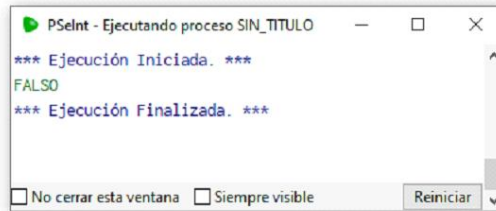


ExpreAritmeRel1_1_1_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas. a= 5, b=7, c=4.
3      //(NOTA. el operador ^ es el operador de exponenciación, es decir,
4      //por ejemplo a ^ 3 sería "a elevado a 3".
5      //En el orden de precedencia, el operador ^ se evalúa después de los paréntesis
6      //pero antes de la multiplicación, división y módulo).
7
8      Definir a,b,c Como Real;
9      Definir resultado Como Logico;
10
11
12      a←5;
13      b←7;
14      c←4;
15
16
17
18      resultado←((3*2+2-4/(2*1)) > (3*(2+2)*1) y (5 > 11 Mod c));
19      Escribir resultado;
20
21  FinProceso
22

```

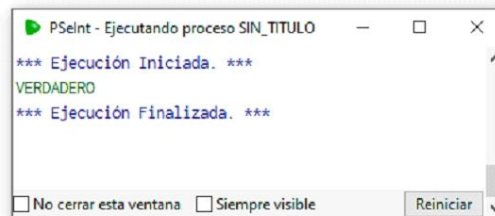


ExpreAritmeRel1_1_2_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas. a= 5, b=7, c=4.
3      //(NOTA. el operador ^ es el operador de exponenciación, es decir,
4      //por ejemplo a ^ 3 sería "a elevado a 3".
5      //En el orden de precedencia, el operador ^ se evalúa después de los paréntesis
6      //pero antes de la multiplicación, división y módulo).
7
8      Definir a,b,c Como Real;
9      Definir resultado Como Logico;
10
11
12
13      a←5;
14      b←7;
15      c←4;
16
17
18
19      resultado←(((3≥3) o (5≠a)) y (no((15/a +2) ≠ 5)));
20      Escribir resultado;
21
22  FinProceso
23

```

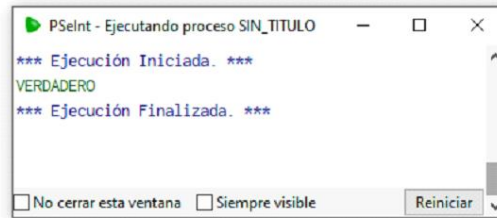


ExpreAritmeRel1_1_3_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas. a= 5, b=7, c=4.
3      //(NOTA. el operador ^ es el operador de exponenciación, es decir,
4      //por ejemplo a ^ 3 sería "a elevado a 3".
5      //En el orden de precedencia, el operador ^ se evalúa después de los paréntesis
6      //pero antes de la multiplicación, división y módulo).
7
8      Definir a,b,c Como Real;
9      Definir resultado Como Logico;
10
11
12
13      a←5;
14      b←7;
15      c←4;
16
17
18
19      resultado←(No(No(((3*(-3))*2)>(3-(-3)*2)o((1+3*2)<6))));
20      Escribir resultado;
21
22 FinProceso
23

```

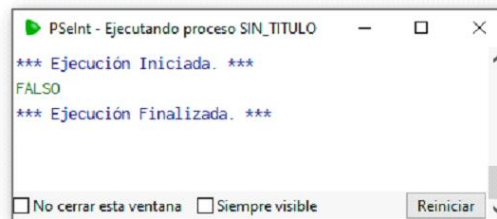


ExpreAritmeRel1_1_4_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas. a= 5, b=7, c=4.
3      //(NOTA. el operador ^ es el operador de exponenciación, es decir,
4      //por ejemplo a ^ 3 sería "a elevado a 3".
5      //En el orden de precedencia, el operador ^ se evalúa después de los paréntesis
6      //pero antes de la multiplicación, división y módulo).
7
8      Definir a,b,c Como Real;
9      Definir resultado Como Logico;
10
11
12
13      a←5;
14      b←7;
15      c←4;
16
17
18
19      resultado←(( (3≥c) y (a>3) y (3>(c-1)) ) o (no( (c≤4) o (a>4) o (6≥b) )));
20      Escribir resultado;
21
22 FinProceso
23

```

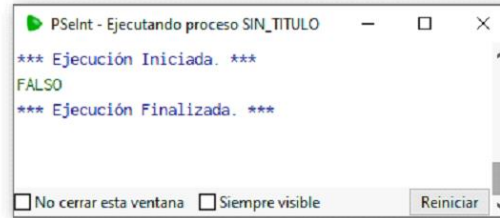


ExpreAritmeRel1_2_1_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3      //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4      //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5      //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6      //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7      //el primer operando entre el segundo")
8
9
10     Definir x,i,z Como Real;//la variable y ya esta definida
11     Definir resultado Como Logico;
12
13
14
15     x←10;
16     i←20;
17     z←30;
18
19
20
21     resultado←((x<10) o (x>10));
22     Escribir resultado;
23
24 FinProceso
25

```



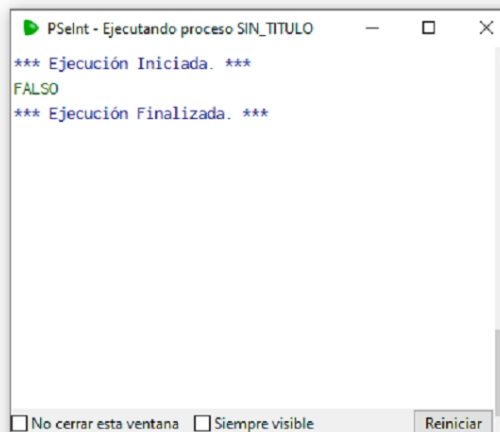
ExpreAritmeRel1_2_2_ArjonillaBermudez_Francisco

Este proceso lógico siempre será Falso independientemente del valor que se otorgue a las variables

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3      //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4      //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5      //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6      //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7      //el primer operando entre el segundo")
8
9
10     Definir x,i,z Como Real;//la variable y ya esta definida
11     Definir resultado Como Logico;
12
13
14
15     x←10;
16     i←20;
17     z←30;
18
19
20
21     resultado←((x > i) y ( i>x ));
22     Escribir resultado;
23
24 FinProceso
25

```

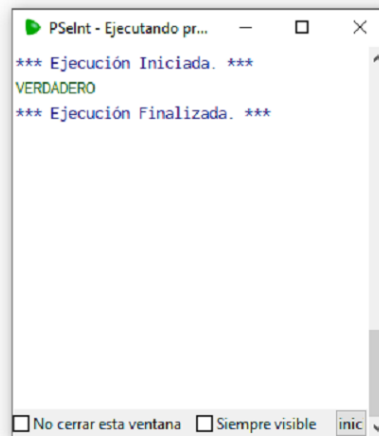


ExpreAritmeRel1_2_3_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3      //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4      //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5      //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6      //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7      //el primer operando entre el segundo")
8
9
10     Definir x,i,z Como Real;//la variable y ya esta definida
11     Definir resultado Como Logico;
12
13
14
15     x←10;
16     i←20;
17     z←30;
18
19
20
21     resultado←((x < (i + z)) y ( (x + 10) ≤ 20));
22     Escribir resultado;
23
24 FinProceso
25

```

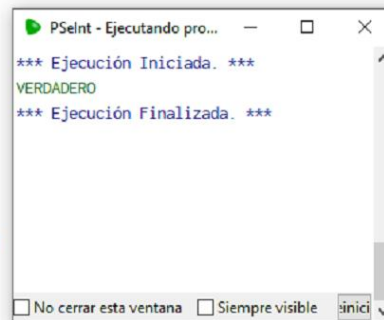


ExpreAritmeRel1_2_4_ArjonillaBermudez_Francisco

```

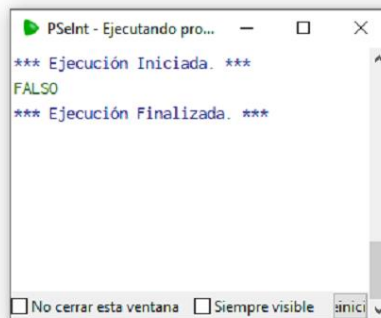
1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3      //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4      //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5      //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6      //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7      //el primer operando entre el segundo")
8
9
10     Definir x,i,z Como Real;//la variable y ya esta definida
11     Definir resultado Como Logico;
12
13
14
15     x←10;
16     i←20;
17     z←30;
18
19
20
21     resultado←(No (No (x < (i + z)) y ( (x + 10) ≠ i)));
22     Escribir resultado;
23
24 FinProceso
25

```



ExpreAritmeRel1_2_5_ArjonillaBermudez_Francisco

```
1  Proceso sin_titulo
2  //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3  //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4  //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5  //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6  //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7  //el primer operando entre el segundo")
8
9
10 Definir x,i,z Como Real;//la variable y ya esta definida
11 Definir resultado Como Logico;
12
13
14
15 x←10;
16 i←20;
17 z←30;
18
19
20
21 resultado←(((z - i) = x) y ((z Mod i) = 2));
22 Escribir resultado;
23
24 FinProceso
25
```



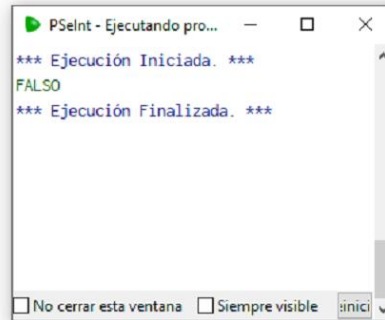
ExpreAritmeRel1_2_6_ArjonillaBermudez_Francisco

Este proceso lógico siempre será Falso independientemente del valor que se otorgue a las variables

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3      //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4      //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5      //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6      //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7      //el primer operando entre el segundo")
8
9
10     Definir x,i,z Como Real; //la variable y ya esta definida
11     Definir resultado Como Logico;
12
13
14
15     x←10;
16     i←20;
17     z←30;
18
19
20
21     resultado←((x < 10) y (x > 10));
22     Escribir resultado;
23
24 FinProceso
25

```



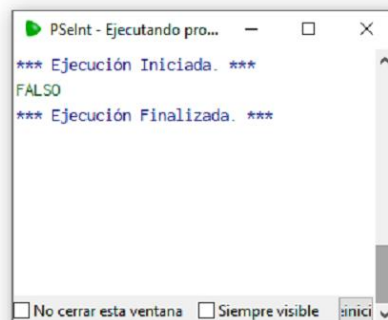
ExpreAritmeRel1_2_7_ArjonillaBermudez_Francisco

Este proceso lógico siempre será Falso independientemente del valor que se otorgue a las variables

```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3      //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4      //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5      //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6      //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7      //el primer operando entre el segundo")
8
9
10     Definir x,i,z Como Real; //la variable y ya esta definida
11     Definir resultado Como Logico;
12
13
14
15     x←10;
16     i←20;
17     z←30;
18
19
20
21     resultado←((x>i)y(i>x));
22     Escribir resultado;
23
24 FinProceso
25

```

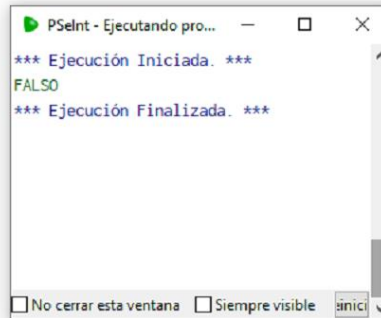


ExpreAritmeRel1_2_8_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2  //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ç
3  //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4  //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5  //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6  //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7  //el primer operando entre el segundo")
8
9
10 Definir x,i,z Como Real;//la variable y ya esta definida
11 Definir resultado Como Logico;
12
13
14
15 x←10;
16 i←20;
17 z←30;
18
19
20 resultado←(No(x<(i+z))o(No(x+10)≤20));
21 Escribir resultado;
22
23 FinProceso
24

```

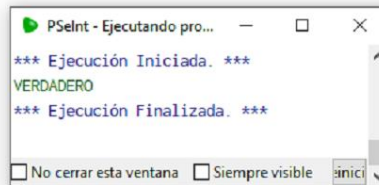


ExpreAritmeRel1_2_9_ArjonillaBermudez_Francisco

```

1  Proceso sin_titulo
2  //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ç
3  //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4  //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5  //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6  //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7  //el primer operando entre el segundo")
8
9
10 Definir x,i,z Como Real;//la variable y ya esta definida
11 Definir resultado Como Logico;
12
13
14
15 x←10;
16 i←20;
17 z←30;
18
19
20 resultado←(NO(x=i)y (x≠ i) y (x<i v i<x)o((i Mod 10) =0)y ( z Mod (x*4) = 30));
21 Escribir resultado;
22
23 FinProceso
24

```



ExpreAritmeRel1_2_10_ArjonillaBermudez_Francisco


```

1  Proceso sin_titulo
2      //Evalúa las siguientes expresiones aritmético-lógicas, siendo x=10, y=20, and z=30. ¿
3      //Ten en cuenta que algunas están escritas en notación de pseudo-código pero otras lo están
4      //en lenguaje de programación C (o Java). Indica si alguna de las expresiones es siempre
5      //verdadera o siempre falsa independientemente de los valores de x, y, z. (NOTA.
6      //El operador % es ? en C y Java- el operador "módulo" o "resto de dividir de forma entera
7      //el primer operando entre el segundo")
8
9
10     Definir x,i,z Como Real; //la variable y ya esta definida
11     Definir resultado Como Logico;
12
13
14
15     x←10;
16     i←20;
17     z←30;
18
19
20     resultado←(NO((x=i)y (x≠ i) y ((x<i ∨ i<x)o((i Mod 10) =0)))y ( z Mod (x*4) = 30));
21     Escribir resultado;
22
23 FinProceso
24

```

