

## SOLUCIONARIO TEMA 2

29. El enunciado de este programa es ambiguo, estas son sus dos interpretaciones:

```
algoritmo_29(x, y){  
    z := sqrt(x) - y;  
}  
  
algoritmo_29(x, y){  
    z := sqrt(x - y);  
}
```

30. algoritmo\_30(l){

```
    A := (l*sqrt(l^2 - (l/2)^2))/2;  
}
```

31. El enunciado de este programa es ambiguo, estas son sus dos interpretaciones:

```
algoritmo_31(x, y){  
    z := x - y;  
}  
  
algoritmo_31(x, y){  
    z := y - x;  
}
```

32. algoritmo\_32(p){

cases of p:

```
    1: L;  
    2: M;  
    3: X;  
    4: J;  
    5: V;  
    6: S;  
    7: D;
```

```
}
```

33. algoritmo\_33(){

```
    n := 0;  
    suma := 0;  
    repeat{  
        scan(x);  
        suma := suma + x;  
        n := n + 1;  
    } until (eoln());  
    media := suma / n;  
    print(media);  
}
```

```
34. algoritmo_34(){
    scan(día, mes);
    if ((día >= 21 and mes = 12) or (mes = 1) or (mes = 2) or (día < 21 and mes = 3)) {
        print("Invierno");
    } else if ((día >= 21 and mes = 3) or (mes = 4) or (mes = 5) or (día < 21 and mes = 6)) {
        print("Primavera");
    } else if ((día >= 21 and mes = 6) or (mes = 7) or (mes = 8) or (día < 21 and mes = 9)) {
        print("Verano");
    } else if ((día >= 21 and mes = 9) or (mes = 10) or (mes = 11) or (día < 21 and mes = 12)) {
        print("Otoño");
    }
}
```

35. El enunciado de este programa es ambiguo, estas son sus dos interpretaciones:

```
algoritmo_35{
    print("10 98 82 66 30 12");
}
algoritmo_35{
    print("11 99 83 67 31 13");
}
```

36. Precondición: true

Datos de salida: un string

Salida estándar: la palabra "Hola Mundo!"

Postcondición: imprime "Hola Mundo!"

37. Datos de entrada: un positivo: n

Precondición: true

Datos de salida: un natural: total

Postcondición: se obtiene la suma de todos los números del 1 al n

38. El enunciado de este programa es ambiguo, estas son sus dos interpretaciones:

Datos de entrada: dos enteros: a, b

Precondición: b /= 0

Datos de salida: dos enteros: q, r

Postcondición: a = b \* q + r ^ r < b

o

Datos de entrada: dos enteros: b, a

Precondición: a /= 0

Datos de salida: dos enteros: q, r

Postcondición: b = a \* q + r ^ r < a

39. Datos de entrada: una secuencia de caracteres

Entrada estándar: dicha secuencia

Precondición: true

Datos de salida: cinco enteros: a, e, i, o, u

Salida estándar: dichos enteros

Postcondición: imprime el nº de apariciones de a, e, i, o, u, en la secuencia, respectivamente

40. Datos de entrada: un real: t en horas

Precondición:  $t > 0$

Datos de salida: tres enteros: h, min, sg

Postcondición: imprime la hora t en horas h, minutos min y segundos sg

41. Datos de entrada: un entero: n

Precondición:  $n > 0$

Datos de salida: un booleano, p

Postcondición: devuelve true si y solo si n es primo, es decir, sus únicos divisores son 1 y n

42. Datos de entrada: un entero: n

Precondición:  $n > 0$

Datos de salida: un booleano, p

Postcondición: devuelve true si y solo si n es pseudoprimo, es decir, tiene otros dos divisores distintos de 1 y N y estos divisores son primos

43. El enunciado de este programa es ambiguo, estas son sus dos interpretaciones:

Datos de entrada: tres reales: a,b,c

Precondición:  $0 \leq a,b,c < 360$

Datos de salida: un booleano, t y un string, s

Postcondición: devuelve true si la suma de los ángulos  $a + b + c = 180$ , y si t es true, entonces s adquiere uno de estos valores:

- a, b, c < 90 -> s = "Acutángulo"
- Uno de los ángulos mide 90 -> s = "Rectángulo"
- Uno de los ángulos mide más de 90 -> s = "Obtusángulo"

o

Datos de entrada: tres reales: a,b,c

Precondición:  $0 \leq a,b,c < 2\pi$

Datos de salida: un booleano, t y un string, s

Postcondición: devuelve true si la suma de los ángulos  $a + b + c = \pi$ , y si t es true, entonces s adquiere uno de estos valores:

- a, b, c <  $\pi/2$  -> s = "Acutángulo"
- Uno de los ángulos mide  $\pi/2$  -> s = "Rectángulo"
- Uno de los ángulos mide más de  $\pi/2$  -> s = "Obtusángulo"

44. Datos de entrada: tres reales: a,b,c

Precondición: a,b,c > 0

Datos de salida: un booleano, t y un string, s

Postcondición: devuelve true si la suma de los dos lados menores es mayor que la longitud del lado mayor, y si t es true, entonces s adquiere uno de estos valores:

- Todos los lados iguales -> s = "Equilátero"
- Dos lados de tres son iguales -> s = "Isósceles"
- Todos sus lados son desiguales -> s = "Escaleno"

45. El enunciado de este programa es ambiguo, estas son sus dos interpretaciones:

Datos de entrada: un natural, 20 patos

Precondición: true

Datos de salida: dos enteros, patas y picos

Postcondición: si cada pato tiene un pico y dos patas, entonces se devolverán 20 picos y 40 patas

o

Datos de entrada: un natural, 2 patos

Precondición: true

Datos de salida: dos enteros, patas y picos

Postcondición: si cada pato tiene un pico y dos patas, entonces se devolverán 2 picos y 4 patas

46. Datos de entrada: tres reales: a, b, c

Entrada estándar: dichos reales

Precondición: true

Datos de salida: un real: mayor

Salida estándar: dicho real

Postcondición: mayor adquiere el valor mayor entre a, b, c

47. Datos de entrada: un entero: n

Precondición: n > 0

Datos de salida: un entero, fact

Postcondición: fact recibe el valor del factorial de n ( $n! = n * (n-1) * \dots * 2 * 1$ )

48. Precondición: true

Datos de salida: una secuencia de asteriscos

Salida estándar: dicha secuencia

Postcondición: una cadena de asteriscos de longitud igual al número de primos entre 1 y 100

49. Datos de entrada: un entero n

Precondición: true

Datos de salida: un entero cnt

Postcondición: cnt recibe el número de cifras que contiene n

50. Datos de entrada: una secuencia de caracteres

Entrada estándar: dicha secuencia

Precondición: todos los caracteres de la secuencia son letras minúsculas, salvo el último que es un espacio en blanco

Datos de salida: la secuencia transformada

Salida estándar: dicha secuencia

Postcondición: se devuelve dicha secuencia transformada , de forma que cada carácter se sustituye por el que esté situado tres posiciones a la derecha. El principio del abecedario se enrosca con el final de forma que dicho recorrido sea circular. El carácter espacio, el último, no se transforma

51. (\*)

Entrada	Descripción del caso	Salida
(5, 2)	Pares de igual signo, ambos números positivos	(2, 1)
(-5, -2)	Pares de igual signo, ambos números negativos	(1, -3)
(5, -2)	Pares de distinto signo, dividendo positivo, divisor negativo	(-4, -3)
(-5, 2)	Pares de distinto signo, dividendo negativo, divisor positivo	(-2, -1)
(0, 2)	Dividendo nulo, divisor positivo	(0, 0)

52. (\*)

Entrada	Descripción del caso	Salida
0.5	Radio dado entre 0 y 1	$0.25\pi$
1	Radio dado igual a 1	$\pi$
2	Radio dado mayor que uno	$4\pi$

53. (\*)

Entrada	Descripción del caso	Salida
(1, 0, -1)	Delta ( $\Delta$ ) es mayor que 0	(1, -1)
(1, 2, 1)	Delta ( $\Delta$ ) es igual a 0	(-1, -1)

54. (\*\*)

Entrada	Descripción del caso	Salida
1	Posición del día Lunes	L
2	Posición del día Martes	M
3	Posición del día Miércoles	X
4	Posición del día Jueves	J
5	Posición del día Viernes	V
6	Posición del día Sábado	S
7	Posición del día Domingo	D

55. (\*\*)

Entrada	Descripción del caso	Salida
(25, 12)	Días de diciembre a partir del 21	Invierno
(14, 2)	Días de enero y febrero	Invierno
(3, 3)	Días de marzo antes del 21	Invierno
(23, 3)	Días de marzo a partir del 21	Primavera
(6, 4)	Días de abril y mayo	Primavera
(2, 6)	Días de junio antes del 21	Primavera
(24, 6)	Días de junio a partir del 21	Verano
(7, 7)	Días de julio y agosto	Verano
(11, 9)	Días de septiembre antes del 21	Verano
(30, 9)	Días de septiembre a partir del 21	Otoño
(31, 10)	Días de octubre y noviembre	Otoño
(6, 12)	Días de diciembre antes del 21	Otoño

56. (\*\*)

Entrada	Descripción del caso	Salida
4.2	Nota entre 0 y 5	Suspenso
6.1	Nota entre 5 y 7	Aprobado
7.5	Nota entre 7 y 9	Notable
9.3	Nota entre 9 y 10	Sobresaliente

57. (\*\*)

Entrada	Descripción del caso	Salida
1	Hora exacta	(1, 0, 0)
1.5	Hora no exacta, pero el número sin la coma es múltiplo de 60	(1, 30, 0)
1.21	Resto de horas no exactas	(1, 12, 36)

58. (\*\*)

Entrada	Descripción del caso	Salida
6	El supremo es un número positivo menor que el m.c.m(3, 5) = 15	1
45	El supremo es mayor o igual a 15	1, 15, 30, 45

59. (\*\*\*)

Entrada	Descripción del caso	Salida
1	Unidad	False
17	Número primo	False

21	Número compuesto de dos factores primos	True
12	Número compuesto de más de dos factores primos	False

60. (\*\*\*)

Entrada	Descripción del caso	Salida
-1	Secuencia vacía	0
5, -1	Secuencia de un número	5
3, 4, -2, 1, -1	Secuencia de más de un número	6

61. (\*\*\*)

Entrada	Descripción del caso	Salida
(1, 1, 3)	La suma de los dos lados menores es menor o igual que la longitud del lado mayor	FALSE
-----	La suma de los dos lados menores es mayor que la longitud del lado mayor	TRUE
(3, 3, 3)	Tres lados iguales	Equilátero
(3, 3, 4)	Dos lados iguales entre sí y el tercero desigual	Isósceles
(6, 4, 3)	Tres lados desiguales	Escaleno

62. (\*\*\*\*)

Entrada	Descripción del caso	Salida
135	El nº no tiene dígitos pares	(0, 0)
2345	El número tiene al menos un dígito par pero no aparecen más de dos dígitos pares seguidos	(2, 1)
224568	El número tiene al menos un dígito par y hay una secuencia de al menos dos dígitos pares seguidos	(5, 3)