

**TPN°3: Control de Flujo****MUY IMPORTANTE**

Dado que uno de los objetivos importantes de esta materia es la realización de algoritmos correctos y eficientes, desarrollados con buen estilo de programación, en muchos ejercicios de esta guía se ha incluido, a modo de heurística, la cantidad tope de líneas de código en que deberían ser implementados.

Consideramos como línea de código a cualquier renglón del mismo que cumpla las normas de estandarización Indian Hill (que se desarrollarán a lo largo del curso) y que no sea ni llave, ni comentario, ni línea de declaración o definición de variables.

Para resolver esta práctica no usar el operador condicional, sino *if-else*.

**Ejercicio 1**

Leer un caracter desde la entrada estándar e imprimir por salida estándar si el caracter es una letra o no y, en caso de ser una letra, indicar si es letra mayúscula o minúscula, sin tener en cuenta el caracter 'ñ'. (En no más de 15 líneas).

### Ejercicio 2

Leer dos caracteres desde la entrada estándar e imprimir por salida estándar si son iguales, si el primero es mayor o si el primero es menor, teniendo en cuenta su valor ASCII. (En no más de 15 líneas).

Ejemplo: si se ingresa 'a', 'g', debe imprimir **El caracter 'a' es menor al caracter 'g'**.

### Ejercicio 3

Una empresa paga a su personal de venta en base a comisiones. El personal de ventas recibe \$300 por semana más un porcentaje de las ventas brutas de esa semana. El porcentaje a aplicar depende del total vendido: si la venta no supera los \$1000, no hay porcentaje, hasta \$2000 es un 5%, hasta \$4000 es un 7%, y más de \$4000 es un 9%.

Por ejemplo, un vendedor que vende \$5000 en una semana, recibe \$300 más 9% de \$5000, o sea un total de \$750.

Hacer un programa en C que lea una única variable que indique la suma vendida en una semana, e imprima por salida estándar el monto total a cobrar. Si la suma vendida es un valor incorrecto pedir un nuevo valor. (En no más de 15 líneas).

### Ejercicio 4

Problema del *else colgante*. Determine la salida estándar para cada uno de los siguientes casos, cuando *x* es 9, *y* es 11. Advierta que el compilador ignora las indentaciones. El compilador siempre asocia un **else** con el **if** anterior, a menos que se le indique lo contrario con llaves {}. A primera vista el programador tal vez no esté seguro de cuál if coincide con cuál else, esto se conoce como el problema del "else colgante". Se eliminaron las sangrías para hacer el problema un poco más 'interesante'.

Determinar la salida primero haciendo el seguimiento y luego comprobándola con la PC.

```
a)    if (x < 10)
        if (y > 10)
            printf("#####\n");
        else
            printf("%%%%%\n");
        printf("*****\n");
```

```
b)    if (x < 10) {
        if (y > 10)
            printf("#####\n");
        }
        else {
            printf("%%%%%\n");
            printf("*****\n");
        }
```

- c) Corregir el siguiente fragmento de código (agregando o quitando llaves) para que el cartel impreso coincida con los valores de cada variable.

```
if ( x < 3 )
printf("x < 3");
if ( z < 5 )
    printf("x < 3 y z < 5");
else
printf("x >= 3");
```

- d) Idem ejercicio anterior

```
if ( x == 1)
if ( z > 1)
    printf("x == 1 y z > 1");
else
if ( x > 1)
    printf("x > 1");
else
    printf("x < 1");
```

### Ejercicio 5

Completar las condiciones para que el siguiente fragmento de código imprima el texto correcto (considerar solo el alfabeto inglés):

```
int a, b, c;
a = getchar();
b = getchar();
c = getchar();

if (.....)
    printf("El caracter %c es una consonante mayúscula \n", a);
else
    if (.....)
        printf("El caracter %c es una consonante minúscula \n", a);
    else
        if (.....)
            printf("El caracter %c es una vocal \n", a);
        else
            printf("El caracter %c no es una letra \n", a);

if ( ..... )
    printf("El valor ASCII de %c es par \n", b);

if ( ..... )
    printf("El caracter %c representa un dígito par \n", b);

if ( ..... )
    printf("El caracter %c es una vocal o es mayúscula \n", c);

if ( ..... )
    printf("El caracter %c no es una vocal pero es letra \n", c);
```

**Ejercicio 6**

Identificar y corregir los errores en los siguientes fragmentos de programas. Resolverlos primero en papel, y luego verificarlos con el debugger **gdb**.

- a) 

```
int a=0;
while ( a <= 4)
    resultado += a;
    a++;
```
- b) 

```
if ( sueldo >= 1000),
    printf ( "El sueldo es mayor o igual a 1.000\n"),
else
    printf ("El sueldo es menor a 1.000\n");
```
- c) 

```
int var1= 1, var2;
while (var1 <= 8)
{
    var2 += var1;
    ++ var1;
}
```
- d) 

```
while ('Y' != (c= getchar())) ;
    printf ("Pulse Y para abandonar este ciclo \n");
```
- e) 

```
char c;
while((c = getchar()) != EOF)
    putchar(c);
```
- f) Dada la siguiente definición:
- ```
enum dias { LUN, MAR, MIE, JUE, VIE, SAB,DOM};
```

corregir el siguiente fragmento de código para imprimir un cartel con el nombre del día

```
enum dias dia = LUN;
printf("Hoy es %d", dia);
```

**Ejercicio 7**

Decir qué envía el siguiente programa a la salida estándar.

```
#include <stdio.h>
#include "getnum.h"

int
main(void)
{
    int i, j, n = 0, suma = 0;

    while ( n <= 0 )
        n = getint("Ingrese cantidad de elementos:");

    for (i=1; i<=n; i++)
    {
        j = getint("Ingrese valor %d:", i);
        suma += j;
    }

    printf("El resultado es : %.2f\n", suma / (float) j);
    return 0;
}
```

**Ejercicio 8**

Realizar los siguientes ejercicios del libro de texto básico (Los tres primeros en no más de 15 líneas cada uno):

**1.3, 1.4, 1.5, 2.1 y 2.2**

**Ejercicio 9**

Realizar los siguientes ejercicios del libro de texto básico (Los dos primeros en no más de 20 líneas cada uno):

**1.8, 1.12 y 1.11**

**Ejercicio 10**

Identificar y corregir el error en cada uno de los segmentos de código

- a) 

```
float x = 1/3;
while (x != 0.52 )
    x+=.01;
```
- b) 

```
for ( y = .1; y != 1.0; y+= .1)
    printf ("%f\n", y);
```

```
c)  int n= ... ;

    switch (n)
    {
        case 1:  printf("El número es 1\n");
        case 2:  printf("El número es 2\n");
                break;
        default: printf("El número no es ni 1 ni 2\n");
                break;
    }
```

d) El siguiente código lee caracteres e imprime el mayor de ellos, en base a su valor ASCII

```
int c,mayor;
mayor = 0;
while ( c=getchar() != EOF )
{
    if ( c > mayor )
        mayor = c;
}
printf ("El mayor es %c\n",mayor);
```

### Ejercicio 11

Escribir un programa que utilice ciclos para imprimir la siguiente tabla de valores, para N desde 1 hasta 20. (En no más de 6 líneas)

| N     | 10*N | 100*N | 1000*N |
|-------|------|-------|--------|
| 1     | 10   | 100   | 1000   |
| 2     | 20   | 200   | 2000   |
| ..... |      |       |        |
| 20    | 200  | 2000  | 20000  |

### Ejercicio 12

Escriba enunciados **for** que impriman las siguientes secuencias de valores:

- a) 3,8,13,18,23
- b) 20,14,8,2,-4,-10
- c) 19,27,35,43,51

**Ejercicio 13**

Dada una variable entera que indica el lado de un cuadrado, imprimir por salida estándar dicho cuadrado en forma de asteriscos.

- a) Resolverlo con dos ciclos anidados.
- b) Resolverlo con un único ciclo *for*.

*Ejemplo:* si el lado es 5, debe imprimir lo siguiente:

```
*****
*****
*****
*****
*****
```

**Ejercicio 14**

Leer desde la entrada estándar una serie de caracteres, hasta que uno de ellos no sea una letra ni un espacio. Luego imprimir por salida estándar el menor alfabéticamente de ellos respetando si fue mayúscula o minúscula, indicando además cuántas letras eran mayúsculas y cuántas minúsculas. No considerar el carácter 'ñ' como letra. (En no más de 15 líneas).

*Ejemplo:* si se ingresa "Aqui murio Solis" debe imprimir la letra "A"  
si se ingresa "Hola mundo" debe imprimir la letra "a"

**Ejercicio 15**

Dado un entero que representa un número binario positivo (todos sus dígitos son cero o uno), imprimir por salida estándar el equivalente decimal. (En no más de 10 líneas)

**Ejercicio 16**

Dado un número entero positivo, imprimir por salida estándar el equivalente en base binaria. El rango de números a considerar puede estar acotado. (En no más de 15 líneas).

**Ejercicio 17**

Escriba y ejecute un programa que imprima todos los múltiplos de 10 ( 10, 100, 1000, etc.) en no más de 5 líneas ¿Qué sucede cuando ejecuta este programa ?

**Ejercicio 18**

Escriba un programa que lea un entero no negativo ( cero o mayor ) y que calcule e imprima su factorial. En caso de ingresar un valor inválido, imprimir un cartel aclarando cuáles son los valores válidos. (En no más de 10 líneas)

**Ejercicio 19**

Leer un número entero e indicar cuantos dígitos de ese número son iguales a 5. Repetir para el caso de ser un número real, indicando cuántos 5 tiene en su parte entera y cuántos en su parte decimal. (En no más de 10 líneas el primer caso, y no más de 25 el segundo caso)

**Ejercicio 20**

Escribir un programa (en no más de 10 líneas) que calcule el valor de la constante matemática  $e$  utilizando la fórmula  $e = 1 + 1/1! + 1/2! + 1/3! + \dots$  e imprima una tabla que indique cantidad de términos de la serie y el resultado parcial, con el siguiente formato:

| N | e   |
|---|-----|
| 1 | 1   |
| 2 | 2   |
| 3 | 2.5 |

¿Qué criterio utilizaría para detener la secuencia? ( No usar cantidad fija de términos ni preguntarle al usuario la cantidad de términos. Tampoco tendría sentido comparar con el verdadero valor de  $e$ ).

**Ejercicio 21**

Escribir un programa que lea un número entero, y escriba por salida estándar dicho número y un cartel indicando si es capicúa o no. (En no más de 10 líneas)

**Ejercicio 22**

Escriba un programa que imprima una tabla de todos los equivalentes de números romanos con números decimales del 1 al 100. (En no más de 50 líneas)

**Ejercicio 23**

Realizar los siguientes ejercicios del libro de texto básico (en no más de 15 líneas cada uno):

**1.9 (con máquina de estados) y 1.10**



**Ejercicio 24**

Completar las condiciones de control de los siguientes ciclos:

```
int n,c;

/* al salir del ciclo n debe ser un número par mayor que cero */
do
    n=getint("");
while(.....);

/* este ciclo debe incluir la lectura de caracteres y debe ejecutarse
 * mientras no sea fin de archivo y los caracteres sean dígitos o
 * minúsculas
 */
while (.....)
    putchar(c);
```

**Ejercicio 25 MUY IMPORTANTE**

Dado el siguiente código explicar DETALLADAMENTE el resultado obtenido al ejecutarlo.

```
#include <stdio.h>

int
main(void)
{
    float a=0.1;

    if (a==0.1)
        printf("SON iguales\n");
    else
        printf("NO SON iguales: a vale %g que no es igual a 0.1\n",a);

    return 0;
}
```