

Para los siguientes ejercicios diseñe los algoritmos o módulos en pseudocódigo, diagrama de flujo, diagrama NS y muestre el ambiente.

CE08A Hacer un módulo denominado Fact que tiene un parámetro formal X de tipo entero, en el cuál recibe un valor entero positivo y devuelve P que es la productoria de X, ver la fórmula en la columna a, luego hacer un algoritmo que calcule las fórmulas que se notan en la tabla según las columnas b y c de más abajo.

a	b	c
$P = \prod_{i=1}^X i$	$Y = \frac{m! + z!}{20 + t! - d!}$	$\binom{n}{k} = \frac{n!}{(n-k)! \times k!}$

CE08B Diseñar un algoritmo que ordene tres números A, B, C en forma ascendente utilizando un procedimiento denominado menorMayor que tiene dos parámetros pasados por referencia que devuelve en el primer parámetro el valor menor y en el segundo el valor mayor de los parámetros respectivamente.

CE08C Analizar, ejecutar y realizar la prueba de escritorio del siguiente algoritmo modular. Dibujar la jerarquía de los módulos mostrando las variables y parámetros.

<pre> <b>Funcion</b> U&lt;-PD (N)   <b>Definir</b> U,Aux <b>Como</b> Entero;   <b>Si</b> N&gt;=0 <b>Entonces</b>     Aux&lt;-N MOD 10;   <b>Sino</b>     Aux&lt;--1;   <b>FinSi</b>   U&lt;-Aux; <b>FinFuncion</b>  <b>SubProceso</b> DG (X <b>por valor</b> d0 <b>por</b> <b>Referencia</b> d1 <b>por Referencia</b> d2 <b>por</b> <b>Referencia</b> d3 <b>por Referencia</b>)   <b>Definir</b> i,D <b>Como</b> Entero;   <b>Para</b> i&lt;-0 <b>Hasta</b> 3 <b>Hacer</b>     D&lt;-PD(X);     X&lt;-trunc(X/10);     <b>Segun</b> i <b>Hacer</b>       0: d0&lt;-D;       1: d1&lt;-D;       2: d2&lt;-D;       3: d3&lt;-D;     <b>FinSegun</b>   <b>FinPara</b> <b>FinSubProceso</b> </pre>	<pre> <b>SubProceso</b> linea (N, Car)   <b>Definir</b> i <b>Como</b> Entero;   <b>Para</b> i&lt;-1 <b>Hasta</b> N <b>Hacer</b>     Escribir Car Sin Saltar;   <b>FinPara</b>   Escribir ''; <b>FinSubProceso</b>  <b>SubProceso</b> Bienvenida   linea(20, '*');   <b>Escribir</b> ' Bienvenidos';   <b>Escribir</b> ' i@i unju.fi';   linea(20, '*'); <b>FinSubProceso</b>  <b>Algoritmo</b> CE08C   <b>Definir</b> M,d0,d1,d2,d3 <b>Como</b> Entero;   <b>Definir</b> Op <b>Como</b> Caracter;   <b>Repetir</b>     <b>Borrar Pantalla</b>;     Bienvenida;     M&lt;-Aleatorio(1000,9999);     <b>Escribir</b> 'Numero generado al azar ',M;     DG(M,d0,d1,d2,d3);     <b>Escribir</b> d3,'um ',d2,'c ',d1,'d ',d0,'u';     linea(13,'-');     <b>Escribir</b> '¿Generar otro número s/n? ' Sin Saltar;     Leer Op;   <b>Hasta Que</b> Op='n' <b>FinAlgoritmo</b> </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1. Realizar un algoritmo que incluya los siguientes módulos:

- Módulo Menú que tiene un parámetro entero pasado por referencia que devuelve una opción.
- Módulo Bienvenida sin parámetros que muestra un mensaje de bienvenida.
- Módulo resto que acepta dos parámetros enteros y devuelve un número entero que es el resto de la división entera del primer parámetro sobre el segundo parámetro. No debe utilizar el operador MOD.
- Módulo primo que acepta un parámetro entero y devuelve un valor lógico verdadero si el parámetro es un número primo y falso si no lo es.
- Módulo NumeroACadena que acepta un parámetro entero número entre 0 y 10, y devuelve una cadena de caracteres que representa el número en letras.
- Módulo Intercambio que acepta dos parámetros pasados por referencia que intercambian los valores de los parámetros.

2. Analizar y ejecutar el algoritmo que calcula la serie de Taylor de la función seno de más abajo, dónde x es el valor de un ángulo (expresado en radianes) y n es el número de términos. Mostrar el esquema de los módulos según la programación modular, detalle los tipos de las variables y los alcances de las mismas (locales, globales). También hacer la prueba de escritorio para 45 ° y 4 términos.

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}, \forall x$$

```

SubProceso Bienvenida (N por valor)
  Escribir '-----'
  Escribir 'Hola ', N
  Escribir '-----'
  Escribir 'Este algoritmo calcula el seno '
  Escribir 'de un ángulo en grado sexagesimal'
  Escribir 'con la serie de Taylor hasta un '
  Escribir 'término determinado por Ud. '
  Escribir '-----'
FinSubProceso

SubProceso VerificaP(X por Referencia)
  Repetir
    Leer X
  Hasta Que X>=0
FinSubProceso

SubProceso VerificaMenor10(m10 por Referencia)
  Repetir
    Leer m10
  Hasta Que (m10>=0) Y (m10<=10)
FinSubProceso

SubProceso Fac <- Factorial ( X por valor )
  P <- 1
  Para i<-1 Hasta X Con Paso 1 Hacer
    P <- P * i
  FinPara
  Fac <- P
FinSubProceso

```

```

SubProceso Pot <- Potencia ( A por valor, B por valor )
  P <- 1
  Para i<-1 Hasta B Con Paso 1 Hacer
    P <- P * A
  FinPara
  Pot <- P
FinSubProceso

Proceso EJserie
  Escribir 'Ingrese su nombre'
  Leer Nombre
  Bienvenida(Nombre)
  Escribir 'Ingresar Ángulo en grados sexagesimales'
  VerificaP(A)
  X <- A*PI/180 //Radianes
  Escribir 'Ingresar N° de términos'
  VerificaMenor10(M)
  S <- 0
  Para N <- 0 Hasta M Con Paso 1 Hacer
    T <- 2*N+1
    S <- S + Potencia(-1,N)/Factorial(T)*Potencia(X,T)
  FinPara
  Escribir 'Seno calculado = ',S
  Escribir 'Seno verdadero = ',Sen(X)
FinProceso

```

3. Escribir un módulo denominado **PMS** que tiene dos parámetros formales B y A. Calcular B (base) elevado a A, siendo B un número real cualquiera y A un valor entero positivo o nulo. Hacerlo mediante multiplicaciones sucesivas de la Base. Si el cálculo no puede realizarse debe devolver cero (0).
4. Ingresar repetidamente el código (DNI) del estudiante (**tipo entero**) y dos notas Nota1 y Nota2 de los exámenes (**tipo real**) la finalización del ingreso de datos es a cuando se ingresa un el código igual a 999 (validar que las notas sean valores mayores o iguales a cero). Calcular el promedio de las notas, mediante un módulo denominado **promedioNotas**. Asignar la calificación de acuerdo a las notas de la tabla, hacer el algoritmo, usando un módulo denominado **calificacionNotas** que recibe un parámetro formal X de tipo real, que es la Nota promedio del estudiante y que devuelve una cadena de caracteres con la calificación correspondiente, como se nota en la tabla siguiente:

Calificación	Sobresaliente	Distinguido	Bueno	Suficiente	Insuficiente	Reprobado
Nota	[9.5,10]	[7.5, 9.5)	[5.5,7.5)	[3.5,5.5)	[0.5,3.5)	[0,0.5)

5. Proponer el enunciado de un algoritmo modular, que utilice por lo menos dos módulos. Realizar el análisis y el diseño del algoritmo propuesto.