

Taller: Precedencia de Operaciones con Variables – 6 Ejercicios Propuestos

Carrera: Ing. Electronica y Automatizacion

Asignatura: Fund. Programacion

Tema del taller: Precedencia de Operaciones con Variables

Docente: Jenny Ruiz

Nombre: Andrango Tobias

Fecha: 24/11/2025 Paralelo: 29583

Ejercicio 1

Expresión: $(3 * j + 8 / k) != i * k - j$

Valores: {'i': 4, 'j': 2, 'k': 8}

Solución paso a paso:

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

```
#include <stdbool.h>
```

```
int main() {
```

```
    int i = 4;
```

```
    int j = 2;
```

```
    int k = 8;
```

```
    int lado_izquierdo = (3 * j + 8 / k);
```

```
    int lado_derecho = (i * k - j);
```

```
    int resultado_entero = (lado_izquierdo != lado_derecho);
```

```
    bool resultado_booleano = (lado_izquierdo != lado_derecho);
```

```

printf("Variables:\n");

printf(" i = %d\n", i);

printf(" j = %d\n", j);

printf(" k = %d\n", k);

printf("\n");

printf("Evaluación del Lado Izquierdo (3 * j + 8 / k):\n");

printf(" (3 * %d + 8 / %d) = (%d + %d) = %d\n", j, k, (3*j), (8/k), lado_izquierdo);


printf("Evaluación del Lado Derecho (i * k - j):\n");

printf(" (%d * %d - %d) = (%d - %d) = %d\n", i, k, j, (i*k), j, lado_derecho);


printf("\n");

printf("Evaluación de la expresión completa (LI != LD):\n");

printf(" (%d != %d) -> %s\n", lado_izquierdo, lado_derecho, resultado_booleano ?
"Verdadero (1)" : "Falso (0)");

return 0;

}

```

```

1  #include <stdio.h>
2  #include <stdbool.h>
3
4  int main() {
5      int i = 4;
6      int j = 2;
7      int k = 8;
8
9      int lado_izquierdo = (3 * j + 8 / k);
10
11     int lado_derecho = (i * k - j);
12     int main::j
13     int resultado_entero = (lado_izquierdo != lado_derecho);
14
15     bool resultado_booleano = (lado_izquierdo != lado_derecho);
16
17     printf("Variables:\n");
18     printf(" i = %d\n", i);
19     printf(" j = %d\n", j);
20     printf(" k = %d\n", k);
21     printf("\n");
22
23     printf("Evaluación del Lado Izquierdo (3 * j + 8 / k):\n");
24     printf(" (3 * %d + 8 / %d) = (%d + %d) = %d\n", j, k, (3*j), (8/k), lado_izquierdo);
25
26     printf("Evaluación del Lado Derecho (i * k - j):\n");
27     printf(" (%d * %d - %d) = (%d - %d) = %d\n", i, k, j, (i*k), j, lado_derecho);
28
29     printf("\n");
30     printf("Evaluación de la expresión completa (LI != LD):\n");
31     printf(" (%d != %d) -> %s\n", lado_izquierdo, lado_derecho, resultado_booleano ? "Verdadero (1)" : "Falso (0)");
32
33
34     return 0;
35 }
36

```

Ejercicio 2

Expresión: $m * (n + p) / 2 \geq p - n$

Valores: {'m': 6, 'n': 3, 'p': 9}

Solución paso a paso:

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

```
#include <stdbool.h>
```

```
int main() {
```

```
    int m = 6;
```

```
    int n = 3;
```

```
    int p = 9;
```

```
    int lado_izquierdo = m * (n + p) / 2;
```

```
    int lado_derecho = p - n;
```

```

int resultado_entero = (lado_izquierdo >= lado_derecho);

printf("--- Evaluación del Ejercicio 2 ---\n");
printf("Expresión:  $m * (n + p) / 2 \geq p - n$ \n");
printf("Valores: m=%d, n=%d, p=%d\n", m, n, p);
printf("-----\n");

printf("Paso 1: Cálculo del Lado Izquierdo (LI)\n");
printf("   $m * (n + p) / 2$ \n", m, n, p);
printf("   $m * (n + p) / 2$ \n", m, (n + p));
printf("   $m * (n + p) / 2 =$  %d\n", (m * (n + p)), lado_izquierdo);

printf("\nPaso 2: Cálculo del Lado Derecho (LD)\n");
printf("   $p - n =$  %d\n", p, n, lado_derecho);

printf("\nPaso 3: Comparación final (LI >= LD)\n");
printf("   $m * (n + p) / 2 \geq p - n \rightarrow$  %s\n", lado_izquierdo, lado_derecho, resultado_entero ?
"Verdadero (1)" : "Falso (0)");

return 0;
}

```

```

1  #include <stdio.h>
2  #include <stdbool.h>
3
4  int main() {
5      int m = 6;
6      int n = 3;
7      int p = 9;
8
9      int lado_izquierdo = m * (n + p) / 2;
10
11     int lado_derecho = p - n;
12
13     int resultado_entero = (lado_izquierdo >= lado_derecho);
14
15     printf("--- Evaluación del Ejercicio 2 ---\n");
16     printf("Expresión: m * (n + p) / 2 >= p - n\n");
17     printf("Valores: m=%d, n=%d, p=%d\n", m, n, p);
18     printf("-----\n");
19
20     printf("Paso 1: Cálculo del Lado Izquierdo (LI)\n");
21     printf("  %d * (%d + %d) / 2\n", m, n, p);
22     printf("  %d * %d / 2\n", m, (n + p));
23     printf("  %d / 2 = %d\n", (m * (n + p)), lado_izquierdo);
24
25     printf("\nPaso 2: Cálculo del Lado Derecho (LD)\n");
26     printf("  %d - %d = %d\n", p, n, lado_derecho);
27
28     printf("\nPaso 3: Comparación final (LI >= LD)\n");
29     printf("  %d >= %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ? "Verdadero (1)" : "Falso (0)");
30
31     return 0;
32 }
33

```

Ejercicio 3

Expresión: $(a + b * c) < (c / b + a)$

Valores: {'a': 5, 'b': 2, 'c': 10}

Solución paso a paso:

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

```
#include <stdbool.h>
```

```
int main() {
```

```
    int a = 5;
```

```
    int b = 2;
```

```
    int c = 10;
```

```
    int lado_izquierdo = (a + b * c);
```

```
    int lado_derecho = (c / b + a);
```

```
int resultado_entero = (lado_izquierdo < lado_derecho);

printf(" Evaluación del Ejercicio 3 \n");

printf("Expresión: (a + b * c) < (c / b + a)\n");

printf("Valores: a=%d, b=%d, c=%d\n", a, b, c);


printf("Paso 1: Lado Izquierdo (LI)\n");

printf(" (%d + %d * %d) = (%d + %d) = %d\n", a, b, c, a, (b * c), lado_izquierdo);


printf("\nPaso 2: Lado Derecho (LD)\n");

printf(" (%d / %d + %d) = (%d + %d) = %d\n", c, b, a, (c / b), a, lado_derecho);


printf("\nPaso 3: Comparación final (LI < LD)\n");

printf(" %d < %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ? "Verdadero  
(1)" : "Falso (0)");


return 0;

}
```

```

1  #include <stdio.h>
2  #include <stdbool.h>
3
4  int main() {
5      int a = 5;
6      int b = 2;
7      int c = 10;
8
9      int lado_izquierdo = (a + b * c);
10
11     int lado_derecho = (c / b + a);
12
13
14     int resultado_entero = (lado_izquierdo < lado_derecho);
15
16
17     printf(" Evaluación del Ejercicio 3 \n");
18     printf("Expresión: (a + b * c) < (c / b + a)\n");
19     printf("Valores: a=%d, b=%d, c=%d\n", a, b, c);
20
21     printf("Paso 1: Lado Izquierdo (LI)\n");
22     printf("  (%d + %d * %d) = (%d + %d) = %d\n", a, b, c, a, (b * c), lado_izquierdo);
23
24     printf("\nPaso 2: Lado Derecho (LD)\n");
25     printf("  (%d / %d + %d) = (%d + %d) = %d\n", c, b, a, (c / b), a, lado_derecho);
26
27     printf("\nPaso 3: Comparación final (LI < LD)\n");
28     printf("  %d < %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ? "Verdadero (1)" : "Falso (0)");
29
30     return 0;
31 }

```

Ejercicio 4

Expresión: $x / (y - 1) + 3 * y == x - y$

Valores: {'x': 14, 'y': 3}

Solución paso a paso:

- Sustituir valores: x=14, y=3

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

```
#include <stdbool.h>
```

```
int main() {
```

```
    int x = 14;
```

```
    int y = 3;
```

```
    int lado_izquierdo = x / (y - 1) + 3 * y;
```

```
    int lado_derecho = x - y;
```

```
    int resultado_entero = (lado_izquierdo == lado_derecho);
```

```

printf("--- Evaluación del Ejercicio 4 ---\n");

printf("Expresión:  $x / (y - 1) + 3 * y == x - y$ \n");

printf("Valores: x=%d,y=%d\n", x, y);

printf("-----\n");


printf("Paso 1: Lado Izquierdo (LI)\n");

printf("  %d / (%d - 1) + 3 * %d\n", x, y, y);

printf("  %d / %d + %d\n", x, (y - 1), (3 * y));

printf("  %d + %d = %d\n", (x / (y - 1)), (3 * y), lado_izquierdo);


printf("\nPaso 2: Lado Derecho (LD)\n");

printf("  %d - %d = %d\n", x, y, lado_derecho);


printf("\nPaso 3: Comparación final (LI == LD)\n");

printf("  %d == %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ?
"Verdadero (1)" : "Falso (0)");


return 0;
}

```

```

1  #include <stdio.h>
2  #include <stdbool.h>
3
4  int main() {
5      int x = 14;
6      int y = 3;
7
8      int lado_izquierdo = x / (y - 1) + 3 * y;
9
10     int lado_derecho = x - y;
11
12     int resultado_entero = (lado_izquierdo == lado_derecho);
13
14     printf("--- Evaluación del Ejercicio 4 ---\n");
15     printf("Expresión: x / (y - 1) + 3 * y == x - y\n");
16     printf("Valores: x=%d, y=%d\n", x, y);
17     printf("-----\n");
18
19     printf("Paso 1: Lado Izquierdo (LI)\n");
20     printf("  %d / (%d - 1) + 3 * %d\n", x, y, y);
21     printf("  %d / %d + %d\n", x, (y - 1), (3 * y));
22     printf("  %d + %d = %d\n", (x / (y - 1)), (3 * y), lado_izquierdo);
23
24     printf("\nPaso 2: Lado Derecho (LD)\n");
25     printf("  %d - %d = %d\n", x, y, lado_derecho);
26
27     printf("\nPaso 3: Comparación final (LI == LD)\n");
28     printf("  %d == %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ? "Verdadero (1)" : "Falso (0)");
29
30     return 0;
31 }
32

```

Ejercicio 5

Expresión: $(u * v - 4) \leq (v + u / 2)$

Valores: {'u': 12, 'v': 4}

Solución paso a paso:

- Sustituir valores: u=12, v=4

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

```
int main() {
```

```
    int u = 12;
```

```
    int v = 4;
```

```
    int lado_izquierdo = (u * v - 4);
```

```
    int lado_derecho = (v + u / 2);
```

```
    int resultado_entero = (lado_izquierdo <= lado_derecho);
```

```

printf("--- Evaluación del Ejercicio 5 ---\n");

printf("Expresión: (u * v - 4) <= (v + u / 2)\n");

printf("Valores: u=%d, v=%d\n", u, v);

printf("-----\n");


printf("Paso 1: Lado Izquierdo (LI)\n");

printf(" (%d * %d - 4) = (%d - 4) = %d\n", u, v, (u * v), lado_izquierdo);


printf("\nPaso 2: Lado Derecho (LD)\n");

printf(" (%d + %d / 2) = (%d + %d) = %d\n", v, u, v, (u / 2), lado_derecho);


printf("\nPaso 3: Comparación final (LI <= LD)\n");

printf(" %d <= %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ?
"Verdadero (1)" : "Falso (0)");


return 0;

}

```

```

1  #include <stdio.h>
2
3  int main() {
4
5      int u = 12;
6      int v = 4;
7
8      int lado_izquierdo = (u * v - 4);
9
10     int lado_derecho = (v + u / 2);
11
12     int resultado_entero = (lado_izquierdo <= lado_derecho);
13
14     printf("--- Evaluación del Ejercicio 5 ---\n");
15     printf("Expresión: (u * v - 4) <= (v + u / 2)\n");
16     printf("Valores: u=%d, v=%d\n", u, v);
17     printf("-----\n");
18
19     printf("Paso 1: Lado Izquierdo (LI)\n");
20     printf(" (%d * %d - 4) = (%d - 4) = %d\n", u, v, (u * v), lado_izquierdo);
21
22     printf("\nPaso 2: Lado Derecho (LD)\n");
23     printf(" (%d + %d / 2) = (%d + %d) = %d\n", v, u, v, (u / 2), lado_derecho);
24
25     printf("\nPaso 3: Comparación final (LI <= LD)\n");
26     printf(" %d <= %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ? "Verdadero (1)" : "Falso (0)");
27
28     return 0;
29 }

```

Ejercicio 6

Expresión: $(q + r * 2) != (q * r - 10)$

Valores: {'q': 7, 'r': 5}

Solución paso a paso:

- Sustituir valores: q=7, r=5

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

```
int main() {
```

```
    int q = 7;
```

```
    int r = 5;
```

```
    int lado_izquierdo = (q + r * 2);
```

```
    int lado_derecho = (q * r - 10);
```

```
    int resultado_entero = (lado_izquierdo != lado_derecho);
```

```
    printf("--- Evaluación del Ejercicio 6 ---\n");
```

```
    printf("Expresión: (q + r * 2) != (q * r - 10)\n");
```

```
    printf("Valores: q=%d, r=%d\n", q, r);
```

```
    printf("-----\n");
```

```
    printf("Paso 1: Lado Izquierdo (LI)\n");
```

```
    printf(" (%d + %d * 2) = (%d + %d) = %d\n", q, r, q, (r * 2), lado_izquierdo);
```

```
    printf("\nPaso 2: Lado Derecho (LD)\n");
```

```
printf(" (%d * %d - 10) = (%d - 10) = %d\n", q, r, (q * r), lado_derecho);
```

```
printf("\nPaso 3: Comparación final (LI != LD)\n");
```

```
printf(" %d != %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ? "Verdadero (1)" : "Falso (0)");
```

```
return 0;
```

```
}
```

```
1  #include <stdio.h>
2
3  int main() {
4      int q = 7;
5      int r = 5;
6
7      int lado_izquierdo = (q + r * 2);
8
9      int lado_derecho = (q * r - 10);
10
11     int resultado_entero = (lado_izquierdo != lado_derecho);
12
13     printf("--- Evaluación del Ejercicio 6 ---\n");
14     printf("Expresión: (q + r * 2) != (q * r - 10)\n");
15     printf("Valores: q=%d, r=%d\n", q, r);
16     printf("-----\n");
17
18     printf("Paso 1: Lado Izquierdo (LI)\n");
19     printf(" (%d + %d * 2) = (%d + %d) = %d\n", q, r, q, (r * 2), lado_izquierdo);
20
21     printf("\nPaso 2: Lado Derecho (LD)\n");
22     printf(" (%d * %d - 10) = (%d - 10) = %d\n", q, r, (q * r), lado_derecho);
23
24     printf("\nPaso 3: Comparación final (LI != LD)\n");
25     printf(" %d != %d -> %s\n", lado_izquierdo, lado_derecho, resultado_entero ? "Verdadero (1)" : "Falso (0)");
26
27     return 0;
28 }
```