: TAB
- `'\\'`: \
- `'\0'`: Un byte en cero (distinto del char '0').

El fin de línea varía en cada sistema operativo:

- **GNU/Linux, *BSD y Mac OS X:** `\n`
- **DOS y Windows:** usan secuencias de `\r\n`
- Otras combinaciones...



Caracteres especiales

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- `'\n'`: LF
- `'\r'`: CR
- `'\t'`: TAB
- `'\\'`: \
- `'\0'`: Un byte en cero (distinto del char '0').

El fin de línea varía en cada sistema operativo:

- **GNU/Linux, *BSD y Mac OS X:** `\n`
- **DOS y Windows:** usan secuencias de `\r\n`
- Otras combinaciones...



Caracteres especiales

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringificación y
concatenación
¿Preguntas?

- `'\n'`: LF
- `'\r'`: CR
- `'\t'`: TAB
- `'\\'`: \
- `'\0'`: Un byte en cero (distinto del char '0').

El fin de línea varía en cada sistema operativo:

- **GNU/Linux, *BSD y Mac OS X:** `\n`
- **DOS y Windows:** usan secuencias de `\r\n`
- Otras combinaciones...



Caracteres especiales

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringificación y
concatenación
¿Preguntas?

- `'\n'`: LF
- `'\r'`: CR
- `'\t'`: TAB
- `'\\'`: \
- `'\0'`: Un byte en cero (distinto del char '0').

El fin de línea varía en cada sistema operativo:

- **GNU/Linux, *BSD y Mac OS X:** `\n`
- **DOS y Windows:** usan secuencias de `\r\n`
- Otras combinaciones...



Salida con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

- Familia de funciones `printf`, `fprintf`, `sprintf`, etc..
- Permiten escribir variables que no sean `char`.
- Se definen como: `printf(formato, var1, ...)`.
- *formato* es un string que puede tener:
 - Caracteres que se imprimirán tal cual aparecen.
 - Especificadores de conversión que indican como imprimir `var1`, `var2`, etc...

Ejemplos

```
double x = 2 / 3;  
printf("%d", 24);  
printf("El resultado es %d\n", 10);  
printf("2/3 = %f\n", x);
```



Salida con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

**Salida con
formato**

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `printf`, `fprintf`, `sprintf`, etc..
- Permiten escribir variables que no sean `char`.
- Se definen como: `printf(formato, var1, ...)`.
- *formato* es un string que puede tener:
 - Caracteres que se imprimirán tal cual aparecen.
 - Especificadores de conversión que indican como imprimir `var1`, `var2`, etc...

Ejemplos

```
double x = 2 / 3;  
printf("%d", 24);  
printf("El resultado es %d\n", 10);  
printf("2/3 = %f\n", x);
```



Salida con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones

stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos

Stringification y concatenación

¿Preguntas?

- Familia de funciones `printf`, `fprintf`, `sprintf`, etc..
- Permiten escribir variables que no sean `char`.
- Se definen como: `printf(formato, var1, ...)`.
- *formato* es un string que puede tener:
 - Caracteres que se imprimirán tal cual aparecen.
 - Especificadores de conversión que indican como imprimir `var1`, `var2`, etc...

Ejemplos

```
double x = 2 / 3;
printf("%d", 24);
printf("El resultado es %d\n", 10);
printf("2/3 = %f\n", x);
```



Salida con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones printf, fprintf, sprintf, etc..
- Permiten escribir variables que no sean char.
- Se definen como: printf(formato, var1, ...).
- *formato* es un string que puede tener:
 - Caracteres que se imprimirán tal cual aparecen.
 - Especificadores de conversión que indican como imprimir var1, var2, etc...

Ejemplos

```
double x = 2 / 3;  
printf("%d", 24);  
printf("El resultado es %d\n", 10);  
printf("2/3 = %f\n", x);
```



Salida con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

- Familia de funciones `printf`, `fprintf`, `sprintf`, etc..
- Permiten escribir variables que no sean `char`.
- Se definen como: `printf(formato, var1, ...)`.
- *formato* es un string que puede tener:
 - Caracteres que se imprimirán tal cual aparecen.
 - Especificadores de conversión que indican como imprimir `var1`, `var2`, etc...

Ejemplos

```
double x = 2 / 3;  
printf("%d", 24);  
printf("El resultado es %d\n", 10);  
printf("2/3 = %f\n", x);
```



Salida con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones printf, fprintf, sprintf, etc..
- Permiten escribir variables que no sean char.
- Se definen como: printf(formato, var1, ...).
- *formato* es un string que puede tener:
 - Caracteres que se imprimirán tal cual aparecen.
 - Especificadores de conversión que indican como imprimir var1, var2, etc...

Ejemplos

```
double x = 2 / 3;  
printf("%d", 24);  
printf("El resultado es %d\n", 10);  
printf("2/3 = %f\n", x);
```




Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - **-** → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - - → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - **-** → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringificación y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - **-** → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - - → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - **-** → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - **-** → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Especificadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Comienzan con % y terminan con un indicador de conversión (por ejemplo %d).
- Entre el % y el indicador de conversión pueden haber 0 o más opciones en el siguiente orden:
 - **0** → Rellenar con ceros.
 - **-** → Alineación a izquierda.
 - **anchura de campo** → Si el valor convertido tiene menos caracteres se rellena con blancos.
 - **.precisión**
 - Mínimo de dígitos al convertir números enteros.
 - Máximo de dígitos después del punto para variables de punto flotante.
 - Máximo de caracteres para los strings.
 - **indicador de longitud** → por ejemplo para imprimir un long int: %ld



Indicadores de conversión

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

Los caracteres de conversión son:

- **i,d**: entero
- **o**: octal
- **X, x**: hexadecimal
- **u**: unsigned
- **c**: char
- **s**: string
- **f**: double
- **e**: ídem f pero con notación exponencial

Más que nunca, aprovechar las manpages: `man 3 printf`



Entrada con formato

Explicación de
la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres
Salida con
formato

**Entrada con
formato**

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
- `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
- Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Entrada con formato

Explicación de
la práctica 2

Seminario de
Lenguajes
opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

**Entrada con
formato**

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
- `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
- Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Entrada con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres
Salida con formato

Entrada con formato

Preprocesador

Conceptos
Stringification y concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
 - `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
 - Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Entrada con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres
Salida con
formato

**Entrada con
formato**

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
- `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
- Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Entrada con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
- `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
- Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Entrada con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
- `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
- Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Entrada con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
- `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
- Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Entrada con formato

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Familia de funciones `scanf()`.
- Formato similar a `printf()`.
- Definición `scanf(formato, ptr1, ...)`
- `ptr1, ptr2, etc...` deberán ser punteros alocados donde se cargarán los datos.
- Retorna:
 - EOF si se alcanzó el fin de archivo antes de leer.
 - La cantidad de datos leídos correctamente (eventualmente 0).
- Mantiene un buffer, si falla al leer un dato, limpiar con `getchar()`.
- Ver ejercicio 11 de la práctica 2.

Más que nunca, aprovechar las manpages: `man 3 scanf`



Preprocesador: Conceptos

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- ¿Qué es?
- ¿Cómo trabaja?
- `#include`
- `gcc -E`
- `#define`, `#ifdef`, `#ifndef`, `#endif`, `#else`, `#undef`
- Definirlas con `gcc -D`
- Macros:
Ejemplo `min(x,y)`. Ver efectos laterales de `min(++x,++y)`



Preprocesador: Conceptos

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- ¿Qué es?
- ¿Cómo trabaja?
- `#include`
- `gcc -E`
- `#define`, `#ifdef`, `#ifndef`, `#endif`, `#else`, `#undef`
- Definirlas con `gcc -D`
- Macros:
Ejemplo `min(x,y)`. Ver efectos laterales de `min(++x,++y)`



Preprocesador: Conceptos

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- ¿Qué es?
- ¿Cómo trabaja?
- `#include`
 - `gcc -E`
 - `#define`, `#ifdef`, `#ifndef`, `#endif`, `#else`, `#undef`
 - Definirlas con `gcc -D`
- Macros:
Ejemplo `min(x,y)`. Ver efectos laterales de `min(++x,++y)`



Preprocesador: Conceptos

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- ¿Qué es?
- ¿Cómo trabaja?
- `#include`
- `gcc -E`
 - `#define`, `#ifdef`, `#ifndef`, `#endif`, `#else`, `#undef`
 - Definirlas con `gcc -D`
 - Macros:
Ejemplo `min(x,y)`. Ver efectos laterales de `min(++x,++y)`



Preprocesador: Conceptos

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- ¿Qué es?
- ¿Cómo trabaja?
- `#include`
- `gcc -E`
- `#define`, `#ifdef`, `#ifndef`, `#endif`, `#else`, `#undef`
- Definirlas con `gcc -D`
- Macros:
Ejemplo `min(x,y)`. Ver efectos laterales de `min(++x,++y)`



Preprocesador: Conceptos

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- ¿Qué es?
- ¿Cómo trabaja?
- `#include`
- `gcc -E`
- `#define`, `#ifdef`, `#ifndef`, `#endif`, `#else`, `#undef`
- Definirlas con `gcc -D`
- Macros:
Ejemplo `min(x,y)`. Ver efectos laterales de `min(++x,++y)`



Preprocesador: Conceptos

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- ¿Qué es?
- ¿Cómo trabaja?
- `#include`
- `gcc -E`
- `#define`, `#ifdef`, `#ifndef`, `#endif`, `#else`, `#undef`
- Definirlas con `gcc -D`
- Macros:
Ejemplo `min(x,y)`. Ver efectos laterales de `min(++x,++y)`



Stringification y concatenación

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos

Stringification y
concatenación

¿Preguntas?

- Se puede convertir en string el argumento de una macro con #:

```
#define x2string(x) #x
int x = 5;
x2string(2 + 5); // Se reemplaza por "2 + 5"
x2string(x); // Se reemplaza por "x"
```

- Se pueden concatenar tokens con ##:

```
#define concat(x, y) x##y
concat(p, uts)("Hola mundo");
printf("%d\n", concat(2, 3)); // Se reemplaza
por 23
```



Stringification y concatenación

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación
¿Preguntas?

- Se puede convertir en string el argumento de una macro con #:

```
#define x2string(x) #x
int x = 5;
x2string(2 + 5); // Se reemplaza por "2 + 5"
x2string(x); // Se reemplaza por "x"
```

- Se pueden concatenar tokens con ##:

```
#define concat(x, y) x##y
concat(p, uts)("Hola mundo");
printf("%d\n", concat(2, 3)); // Se reemplaza
por 23
```



Sobre el compilador y los strings

Explicación de la práctica 2

Seminario de Lenguajes opción C

Entrada/Salida

Definiciones
stdin, stdout, stderr

Lectura-escritura de caracteres

Salida con formato

Entrada con formato

Preprocesador

Conceptos

Stringification y concatenación

¿Preguntas?

- Si se escriben 2 strings contiguos, se unen para formar uno solo:

```
printf("Hola mundo\n");  
// Lo anterior es lo mismo que:  
printf("Hola ""mundo\n");  
// Y es lo mismo que  
printf(x2string(Hola ) " " x2string(mundo\n));
```



Explicación de la práctica 2

Seminario de Lenguajes
opción C

Entrada/Salida

Definiciones
stdin, stdout,
stderr

Lectura-
escritura de
caracteres

Salida con
formato

Entrada con
formato

Preprocesador

Conceptos
Stringification y
concatenación

¿Preguntas?

¿Preguntas?