



# FUNDAMENTOS DE COMPUTACIÓN

## ICM00794



"Impulsando la sociedad del Conocimiento"

[Principal] [Curso] [Material] [Tareas] [Exámenes] [Proyectos] [Políticas] [Soluciones]

### 1ra Evaluación II Término 2008 - 2009. Diciembre 09 , 2008

#### Tema 1

<b>Tema:</b>	<b>Lazos de Repetición</b>	
<b>Complemento:</b>	Cocientes y Residuos	
<b>Revisión de Temas Relacionados:</b>	Parcial I Término 2005 - 2006. Tema 3	Convertir de Binario a Decimal
	Parcial II Término 2003 - 2004. Tema 1	Cambio de Base
<b>Aproximación a Solución</b>	Realizar la conversión mediante: - Extraer cada dígito del número - Ponderar el valor de cada dígito por su peso en base Octal	Se supone que el número ingresado será octal. Por lo que no se valida
<b>TAREA</b>	Validar que el número ingresado sea OCTAL	
	Realizar una Función del algoritmo	Para segunda Evaluación

#### Propuesta de solución:

Inicio	% Conversión de OCTAL a Decimal	
Ingreso	n=input('lectura del odometro defectuoso:');	se supone un numero ingresado OCTAL
Procedimiento	s=0; i=0; while ~(n<=0) r=mod(n,10); n=fix(n/10); s=s+r*(8^i); i=i+1; end	extraer dígitos hasta que no quede ninguno  Ponderar los valores de los dígitos de octal a decimal
Salida	disp(s);	
Fin		

#### Tema 2

<b>Tema:</b>	<b>Arreglos una dimensión</b>	
<b>Complemento:</b>	Aleatorios no repetidos	
<b>Revisión de Temas</b>	Mejoramiento I Término 2005 - 2006. Tema 4.	Pozo millonario

<b>Relacionados:</b>	2da Evaluación I Término 2008-2009. Septiembre 2, 2008	Matriz de Juego Memotest
<b>Aproximación a Solución</b>	Se utiliza un arreglo de banderas de sorteado con ceros (no seleccionados)	Arreglo de banderas
	Se marcan con 1 los que salen seleccionados aleatoriamente, validando que sean no repetidos	la selección es mediante el puntero
<b>TAREA</b>	Realizar selecciones no repetidas de 20 números entre 1 y 1000	El arreglo de banderas no resulta practico. Se validaría con respecto a los números que ya salieron en sorteos previos.

### Propuesta de solución

Inicio	%sorteo de jueces de CSJ	
Ingreso		No hay ingreso dado que la cantidad de jueces es fijada por la Nueva Constitución 2008
Procedimiento	<pre>n=31; m=21; i=1; while ~(i&gt;n)     sorteado(i)=0;     i=i+1; end i=1; while ~(i&gt;m)     a=fix(rand*n)+1;     while ~(sorteado(a) ==0)         a=fix(rand*n)+1;     end     sorteado(a)=1;     i=i+1; end</pre>	<p>Se inicializa el arreglo de sorteados con cero.</p> <p>Se sortea el juez y se marca con 1 solo si no ha sido favorecido anteriormente. (Numero sorteado no repetido)</p>
Salida	<pre>i=1; while ~(i&gt;n)     if sorteado(i)==1         disp(i);     end     i=i+1; end</pre>	Se muestra la tabla de solo números de sorteado que tienen como marca 1 (seleccionados)
Fin		

### Tema 3

<b>Tema:</b>	<b>Arreglos</b>	
<b>Complemento:</b>	Algoritmo de mayor	
<b>Revisión de Temas Relacionados:</b>	Parcial II Término 2005 - 2006. Tema 3	Tiempos diarios de un atleta en entrenamiento

<b>Aproximación a Solución</b>	Ingresar en arreglos los tiempos	diferenciar las unidades de horas y minutos
	convertir a la misma unidad para poder encontrar las diferencias de tiempo transcurrido	
<b>TAREA</b>	Mostrar el tiempo total transcurrido en horas y minutos	

### Propuesta de Solución

Inicio	% Recorrido Metrovia	
Ingreso	<pre>n=input('numero de estaciones: '); i=1; while ~(i&gt;n);     fprintf('Estacion %d \n',i);     hora(i)=input('Hora: ');     minuto(i)=input('minuto:');     i=i+1; end</pre>	Se ingresa en arreglos de hora y minuto cada uno de los datos de arriba a cada estación
Procedimiento	<pre>i=1; while ~(i&gt;n);     Tmin(i)=hora(i)*60+minuto(i);     i=i+1; end tiempo(1)=0; s=0; i=2; while ~(i&gt;n);     tiempo(i)=Tmin(i)-Tmin(i-1);     s=s+tiempo(i);     i=i+1; end mayor=1; i=1; while ~(i&gt;n);     if tiempo(mayor)&lt;tiempo(i)         mayor=i;     end     i=i+1; end</pre>	<p>Se convierten los tiempos a unidades de minutos, respecto al inicio del día.</p> <p>Con lo que se puede diferenciar los minutos transcurridos.</p> <p>Se busca el puntero del tramo de mayor recorrido, algoritmo del mayor.</p>
Salida	<pre>fprintf('Mayor tiempo entre estacion %d y %d \n',mayor-1,mayor); fprintf('Tiempo total del recorrido %f \n',s);</pre>	Muestra los resultados.
Fin		

### Tema 4

<b>Tema:</b>	<b>Arreglos</b>	
<b>Complemento:</b>	Pitágoras y acumuladores.	

<b>Revisión de Temas Relacionados:</b>	Mejoramiento I Término 2005 - 2006. Tema 3	Calculo de millas de viajero frecuente. Viaje con retorno al punto inicial
	Parcial I Término 2003 - 2004. Tema 4	Determinar el lado mayor del polígono
<b>Aproximación a Solución</b>	ingresar las coordenadas de los puntos en arreglos x(i) - y(i)	
	referenciar siempre al primer punto	
<b>TAREA</b>	Encontrar el triángulo de mayor área.	Prueba de escritorio

### Solución Propuesta

Inicio	% Area de poligono	
Ingreso	<pre>n=input('numero de lados: '); i=1; while ~(i&gt;n)     fprintf('Punto %d \n',i);     x(i)=input('coordenada x:');     y(i)=input('coordenada y:');     i=i+1; end</pre>	<p>Ingreso de puntos del polígono.</p> <p>La cantidad de puntos es la misma que la de lados.</p>
Procedimiento	<pre>i=2; total=0; while ~(i&gt;=n)     a=sqrt((x(1)-x(i))^2+(y(1)-y(i))^2);     b=sqrt((x(1)-x(i+1))^2+(y(1)-y(i+1))^2);     c=sqrt((x(i)-x(i+1))^2+(y(i)-y(i+1))^2);     s=(a+b+c)/2;     area=sqrt(s*(s-a)*(s-b)*(s-c));     total=total+area;     i=i+1; end</pre>	<p>Siempre referenciado al primer punto para formar los triángulos.</p> <p>Revisar formulas de Pitágoras en matlab.</p> <p>Acumular las áreas de los triángulos para el total del polígono.</p>
Salida	disp(total);	
Fin		

Actualizado: 20/01/2009  
Revisión: 7

©2008 [ICM-ESPOL](http://www.icm-espol.edu.ec)

Comentarios:  
[edelros@goliat.espol.edu.ec](mailto:edelros@goliat.espol.edu.ec)