

CE10A Mediante un menú de opciones realizar el siguiente programa modular:

- Cargar un vector hasta que el operador no quiera ingresar más datos, validar que se ingresen números enteros impares.
- Mostrar un mensaje que muestre si los datos del vector están repetidos o no.
- Contar y mostrar cuántos valores K hay en el vector.
- Calcular y mostrar el promedio de los datos del vector.
- Mostrar a opción del operador: la posición de los datos y los datos que superan el promedio o aquellos que no lo hacen.

CE10B Hacer un módulo que cargue un vector con N valores enteros al azar pertenecientes al intervalo [100,999] que no estén repetidos, validados por el módulo.

CE10C Analizar los programas en Pascal y C del cálculo del máximo común divisor; luego ejecute los programas en la página de <http://ideone.com/>

pseudocódigo	Pascal	C
<u>algoritmo</u> MaxComDiv <u>variable</u> entero: A,B <u>inicio</u> leer A,B <u>mientras</u> A<>B <u>hacer</u> <u>si</u> A>B <u>entonces</u> A ← A-B <u>sino</u> B ← B-A <u>fin</u> <u>finmientras</u> <u>escribir</u> 'MCD=',A <u>fin</u>	program MaxComDiv; var A,B:integer; begin readln(A,B); while A<>B do if A>B then A:=A-B else B:=B-A; writeln('MCD=',A); end.	int main() { int a,b; scanf("%d" "%d",&a,&b); while (a!=b){ if (a>b) a=a-b; else b=b-a; } printf("MCD %d\n", a); return 0; }
pseudocódigo (modular)	Pascal (modular)	C (modular)
<u>funcion</u> MCD(x,y):entero [parametros x,y:entero p/v] <u>inicio</u> <u>mientras</u> x<>y <u>hacer</u> <u>si</u> x>y <u>entonces</u> x ← x-y <u>sino</u> y ← y-x <u>fin</u> <u>finmientras</u> MCD ← x <u>finfuncion</u> <u>algoritmo</u> MaxComDiv <u>variable</u> entero: A,B <u>inicio</u> leer A,B <u>escribir</u> 'M.C.D.=' ,MCD(A,B) <u>fin</u>	program MaxComDiv; var A,B:integer; function MCD(x,y:integer):integer; begin while x<>y do if x>y then x:=x-y else y:=y-x; MCD:=x; end; begin readln(A,B); writeln('M.C.D.=' ,MCD(A,B)); end.	#include <stdio.h> int main() { int a,b; int mcd (int a, int b); scanf("%d" "%d",&a,&b); printf("MCD %d\n", mcd(a,b)); return 0; } int mcd (int a, int b) { while (a!=b){ if (a>b) a=a-b; else b=b-a; } return a; }

- Cargar un vector con N números reales y, a pedido del usuario: mostrar los números que superan al promedio y su posición, verificar si los elementos del vector se hallan ordenados en forma descendente, muestre el mínimo con su posición.
- Realizar un programa que permita cargar los resultados de un examen, para los objetivos que se buscan NO es necesario ingresar el nombre ni los demás datos de los alumnos. El programa debe realizar, mediante un menú de opciones, las siguientes operaciones:

- a. Cargar las notas hasta que el operador no quiera ingresar más datos, validar que las notas estén en el intervalo [1, 10]. Si el valor ingresado NO es válido mostrar un mensaje de error y volver a solicitarlo hasta que el valor sea correcto (utilice un sub algoritmo para esta validación).
- b. Mostrar las notas y el orden en que fueron ingresadas.
- c. Mostrar el Promedio de las notas ingresadas.
- d. Mostrar las notas que superan al promedio y su posición.
- e. Calcular cual fue la nota máxima obtenida y mostrar cuantos alumnos la obtuvieron.
- f. Solicitar una nota y mostrar cuantos alumnos la obtuvieron.
- g. Mostrar el porcentaje de alumnos que obtuvieron la nota máxima y el porcentaje de alumnos que obtuvieron la nota mínima.
- h. Salir

Consideración: para poder realizar cualquiera de las opciones (entre la opción 2 a la 7), debe verificar que el vector se encuentre cargado, en caso contrario mostrar el mensaje "Debe cargar las notas antes de realizar esta operación". También debe limpiar la pantalla cada vez que se termine de procesar una opción.

3. Realizar un programa que genere aleatoriamente 20 números enteros y los guarde en un vector, luego realizar lo siguiente:
 - a. Mostrar los primeros N valores del vector. Validar que N sea válido.
 - b. Mostrar la suma de los valores pares.
 - c. Buscar un número dado y mostrar su posición, en caso de no existir mostrar un mensaje de error.
 - d. Verificar si el vector se encuentra ordenado en forma ascendente.
4. Proponer el enunciado de un algoritmo modular, que utilice la estructura de datos: arreglos unidimensionales. Realizar el análisis y el diseño del algoritmo propuesto.