



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de programacion

Grupo: 3

No de Práctica(s): 5

Integrante(s): Velasco Gomez Noe Abimael

*No. de Equipo de
cómputo empleado:* 19

No. de Lista o Brigada: 3989

Semestre: 2020-1

Fecha de entrega: 16/09/19

Observaciones:

CALIFICACIÓN: _____

Pseudocódigo :

Objetivo: Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

La palabra pseudocódigo formada por el prefijo “*pseudo*”, que según el diccionario de la RAE (Real Academia Española) significa “*falso*”, de ahí parte para que también sea conocido como “falso lenguaje”.

Empezando por la importancia de el pseudocodigo, tenemos que decir que es una forma más breve de realizar que un diagrama de flujo, el cual realiza la misma función que realiza nuestro pseudocodigo, a pesar de que sea más breve de realizar no quiere decir que pierda el encanto y complejidad de un diagrama de flujo nuestro pseudocodigo; permitiéndonos así que podamos representar la solución a un algoritmo (problema) de la forma más detallada posible, utilizando acciones sucesivas.

1)Desarrollar pseudocódigo que reciba un número obtenga su factorial.

```
INICIO
X: INT
n: INT
r: INT
R: INT
READ X
IF X==0:
  _PRINT “La factorial es 1 de X”
ELSE:
  IF X>0:
    _n=1
    Do:
      _r=X - n
      _R=X * r
      _n=n + 1
    WHILE r>1
    _PRINT ”R es factorial de X”
  ENDIF
ELSE:
  _X<0
  _PRINT “X no tiene factorial”
ENDIF
FIN
```

*Verificar el algoritmo con los valores:

0
2
-4
5

a)0:

INICIO

X: INT

n: INT

r: INT

R: INT

READ 0

IF 0==0:

_PRINT "La factorial es 1 de X"

ELSE:

IF X>0:

_n=1

Do:

_r=X - n

_R=X * r

_n=n + 1

WHILE r>1

_PRINT "R es factorial de X"

ENDIF

ELSE:

_X<0

_PRINT "X no tiene factorial"

ENDIF

FIN

b) 2:

INICIO

X: INT

n: INT

r: INT

R: INT

READ 2

IF X==0:

_PRINT "La factorial es 1 de X"

ELSE:

IF 2>0:

_n=1

Do:

_r=2 - n

_R=2 * r

_n=n + 1

WHILE r>1

_PRINT "R es factorial de 5"

ENDIF

ELSE:

_X<0

_PRINT "X no tiene factorial"

ENDIF

FIN

c) -4:

INICIO

X: INT

n: INT

r: INT

R: INT

READ -4

IF X==0:

_PRINT "La factorial es 1 de X"

ELSE:

IF X>0:

_n=1

Do:

_r=X - n

_R=X * r

_n=n + 1

WHILE r>1

_PRINT "R es factorial de X"

ENDIF

ELSE:

-4<0

_PRINT "-4 no tiene factorial"

ENDIF

FIN

d) 5:

INICIO

X: INT

n: INT

r: INT

R: INT

READ 5

IF X==0:

_PRINT "La factorial es 1 de X"

ELSE:

IF 5>0:

_n=1

Do:

_r=5- n

_R=5 * r

_n=n + 1

WHILE r>1

_PRINT "R es factorial de 5"

ENDIF

ELSE:

_X<0

_PRINT "X no tiene factorial"

ENDIF

FIN

2) Desarrollar un pseudocódigo que lea 2 datos, nivel e ingreso. El programa debe:

Verificar que no se tiene un nivel mayor al ingreso (el ingreso debe ser mayor que la base)

Mostrar el impuesto a pagar

El porcentaje del impuesto se aplica a la diferencia entre el ingreso y la base. Y el impuesto total se calcula con la suma del resultado anterior más la cuota fija.

Para calcular impuestos, se hace a través de una tabla como la siguiente:

Nivel	Base (\$)	Cuota fija (\$)	Impuesto (%)
1	0.00	0.00	1.92
2	6,942.21	133.28	6.40
3	58,922.16	3,460.00	10.88
4	103,550.45	8,315.57	16.00
5	120,372.84	11,007.14	17.92
6	144,119.24	15,262.49	21.36
7	290,667.76	46,565.26	23.52
8	458,132.30	85,952.92	30.00
9	874,650.01	210,908.23	32.00
10	1,166,200.01	304,204.21	34.00
11	3,498,600.01	1,097,220.21	35.00

INICIO

nivel: N

ingreso:INT

base: INT

cuota fija: INT

impuesto p: INT

impuesto total: INT

READ nivel, ingreso

IF nivel < ingreso>base

IF nivel<=11

_impuesto p = ingreso – base

_impuesto total = impuesto p + cuota fija

_PRINT “ ya acabe de calcular el resultado del impuesto total”

ELSE:

_nivel>11

_nivel>ingreso

_ingreso<base

_PRINT “no se puede calcular el impuesto”

ENDIF

FIN

* Verificar el algoritmo con los pares:

a)(1,5000)

INICIO

nivel: N

ingreso:INT

base: INT

cuota fija: INT

impuesto p: INT

impuesto total: INT

READ 1, 5000

IF 1 < 5000>0

IF nivel<=11

_impuesto p = 5000 – 0

_impuesto total =5000+ 0

_PRINT “ ya acabe de calcular el resultado del impuesto total”

ELSE:

_nivel>11

_nivel>ingreso

_ingreso<base

_PRINT “no se puede calcular el impuesto”

ENDIF

FIN

b) (7,8000)

INICIO

nivel: N

ingreso:INT

base: INT

cuota fija: INT

impuesto p: INT

impuesto total: INT

READ 7, 8000

IF nivel < ingreso>base

IF nivel<=11

_impuesto p = ingreso – base

_impuesto total = impuesto p + cuota fija

_PRINT “ ya acabe de calcular el resultado del impuesto total”

ELSE:

_7>11

_7>8000

_8000< 290,667.76

_PRINT “no se puede calcular el impuesto”

ENDIF

FIN

c) (12,5000000)

INICIO

nivel: N

ingreso:INT

base: INT

cuota fija: INT

impuesto p: INT

impuesto total: INT

READ 12,5000000

IF nivel < ingreso>base

IF nivel<=11

_impuesto p = ingreso – base

_impuesto total = impuesto p + cuota fija

_PRINT “ ya acabe de calcular el resultado del impuesto total”

ELSE:

_12>11

_12>5000000

_5000000< no existe

_PRINT “no se puede calcular el impuesto”

ENDIF

FIN

En conclusión el usar un pseudocodigo como modo de soluciones algorítmicas es mucho más breve que un diagrama de flujo, como ya había establecido, sin embargo a diferencia del diagrama de flujo, en lo personal se dificulta encontrar un orden además de que la dificultad de hacer un pseudocodigo es un poco mayor, ya que en mi opinión se requiere un mayor análisis de las cosas, tanto para leerlos como para hacerlos.