

Buscar cursos



Mis cursos Institutos Computación P1 Laboratorio Letra Primera Tarea

## Letra Primera Tarea

# Laboratorio Programación 1. Primera Tarea 2019.

## Información general

Se sugiere leer con mucha atención todo el texto antes de comenzar la tarea. Es muy importante que se respeten todos los requisitos solicitados en esta sección y las siguientes. Si surgen dudas, pedimos que las formulen en el foro correspondiente a la tarea.

## Individualidad

Esta tarea se deberá realizar en forma individual. Para todas las tareas rige el Reglamento del Instituto de Computación ante Instancias de No Individualidad en los Laboratorios (lectura obligatoria).

## Calendario

**Entrega:** La tarea debe entregarse en el periodo que va del 03/10 al 07/10. Se aceptan entregas **hasta el lunes 07/10 a la hora 22:00**. Los trabajos deberán ser entregados dentro de los plazos establecidos. Los resultados de la entrega se publican el martes 08/10.

**Re-Entrega:** Todos los estudiantes que realizaron la entrega pueden corregir errores y realizar una segunda entrega (*re-entrega*). La re-entrega se realizará el día 10/10.

## Forma de entrega

Se debe entregar un **único** archivo de nombre **tarea1.pas** que debe contener el código solicitado. **No** se debe entregar el código binario o ejecutable de la tarea.

# Multiplicación Rusa

La *multiplicación rusa* es un método para calcular el producto de dos números utilizando solamente multiplicaciones y divisiones por el número 2.

Dados dos números *A* y *B* enteros **positivos** se construye la siguiente tabla:

Mitad	Doble
<i>A</i>	<i>B</i>
<i>A</i> <sub>1</sub>	<i>B</i> <sub>1</sub>
⋮	⋮
<i>A</i> <sub><i>k</i></sub>	<i>B</i> <sub><i>k</i></sub>

De tal manera que la columna de la izquierda comienza con *A* y cada número siguiente es la mitad *entera* del anterior:

$$A_{i+1} = A_i \text{ div } 2$$

Mientras que la columna derecha comienza con *B* y cada número siguiente es *el doble* del anterior:

$$B_{i+1} = B_i \times 2$$

La tabla termina cuando llegamos a *A<sub>k</sub>* = 1.

A partir de la tabla se calcula el resultado como la suma de todos los números de la segunda columna cuyo correspondiente de la primera columna es impar. Ese es el resultado de la multiplicación.

Ejemplo: Generamos la tabla para calcular 13 por 7:

Mitad	Doble	Paridad
13	7	impar
6	14	par
3	28	impar
1	56	impar

Las filas con un número par en la primera columna no se consideran para la suma. Por lo tanto el resultado es:

$$7 + 28 + 56 = 91$$

## Se pide

Escribir una función:

```
function MultiplicacionRusa(a,b: integer): integer;
```

que calcule la multiplicación de  y  usando el método de la multiplicación rusa y retorne su valor.

Esta función se considera en un contexto como el siguiente:

```

program PrincipalTarea1;
type
    EnteroPositivo = 1 .. MaxInt;
var
    num1,num2 : EnteroPositivo;

{ funciones aritméticas }
function doble(x : integer) : integer;
begin
    doble:= x + x
end;

function mitad(x : integer) : integer;
begin
    mitad:= x div 2
end;

function esImpar(n : integer) : boolean;
begin
    esImpar:= odd(n)
end;

{ funcion a escribir para la tarea }
function MultiplicacionRusa(a,b: integer): integer;
...
begin
{ cuerpo de la función a escribir }
...
end;

begin
    write('Ingrese números: ');
    readln(num1,num2);

    writeln('Resultado: ',MultiplicacionRusa(num1,num2))
end.

```

En el cuerpo de la función `MultiplicacionRusa`, no puede usarse ningún operador aritmético salvo el operador de suma (+). Sí se puede utilizar las funciones `doble`, `mitad` y `esImpar` que aparecen en el código anterior.

Por razones vinculadas a la corrección de esta tarea, no se pueden utilizar comentarios de la forma `(*...*)`; en su lugar pueden usarse los de la forma `{...}`.

Se debe entregar: un archivo de nombre `tarea1.pas` que contenga sólo el código de la función pedida, incluido su encabezado.

Para la corrección, se compilarán las tareas con la versión 3.0.4 de Free Pascal para Linux.

La compilación de la tarea se realizará de esta manera:

1. Se ubican en una misma carpeta: el archivo `tarea1.pas` junto con el archivo `PrincipalTarea1.pas` que se adjunta al final de este documento.
2. Se ejecuta el comando `fpc -Co -Cr -Miso -gl PrincipalTarea1.pas`

Es importante que se respeten las mayúsculas y minúsculas en los nombres de archivos.

**NO** se debe compilar con el IDE de Free Pascal. (ver instructivos de compilación)

No está permitido utilizar facilidades de Free Pascal que no forman parte del estándar y no se dan en el curso. Así por ejemplo, no se puede utilizar ninguna de las palabras siguientes: `uses`, `crt`, `readkey`, `longint`, `string`, `break`, etcétera.

En esta tarea, como en todos los problemas de este curso, se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera, se hará énfasis en buenas prácticas de programación que lleven a un código legible, bien documentado y mantenible, tales como:

- indentación adecuada
- utilización correcta y apropiada de las estructuras de control
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas
- nombres mnemotécnicos para variables, constantes, etcétera.

- 
- Descarga del archivo `PrincipalTarea1.pas` : <https://eva.fing.edu.uy/mod/resource/view.php?id=93814>

Última modificación: viernes, 27 de septiembre de 2019, 16:58

---

Por consultas sobre los cursos, dirigirse a los docentes o al instituto respectivo. Por consultas sobre el uso del sitio, contactarse con [moodle@fing.edu.uy](mailto:moodle@fing.edu.uy)