

2.3 Ejercicios en C: 9, 13 y 15 p 49 Corona y Ancona

Emilio Rojas Badillo

Instituto Tecnológico de Querétaro

Fundamentos de Programación

María Luisa Montes Almanza

2025-09-21

Índice

| | |
|--------------------------------|---|
| 1. Ejercicio 9 | 3 |
| 1.1. Primeros 3 pasos | 3 |
| 1.2. Código | 4 |
| 2. Ejercicio 13 | 5 |
| 2.1. Primeros 3 pasos | 5 |
| 2.2. Código | 6 |
| 3. Ejercicio 15 | 7 |
| 3.1. Primeros 3 pasos | 7 |
| 3.2. Código | 8 |
| 4. Pruebas de escritorio | 9 |

1. Ejercicio 9

1.1. Primeros 3 pasos

1) Definición

$$\text{Total segundos} = (\text{días} * 24 * 3600) + (\text{horas} * 3600) + (\text{minutos} * 60) + \text{segundos}$$

2) Análisis

| | | |
|----------|----------------------------|------------------------------|
| C | (P) | $\xrightarrow{\text{Total}}$ |
| días | | |
| horas | Total = (días * 24 * 3600) | |
| minutos | + (horas * 3600) | |
| segundos | + (minutos * 60) + | |
| | segundos | |

Condiciones

$$\begin{aligned} \text{días} &\geq 0 & \text{Total} &\geq 0 \\ \text{horas} &\geq 0 & & \\ \text{minutos} &\geq 0 & & \\ \text{segundos} &\geq 0 & & \end{aligned}$$

3) Algoritmo

Operación de datos

| Variante | Tipo | Comentario |
|----------|------|---------------------|
| días | Real | días a convertir |
| horas | Real | horas a convertir |
| minutos | Real | minutos a convertir |
| segundos | Real | segundos a sumar |
| total | Real | Suma de segundos |

Diseño de flujo

```

    graph TD
        Start((Inicio)) --> InputD[Ingresar días]
        InputD --> D[days]
        D --> InputH[Ingresar minutos]
        InputH --> H[minutos]
        H --> InputM[Ingresar horas]
        InputM --> M[horas]
        M --> InputS[Ingresar segundos]
        InputS --> S[segundos]
        S --> Total[Total = (días * 24 * 3600) + (horas * 3600) + (minutos * 60) + segundos]
        Total --> TotalText["Total de segundos"]
        TotalText --> Fin((Fin))
    
```

Pseudocódigo

```

    Principal()
    Real días, minutos, horas, segundos, total
    INICIO
        IMPRIMIR "Ingrese días"
        LEER días
        IMPRIMIR "Ingrese horas"
        LEER horas
        IMPRIMIR "Ingrese minutos"
        LEER minutos
        IMPRIMIR "Ingrese segundos"
        LEER segundos
        HACER total <- (días * 24 * 3600) +
                    (horas * 3600) +
                    (minutos * 60) +
                    segundos
        IMPRIMIR "Total de segundos: ", total
    FIN
    
```

1.2. Código

```

21 September 01:25 PM
File Edit Mode Functions Variables Units Help
day + hour + minute + second(50) -> s
day + hour + minute + (second × 50) to second
RAD

= 90 110 s

warning: Git tree '/horas'
warning: Git tree '/minutos'
warning: Git tree '/segundos'

Total de segundos en 1 dias 1 horas 1, 50 minutos y 0 segundos: 90110
0.00[emilio@user1012007][ej9]{main}$

Terminal bash
ej9 95/38

```

```

1 #include <stdio.h>
2
3 int main(int argc, char *argv[]) {
4     int dias, horas, minutos, segundos;
5
6     do {
7         printf("Ingrese dias: \n");
8         scanf("%d", &dias);
9     } while (dias < 0);
10    do {
11        printf("Ingrese horas: \n");
12        scanf("%d", &horas);
13    } while (horas < 0);
14    do {
15        printf("Ingrese minutos: \n");
16        scanf("%d", &minutos);
17    } while (minutos < 0);
18    do {
19        printf("Ingrese segundos: \n");
20        scanf("%d", &segundos);
21    } while (segundos < 0);
22
23    float total =
24        (dias * 24 * 60 * 60) +
25        (horas * 60 * 60) +
26        (minutos * 60) +
27        segundos;
28
29    printf("Total de segundos es %.2f",
30           dias, horas, minutos, segundos, total);
31
32    return 0;
33 }

```

2. Ejercicio 13

2.1. Primeros 3 pasos

1) Definición

$$A = \frac{1}{2} [(x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2))]$$

2) Análisis

$$A = \frac{1}{2} [(x_1 * (y_2 - y_3) + x_2 * (y_3 - y_1) + x_3 * (y_1 - y_2))] / 2$$

3) Algoritmo

| x_1, x_2, x_3 | y_1, y_2, y_3 | A | Comentario |
|-----------------|-----------------|------|-------------|
| Real | Real | Real | Área de A |
| Real | Real | Real | Área de B |
| Real | Real | Real | Área de C |
| Real | Real | Real | Origen de A |
| Real | Real | Real | Origen de B |
| Real | Real | Real | Origen de C |
| Real | Real | Real | Arctotal |

Diagrama de flujo

```

    graph TD
        Inicio([Início]) --> InsertarA([Inserir coord. A])
        InsertarA --> CoordenadasA["x1, y1"]
        CoordenadasA --> InsertarB([Inserir coord. B])
        InsertarB --> CoordenadasB["x2, y2"]
        CoordenadasB --> InsertarC([Inserir coord. C])
        InsertarC --> CoordenadasC["x3, y3"]
        CoordenadasC --> CalcularA["A = 1/2 * (x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)) / 2"]
        CalcularA --> MostrarA(["A = ", A])
        MostrarA --> Fin([Fim])
    
```

Pseudo código

PRINCIPAL()
 DECLARAR $x_1, y_1, x_2, y_2, x_3, y_3, A$
INICIO
 IMPRIMIR "Informe coord. A"
 LEER x_1, y_1
 IMPRIMIR "Informe coord. B"
 LEER x_2, y_2
 IMPRIMIR "Informe coord. C"
 LEER x_3, y_3
 HACER $A \leftarrow (x_1 * (y_2 - y_3) + x_2 * (y_3 - y_1) + x_3 * (y_1 - y_2)) / 2$
 IMPRIMIR "A = ", A
FIN

2.2. Código

The screenshot shows a terminal window titled 'Flake.nix' with the command 'p49e13N_6A_23..'. The terminal displays the following code:

```
1 #include <math.h>
2 #include <stdio.h>
3
4 int main(int argc, char *argv[]) {
5     float x1, y1, x2, y2, x3, y3, a;
6     printf("ingrese coordenada A: \n");
7     scanf("%f %f", &x1, &y1);
8     printf("ingrese coordenada B: \n");
9     scanf("%f %f", &x2, &y2);
10    printf("ingrese coordenada C: \n");
11    scanf("%f %f", &x3, &y3);
12
13    a = fabs(x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2));
14
15    printf("El area total es: %f\n", a);
16    return 0;
17 }
```

Output from the terminal:

```
warning: creating lock file '/home/emilio/Documents/ITQ/semestre-1/2.3/ej13/flake.lock':
  • Added input 'nixpkgs':
    'github:nixos/nixpkgs/8eaee110344796db060382e15d3af0a9fc396e0e?n
    arHash=sha256-icGWf/LTy%2BaY0zFu8q12lK8KuZp7yvdhStehhyX1v8w%3D' (202
    5-09-19)
warning: Git tree '/home/emilio/Documents/ITQ/semestre-1' is dirty
ingrese coordenada A:
8 3
ingrese coordenada B:
6 7
ingrese coordenada C:
6 3
El area total es: 4.00
[emilio@user1012007][ej13]{main}$
```

The terminal window has a dark theme and includes status icons at the top right (49%, 100%) and bottom right (ej13, 96/22).

3. Ejercicio 15

3.1. Primeros 3 pasos

Ej 15

1) Definición

$$\text{Perímetro} = 3 \times 1$$

$$\text{Altura} = (3^{(1/2)} \times 1) / 2$$

$$\text{área} = ((3^{(1/2)}) / 4) \times 1^2$$

2) Análisis

Condicionales

- $\text{lado} > 0$
- $P > 0$
- $h > 0$
- $a > 0$

3) Algoritmo

| variable | tipo | comentario |
|----------|------|-------------------|
| lado | Real | lado de triángulo |
| P | Real | perímetro |
| h | Real | altura |
| a | Real | área |

Diseño de flujo

```

graph TD
    Inicio([Inicio]) --> Ingreselado([Ingresar el lado])
    Ingreselado --> Lado[lado]
    Lado --> P["P = 3 * lado"]
    P --> H["h = (3^(1/2)) * lado / 2"]
    H --> A["a = ((3^(1/2)) / 4) * lado^2"]
    A --> Fin([Fin])
    
```

Pseudocódigo

Principal()

Decl lado, P, h, a

INICIO

IMPRIMIR "Ingresar el lado"

LEER lado

HACER $P \leftarrow 3 * \text{lado}$

HACER $h \leftarrow (3^{(1/2)} \times \text{lado}) / 2$

HACER $a \leftarrow ((3^{(1/2)}) / 4) \times \text{lado}^2$

IMPRIMIR "Perímetro:", P

IMPRIMIR "altura:", h

IMPRIMIR "área:", a

FIN

3.2. Código

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a dark background and contains the following code and its execution:

```
p49e15N_6A_23.x  flake.nix
1 #include <math.h>
2 #include <stdio.h>
3
4 int main(int argc, char *argv[]) {
5     float lado, p, h, a;
6     do {
7         printf("Ingrese el lado: \n");
8         scanf("%f", &lado);
9     } while (lado <= 0);
10    p = 3.0 * lado;
11    h = (sqrt(3) * lado) / 2.0;
12    a = (sqrt(3) * pow(lado, 2)) /
13    printf("perimetro: %.2f\n", p)
14    printf("altura: %.2f\n", h);
15    printf("area: %.2f\n", a);
16    return 0;
17 }
```

[emilio@user1012007][ej15]{main}\$ nix run
warning: Git tree '/home/emilio/Documents/ITQ/semanestre-1' is dirty
Ingrese el lado:
1
perimetro: 3.00
altura: 0.87
area: 0.43
[emilio@user1012007][ej15]{main}\$ nix run
warning: Git tree '/home/emilio/Documents/ITQ/semanestre-1' is dirty
Ingrese el lado:
6
perimetro: 18.00
altura: 5.20
area: 15.59
[emilio@user1012007][ej15]{main}\$

The terminal window is titled "flake.nix". The status bar at the bottom of the screen shows "ej15" and "303/35".

4. Pruebas de escritorio

Prueba de escritorio P5 2.9

$$\begin{array}{|c|c|c|c|c|c|} \hline d & h & m & s & total \\ \hline 1 & 1 & 1 & 50 & total = (1 * 24 * 3600) + (1 * 3600) + (1 * 60) + 50 \\ & & & & total = (86400) + (3600) + (60) + 50 \\ & & & & total = 90110 \end{array}$$



Prueba de escritorio P5 2.13

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline x_1 & x_1 & x_2 & x_2 & x_3 & x_3 & A \\ \hline 8 & 3 & 6 & 17 & 6 & 3 & \\ \hline \end{array}$$

$$\begin{aligned}
 A &= 1(8 * (8 - 3) + 6 * (3 - 3) + 6 * (3 - 7)) / 2 \\
 A &= 1(32) + 24 / 2 \\
 A &= 8 / 2 \\
 A &= 4
 \end{aligned}$$

Prueba de escritorio P5 2.15

$$\begin{array}{l}
 l = 10 \\
 6 \\
 \rho = 3 * 6 \\
 \rho = 18
 \end{array}
 \left\{
 \begin{array}{l}
 \rho = 18 \\
 h = ((3^{\wedge}(1/2)) * l) / 2 \\
 h = 5 * 2
 \end{array}
 \right\}
 \left\{
 \begin{array}{l}
 h = ((3^{\wedge}(1/2)) * l) / 2 \\
 h = 5 * 2
 \end{array}
 \right\}
 \left\{
 \begin{array}{l}
 a = (\tilde{3}^{\wedge}(1/2) * 6^{\wedge}2) * l \\
 a = (\tilde{3}^{\wedge}(1/2) * 36) * 10 \\
 a = 15.59
 \end{array}
 \right\}$$