

TRABAJO PRÁCTICO N° 3

Ejercicios 2

Fecha de entrega:

Forma de entrega: Documento digital en PDF, que incluya las consignas, los datos del alumno, webgrafía y licencia Creative Commons, con permisos de modificación e impresión, cuyo nombre debe ser “Nombre_Apellido-TP3.pdf” enviado a consultas@profmatiasgarcia.com.ar el cual debe tener como asunto “Estructurada TM - tu Nombre y Apellido - TP 3”.

Ayuda: En caso de no comprender alguna consigna o tener dudas, puede solicitarse asistencia enviando un email a consultas@profmatiasgarcia.com.ar con el asunto “Estructurada TM Nombre y Apellido TP3 CONSULTA”.

Modalidad: Individual

Ejercicio 1

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
}
int main()
{
    int x = 1, y;
    y = f1(x);
    printf("%d , %d", x, y);
    return 0;
}
```

- a) 2 , 2
- b) 1 , 2
- c) 1 , 1
- d) 2 , 1

Ejercicio 2

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int y, int x)
{
    return x + y;
}
int main()
{
    int x = 3, y = 7, z;
    printf("%d , %d , %d", x, y, f1(x,y));
    return 0;
}
```

- a) 3 , 7 , 10
- b) Se produce un error en la línea “int f1 ...”
- c) Se produce un error en la línea de return x + y;
- d) Se produce un error en la línea de printf

Ejercicio 3

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
}
int f2(int x, int y)
{
    return x + y;
}
int main()
{
    int x = 3, y = 7, a, b, c;
    a = f1(x);
    b = f1(y);
    c = f2(a,b);
    printf("%d , %d , %d", x, y, c);
    return 0;
}
```

- a) 3 , 7 , 10
- b) 4 , 8 , 12
- c) 3 , 7 , 12
- d) Ninguna de las anteriores

Ejercicio 4

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
}
int f2(int x, int y)
{
    return x + y;
}
int main()
{
    int x = 3, y = 7;
    printf("%d , %d , %d", x, y, f2(f1(x), f1(y)));
    return 0;
}
```

- a) 3 , 7 , 10
- b) 4 , 8 , 12
- c) 3 , 7 , 12
- d) Se produce un error en la línea de printf

Ejercicio 5

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
}
int f2(int x, int y)
{
    return x + y;
}
int main()
{
    int x = 3, y = 7;
    printf("%d , %d , %d", x, y, f2(f1(x), f1(f1(y))));
    return 0;
}
```

- a) 3 , 7 , 10
- b) 3 , 7 , 13
- c) 3 , 7 , 12
- d) Se produce un error en la línea de printf

Ejercicio 6

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
}
int f2(int x, int y)
{
    x = f1(x);
    y = f1(y);
    return x + y;
}
int main()
{
    int x = 3, y = 7;
    printf("%d , %d , %d", x, y, f2(x, y));
    return 0;
}
```

- a) 4 , 8 , 12
- b) 3 , 7 , 12
- c) Se produce un error en tiempo de compilación
- d) Se produce un error en tiempo de ejecución

Ejercicio 7

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x, int y)
{
    x++;
    return x;
}
int f2(int x, int y)
{
    x = f1(x);
    y = f1(y);
    return x + y;
}
int main()
{
    int x = 3, y = 7;
    printf("%d , %d , %d", x, y, f2(x, y));
    return 0;
}
```

- a) 4 , 8 , 12
- b) 3 , 7 , 12
- c) Se produce un error en tiempo de compilación
- d) Se produce un error en tiempo de ejecución

Ejercicio 8

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
    x++;
    return x;
}
int f2(int x, int y)
{
    x = f1(x);
    y = f1(y);
    return x + y;
    return x * y;
}
int main()
{
    int x = 3, y = 7;
    printf("%d , %d , %d", x, y, f2(x, y));
    return 0;
}
```

- a) 3 , 7 , 12
- b) 3 , 7 , 14
- c) 3 , 7 , 32
- d) 3 , 7 , 45

Ejercicio 9

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
}
int f2(int x)
{
    x--;
    return f1(x);
    x++;
}
int f3(int x, int y)
{
    x = f2(x);
    y = f2(y);
    return x + y;
}
int main()
{
    int x = 3, y = 7;
    printf("%d , %d , %d", x, y, f3(x, y));
    return 0;
}
```

- a) 3 , 7 , 12
- b) 3 , 7 , 8
- c) 3 , 7 , 14
- d) 3 , 7 , 10

Ejercicio 10

Indica qué valores se escriben cuándo se compila y ejecuta el siguiente programa:

```
#include <stdio.h>
int f1(int x)
{
    x++;
    return x;
}
int f2(int x)
{
    x--;
    return f1(x);
}
int f3(int x, int y)
{
    x = f2(x);
    y = f2(y);
    return x + y;
}
void f4(int y, int x)
{
    y *= 2;
    x++;
}
```

```
printf("%d , %d , %d", x, y, f3(x, y));  
}  
Int main()  
{  
    int x = 3, y = 7;  
    f4(x, y);  
    return 0;  
}
```

- a) 14 , 4 , 18
- b) 13 , 3 , 16
- c) 8 , 6 , 14
- d) Ninguna de las anteriores

Ejercicio 11

Escribe una función *primo* que determine si un número es primo (la función devolvería 1) o no (devolvería 0). Escribe también un programa, utilizando dicha función, que pida al usuario un número positivo y escriba por pantalla si es o no primo.

Ejercicio 12

Escribe un función *divisorK* que calcule de entre todos los divisores de un número n entero positivo, cuál es el k -ésimo, donde k también es un parámetro de la función. Por ejemplo, si $n=10$ y $k=3$, la función devolverá el 3º de los divisores de 10, es decir 5 (los divisores son 1, 2, 5 y 10). Si k es mayor que el número de divisores la función devuelve -1. Escribe también un programa, utilizando dicha función, que pida al usuario n y k y escriba por pantalla el resultado.

Ejercicio 13

Escribe una función *multiplosK* que calcule la suma de todos los números múltiplos del entero k , comprendidos entre 1 y el entero positivo n . Por ejemplo, si $k=2$ y $n=6$, devuelve 12 (2+4+6). Escribe también un programa, utilizando dicha función, que pida al usuario n y k y escriba por pantalla el resultado.

Ejercicio 14

Escribe una función *potencia3* que a partir de tres números enteros calcule el mayor elevado al menor. Por ejemplo si los números son 2, 3 y 5, devuelve 25 (5²)

Ejercicio 15

Escribe una función *naturales* que imprima los n primeros números naturales, cada uno en una línea, y junto a cada uno de ellos un número de caracteres similar, donde dicho carácter viene determinado por un parámetro c . Por ejemplo, si $n=5$ y $c='*'$, la función escribe:

```
1 *  
2 **  
3 ***  
4 ****  
5 *****
```

Escribe también un programa, utilizando dicha función, que pida al usuario n y c y escriba por pantalla el resultado.

Ejercicio 16

Escribe una función *rectangulo* que dibuje un rectángulo de $m \times n$ caracteres, donde dicho carácter viene determinado por un parámetro c , m indica el número de filas y n de columnas. Por ejemplo, si $m=4$, $n=5$ y $c='#'$, la función escribe:

```
#####  
#####  
#####  
#####
```

Escribe también un programa, utilizando dicha función, que pida al usuario n y c y escriba por pantalla el resultado.

Ejercicio 17

Escribir un programa que dibuje un cuadrado relleno de asteriscos, tal que la longitud de cada lado sea una dada a priori por el usuario. El programa forzosamente debe utilizar la función *rectangulo* del ejercicio anterior. Por ejemplo, para longitud 5, deberá obtener

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

Ejercicio 18

Escribe una función *rectanguloHueco* que dibuje un rectángulo hueco de $m \times n$ caracteres, donde dicho carácter viene determinado por un parámetro c , m indica el número de filas y n de columnas. Por ejemplo, si $m=4$, $n=5$ y $c='@'$, la función escribe:

```
@@@@@  
@      @  
@      @  
@@@@@
```

Escribe también un programa, utilizando dicha función, que pida al usuario n y c y escriba por pantalla el resultado.

Ejercicio 19

Escribe una función *triangulo* que dibuje un triángulo isósceles, tal que su base es vertical y la altura (nótese que es horizontal) contiene n caracteres (donde n es un número dado por el usuario). Además de n , tiene otros dos parámetros: c que indica el carácter que se pintará y *hueco* que es un entero que vale 1 si el triángulo es hueco o 0 si se ha de rellenar. Por ejemplo, para $n=4$, $c='*'$ y *hueco*=0 deberá obtener

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