Ejercicio 1) Lotería
Un apostador necesita procesar los 20 premios de la lotería, para luego conocer el promedio de todos los impares, el mayor, menor y la cantidad de los números pares que han salido sorteados.
Desarrolle una aplicación que permita ingresar los 20 números y presente los resultados.

Nº	NOCTURNA ENTRE RIOS (21:00 hs.)		
1	3539	11	3497
2	1463	12	6720
3	1471	13	4044
4	3936	14	9985
5	3154	15	9266
6	8740	16	6642
7	3730	17	0579
8	2342	18	3446
9	5528	19	2744
10	9731	20	7476

ANALISIS

entradas premios

salidas promedio de todos los impares (promimp)
el mayor (meyor)
menor (menor)
la cantidad de los números pares que han salido sorteados (contpares)

procesos promedio <- acumimpares/contimpares
mayor (pertenece al conjunto de los premios ingresados)
menor (pertenece al conjunto de los premios ingresados)
contpares<- (conjunto de numeros pares de los ingresados)

AMBIENTE

nombre	tipo	descripcion
premio	entero	premios sorteados
mayor		numero mayor sorteado
menor		numero menor sorteado
contpares		contador de num pares
conimpres		contador de num impares
acumimpares		acumulador de impares
promedio	real	promedio de impares

ESTRATEGIA

inicializar acumladores y contadores
iterar*
ingresar numero del premio(azar)
mostrar numeros ingresados
verificar mayorº
actualizar mayorº
/º
verificar menorº
actualizar menorº
/º
verificar imparesº
actualizar contador y acumulador de imparº
actualizar contador de los paresº

calcular el promedio de los impares
mostrar resultadosemi

ALGORITMO

```
Proceso loteria
        Definir num,mayor,menor,contpares,contimpares,acumimpares,i Como Entero;
        Definir promedio Como Real;
        contimpares <- 0;
        contpares <- 0;
        acumimpares <- 0;
        mayor <- 0;
        menor <- 100000;
        Para i<-1 Hasta 20 Hacer
                num <- azar(9999);
                Escribir, num;
                si num>mayor Entonces
                        mayor <- num;
                FinSi
                si num<menor Entonces
                        menor <- num;
                FinSi
                si num mod 2 == 0 Entonces

                        contpares <- (contimpares + 1);
                sino
                        acumimpares <- (acumimpares + num);
                        contimpares <- (contimpares +1);
                FinSi
        FinPara

        promedio <- (acumimpares/contimpares);
        Escribir "el promedio de todos los impares es : ", promedio;
        Escribir "el mayor de los premios es : ", mayor;
        Escribir "el menor de todos los premio es : ", menor;
        Escribir "la cantidad de premios pares es : ", contpares;

FinProceso
```

VERIFICACION

Ejercicio 2) Disc Jockey
Un Disc Jockey requiere de un proceso que

a) le permita ingresar una lista de canciones con Nombre y duración en MM:SS, informando al final el tiempo total en HH:MM:SS y La canción con mayor y menor duración
b) Modifique el ejercicio anterior para que corte en forma automática la carga en el caso que supere los 74 Minutos

ANALISIS

entradas nombre, duracion (minutos, segundos)

salidas tiempo total (hs, min, seg), mayor, menor

procesos mayor<- (pertenece al conjunto de las canciones ingresadas)
menor<- (pertenece al conijunto de las canciones ingresadas)
hs <- seg / 3600
min <- (seg mod 3600) / 60
seg<- (seg mod 3600) mod 60

AMBIENTE

variable	tipo	variable
nombre	caracter	nombre de la cancion
minutos	entero	minutos de la cancion entrante
acumsegundos	entero	segundos de la cancion entrante
hs	entero	horas de la lista de musica
min	entero	minutos de la lista de musica
acummseg	entero	segundos de la lista de musica
mayor	entero	cancion de mayor duracion
menor	entero	cancion de menor duracion
canmay	caracter	cancion de mayor duracion
canmen	caracter	cancion de menor duracion
cortador	logico	valor que seusa de bandera

cortante	entero	valor que usare para cambiar en una condicion el valor de cortador a verdadero

ESTRATEGIA

inicializar mayores y menores y acumuladores
iterar*
    ingresar nombre de la cancion
    ingresar minutos y segundos de la cancion
    convertir la duracion de la cancion a segundos
    verificar mayorº
            actualizar mayorº
             /º
    verificar menorº
            actualizar menorº
             /º
    acumular cancion en segundos

calcular horas
calcular mintuos
calcular segundos
mostrar en pantalla

ALGORITMO

```
Proceso discjokey
        Definir nombre,canmay,canmen Como Caracter;
        Definir cortante, minutos,acumsegundos,hs,min,seg, acumseg,mayor,menor
Como Entero;
        Definir cortador Como Logico;
        cortador <- falso;
        mayor <- 0;
        menor <- 50000;
        acumseg <- 0;
        acumsegundos <- 0;
        Repetir
                Escribir "ingresar nombre de la cancion";
                Leer nombre;
                Escribir "ingresar minutos y segundos de la cancion";
                leer minutos;
                leer acumsegundos;

                acumsegundos <- acumsegundos + (minutos * 60);

                si mayor>acumsegundos Entonces
                        mayor <- acumsegundos;
                        canmay <- nombre;
                FinSi
                si menor<acumsegundos Entonces
                        menor<- acumsegundos;
                        canmen<- nombre;
```

```
                FinSi

                acumseg <- (acumseg + acumsegundos);

                Escribir "¿desea seguir ingresando canciones? pulse 1 para SI y 2 para
NO";
                leer cortante;

                si cortante <> 2 Entonces
                        escribir " ";
                sino
                        cortador <- Verdadero;
                FinSi

        Hasta Que cortador

        hs <- trunc(acumseg / 3600) ;
        min<-trunc((acumseg mod 3600)/60);
        seg <- trunc((acumseg mod 36000) mod 60);

        Escribir "la cancion mas larga es ", canmay, " con una duracion de ", mayor, "
segundos";
        Escribir "la cancion mas corta es ", canmen, " con una duracion de ", menor, "
segundos";
        Escribir "la lista de reproduccion tiene una duracion total de ", hs,":",min,":",seg;


FinProceso
```

B) Proceso discjokey

```
        Definir nombre,canmay,canmen Como Caracter;
        Definir cortante, minutos,acumsegundos,hs,min,seg, acumseg,mayor,menor
Como Entero;
        Definir cortador Como Logico;
        cortador <- falso;
        acumseg <- 0;
        acumsegundos <- 0;
        Repetir
                Escribir "ingresar nombre de la cancion";
                Leer nombre;
                Escribir "ingresar minutos y segundos de la cancion";
                leer minutos;
                leer acumsegundos;

                acumsegundos <- acumsegundos + (minutos * 60);

                si mayor<acumsegundos Entonces
                        mayor <- acumsegundos;
                        canmay <- nombre;
                FinSi
                si menor>acumsegundos Entonces
                        menor<- acumsegundos;
                        canmen<- nombre;
                FinSi

                acumseg <- (acumseg + acumsegundos);
```

```
                cortante <- trunc(acumseg/60);

                si cortante < 74 Entonces
                        escribir " ";
                sino
                        cortador <- Verdadero;
                FinSi

        Hasta Que cortador

        hs <- trunc(acumseg / 3600) ;
        min<-trunc((acumseg mod 3600)/60);
        seg <- trunc((acumseg mod 36000) mod 60);

        Escribir "la cancion mas larga es ", canmay, " con una duracion de ", mayor, " segundos";
        Escribir "la cancion mas corta es ", canmen, " con una duracion de ", menor, " segundos";
        Escribir "la lista de reproduccion tiene una duracion total de ", hs,":",min,":",seg;


FinProceso
```

VERIFICACION

Ejercicio 3) Tornillos

Una fábrica de tornillos realiza el control de calidad de su producción evaluando 10 productos de cada lote.
Al iniciar cada lote se ingresa el número de código, la medida esperada y la medición de los 10 elementos tomados al azar. Al finalizar la carga debe informar el mayor error absoluto y el porcentaje de productos con fallas. Al terminar de procesar todos los lotes (ingresando el número de código 0) debe informar:

- -Cantidad de lotes procesados
- -% total de fallas
- -Lote con menor cantidad de fallas
- -Lote con mayor cantidad de fallas.

ANALISIS

entradas codigo, medesp, **medevaluada** 10 elementos al azar(usar un para)

salidas al finalizar la carga informar: mayor error absoluto (**mayerror**),
porcentaje de productos con fallas (**porcentajefalla**)

al finalizar los lotes informar: cantidad de lotes procesados **(lotesproc**),
porcentaje de fallas (**porcfallastotales**),
lote con mas fallas (**masfallado**),
lote con menos fallas (**menosfallado**)

procesos mayerror <- **error** <- abs(medevaluada - medesp)
porcentajefalla <- **contfallas***10

lotesproc <- lotesproc + 1
porcfallastotales <- **acumporcfalla** / lotesproc
masfallado<- contfalla, codigo
menosfallado <- contfalla, codigo

AMBIENTE

nombe	**tipo**	**descripcion**
codigo	**entero**	codigo del lote
medesp		medida esperada de tornillo en el lote
medevaluada		medida de tornillo tomado para muestra
mayerror		**mayor error o diferencia entre el esperado y evaluado**
porcentajefalla		**porcentaje de falla por lote**
pocfallastotales	**entero**	**porcentajje de falla de todos los lotes**
masfallado		**lote con mas fallas**
menosfallado		**lote con menos fallas**
contfallas		**contador de fallas por lote**
acumporcfalla		**acumulador de porcentaje de fallas de cada lote**
error		**error o diferencia entre el esperado y evaluado**
lotesproc		**cantidadde lotes procesados**
codmasfallado	**entero**	**codigo del lote mas fallado**
codmenosfallado	**entero**	**codigo del lote menos fallado**
i	**entero**	**contador del bucle**

ESTRATEGIA

inicialiar variables
ingresar el codigo del lote

iterar *

contador de lotes + 1
ingresar la medida esperada

iterar con para*
ingresar medida a evaluar
verificar si hay errorº
no hay errorº
contador de fallas + 1º
calcular error
verificar mayor errorº
actualizar mayor errorº
/º
reestablecer error en 0
finPara

calcular y mostrar porcentaje de fallas
mostrar mayor error absoluto (con el codigo)
acumular los porcentajes de fallas
verificar lote con mas fallasº
actualizar lote con mas fallasº
/ º
verificar lote con menos fallasº
actualizar lote con menos fallasº
/º
restablecer variables (mayerror, error, porcentajefalla,contfallas)
ingresar nuevo codigo de lote o 0 para terminar

finMientras

mostrar Cantidad de lotes procesados, porcentaje total de fallas, Lote con menor cantidad de fallas, Lote con mayor cantidad de fallas.

ALGORITMO

Proceso tornillos
Definir codmasfallado, codmenosfallado,codigo,medesp,medevaluada,mayerror,error,porcentajefalla,contfallas Como Entero;

```
Definir porcfallastotales,masfallado,menosfallado,acumporcfallas,lotesproc Como Entero;

contfallas <- 0;
acumporcfallas <- 0;

Escribir 'ingrese el codigo del lote';
Leer codigo;

Mientras codigo<>0 Hacer
        lotesproc <- lotesproc+1;

        Escribir 'ingresar la medida esperada en centimetros';
        Leer medesp;

        Para i<-1 Hasta 10 Hacer
                medevaluada <- azar(10);
                si medesp <> medevaluada Entonces
                        contfallas <- contfallas + 1;
                        error <- abs(medevaluada - medesp);
                        si mayerror<error Entonces
                                mayerror <- error;
                        FinSi
                FinSi
        error <- 0;

FinPara

porcentajefalla <- contfallas*10;

Escribir "el lote ", codigo, "tiene como mayor error ",mayerror," cm y un porcentaje de falla del ",porcentajefalla, "%";

        acumporcfallas <- acumporcfallas + porcentajefalla;

        si masfallado < contfallas Entonces
                masfallado <- contfallas;
                codmasfallado <- codigo;
        FinSi
        si menosfallado>contfallas Entonces
                menosfallado<- contfallas;
                codmenosfallado<- codigo;
        FinSi

        mayerror <- 0;
        porcentajefalla <- 0;
        contfallas <- 0;

        Escribir "ingrese un nuevo codigo de lote para procesar o 0(CERO) para finalizar el procesamiento";
        leer codigo;

FinMientras

Escribir "la cantidad de lotes procesados es: ", lotesproc;
porcfallastotales <- trunc(acumporcfallas/lotesproc);
```

Escribir "la cantidad de fallas totales de los lotes procesados es del ", porcfallastotales, "%";
Escribir "el lote con mayor cantidad de fallas es ", codmasfallado, " con ",masfallado, " fallas";
Escribir "el lote con mayor cantidad de fallas es ", codmenosfallado, " con ", menosfallado, " fallas";

FinProceso

VERIFICACION

Ejercicio 4)
Una forma de determinar si un número es primo consiste en verificar si es divisible por sí mismo y por uno.
Implemente una aplicación que determine si un valor entero ingresado por el usuario es o no un número primo.

ANALISIS

entradas numero

salidas informar si el numero ingresado es primo o no

procesos si contmods == 0 entonces el numero es primo, sino, no es primo

AMBIENTE

nombre	**tipo**	**descripcion**
num	**entero**	**numero que se ingresa**
i		**divisor**
contmods		**contador de modulos distintos de 0**

ESTRATEGIA

ingresar numero
verificar que no sea 1º
informar que no es primoº
iterar*
dividir el numero
analizar moduloº
incrementar contador de moduloº
/º
finpara
analizar contador de modulosº
informar "el numero es primo"
informar "el numero no es primo"

ALGORITMO

```
Proceso primos
	definir num, i, contmods Como Entero;
	Escribir "ingrese un numero ";
	leer num;

	si num == 1 Entonces
		Escribir "no es primo";
	SiNo
		para i<-2 Hasta num/2 Con Paso 1 Hacer
			si num mod i == 0 Entonces
				contmods <- contmods+1;
			FinSi

		FinPara

		si contmods == 0 Entonces
			escribir "es primo ";
		SiNo
			Escribir " no es primo";
		FinSi

	FinSi


FinProceso
```

VERIFICACION

Ejercicio 5)
Considerando el ejercicio anterior realice un proceso que muestre en pantalla todos los números primos que pertenezcan a un rango ingresado por el usuario. Considere cualquier orden de ingreso de los valores del rango.

ANALISIS

entradas inicio, final

salidas informar en pantalla los numeros primos que se encuentran en el rango de valores establecido

procesos si contmods == 0 entonces el numero es primo, sino, no es primo

AMBIENTE

nombre	**tipo**	**descripcion**
corriente	**entero**	**numero que se evalua**
i		**divisor**

contmods		**contador de modulos distintos de 0**
inicio		**rango en el que inicia el valor**
final		**rango en el que finaliza el valor**

ESTRATEGIA

ingresar el rango de valores
iterar con para*

verificar que no sea 1º
/
iterar*
dividir el numero
analizar moduloº
incrementar contador de moduloº
/º
finpara
analizar contador de modulosº
informar “el numero es primo”º
/º

ALGORITMO

```
Proceso primos
	Definir inicio,final,i,corriente,contmods Como Entero;
	Escribir 'ingrese el rango de valores';
	Escribir 'desde ';
	Leer inicio;
	Escribir 'hasta';
	Leer final;
	Para corriente<-inicio Hasta final Hacer
		Si corriente<>1 Entonces
			Para i<-2 Hasta corriente/2 Hacer
				Si corriente MOD i==0 Entonces
					contmods <- contmods+1;
				FinSi
			FinPara
			Si contmods==0 Entonces
				Escribir"", corriente;
			FinSi
		FinSi
		contmods<- 0;
	FinPara
FinProceso
```

VERIFICACION

Ejercicio 6)
Se ingresa una lista de notas correspondientes a una evaluación de programación numeradas entre 0 y 10. Al finalizar se debe mostrar en pantalla:

- Cantidad de notas
- Promedio
- Cantidad de aprobados y no aprobados
- Porcentaje de alumnos con:
  - Muy Bueno (8 o más)
  - Bueno (6 o 7)
  - Regular (4 o 5)
  - Insuficiente (3 o menos)

ANALISIS

entradas nota

salidas contnotas, prom, aprobados, desaprobados, porcmb, porcb, porcr, porci

procesos contnotas<- contnotas+1
prom<- acumnotas/contnotas
aprobados<- nota=> 6
desaprobados <- nota<6
porcmb <- contmb*10
porb<- contb*10
porcr<- contr*10
porci<- conti*10

AMBIENTE

nombre	tipo	descripc ion
nota	entero	nota que del parcial
contnota		contador de notas ingresadas
prom	real	promedio de todas las notas ingresadas
aprobados	entero	notas aprobadas
desaprobados		notas desaprobadas
porcmb		porcentaje de muy bueno
porcb		porcentaje de bueno
porcr		porcentaje de regualar
porci		porcentaje de irregular

acumnotas		acumulador de notas
contmb		contador de muy buenos
contb		buenos
contr		regular
conti		irregular

Se ingresa una lista de notas correspondientes a una evaluación de programación numeradas entre 0 y 10. Al finalizar se debe mostrar en pantalla:

- Cantidad de notas
- Promedio
- Cantidad de aprobados y no aprobados
- Porcentaje de alumnos con:
  - Muy Bueno (8 o más)
  - Bueno (6 o 7)
  - Regular (4 o 5)
  - Insuficiente (3 o menos)

ESTRATEGIA

inicializar variables
ingresar la nota

iterar*
contador de notas ++
actualizo el acumulador de las notas
verifico si es aprueba^{o}
aprobado ++^{o}
desaprobado++^{o}
segun nota^{o}
8,9,10: actualizar contador
6,7: actualizar contador
4,5:actualizar contador
1,2,3: actualizar contador
ingresar nueva nota
finmietras
calcular y mostrar promedio
mostrar cantidad de notas, aprobados y no, porcentaje de notas mb, b, r, i

ALGORITMO

```
Proceso notas
        definir nota, contnota,aprobados,desaprobados, acumnotas Como Entero;
        definir contmb,contb,contr,conti Como Entero;
        definir porcmb,porcb,porcr,porci,prom Como Real;

        contnota <- 0;
        acumnotas<-0;

        Escribir "ingrese una nota";
        leer nota;
```

```
Mientras nota<> -1 Hacer

        contnota<-contnota+1;
        acumnotas<- acumnotas+nota;

        si nota >= 6 Entonces
                aprobados <- aprobados+1;
        SiNo
                desaprobados<- desaprobados + 1;
        FinSi
        segun nota Hacer
                8,9,10: contmb <- contmb +1;
                6,7: contb<- contb+1;
                4,5:contr<- contr+1;
                1,2,3:conti<- conti+1;
        FinSegun

        Escribir "ingresar una nueva nota o -1 para finalizar el ingreso";
        leer nota;
FinMientras

Escribir "la cantidad de notas ingresadas fueron: ", contnota;
prom<- acumnotas/contnota;
Escribir "el promedio de todas las notas ingresadas es : ", prom;
Escribir "la cantidad de personas que aprobaron el parcial son: ", aprobados;
Escribir "la cantidad de personas que desaprobaron el parcial son: ",
desaprobados;
porcmb<- (100/contnota)*contmb;
porcb<- (100/contnota)*contb;
porcr<- (100/contnota)*contr;
porci<- (100/contnota)*conti;

Escribir "el porcentaje de notas muy buenas es del ", porcmb,"%";
Escribir "el porcentaje de notas buenas es del ", porcb,"%";
Escribir "el porcentaje de notas regulares es del ", porcr,"%";
Escribir "el porcentaje de notas irregulares es del ", porci,"%";

FinProceso
```

VERIFICACION

Ejercicio 7)
Implemente una pequeña aplicación que permita calcular el total a pagar por una compra ingresando la cantidad y el precio unitario de cada producto. Debe informar además la cantidad de productos adquiridos.
ANALISIS

entradas producto, precio, cantidad

salidas apagar, adquiridos

procesos apagar <- apagar+precio
adquiridos<- adquiridos + cantidad

AMBIENTE

nombre	tipo	descripcion
producto	caracter	nombre del producto
precio	entero	precio del producto
cantidad		cantidad de cada producto
apagar		precio final del la compra
adquiridos		cantidad de productos adquiridos

Implemente una pequeña aplicación que permita calcular el total a pagar por una compra ingresando la cantidad y el precio unitario de cada producto. Debe informar además la cantidad de productos adquiridos.

ESTRATEGIA

ingresar el nombre del producto, precio y cantidad
iterar*
mostrar producto y cantidad
calcular el precio producto*cantidad
acumular precio total
acumular cantidad de productos adquiridos
ingresar nuevo producto
fin mientras

mostrar total a pagar y cantidad de productos adquiridos

ALGORITMO

```
Proceso compras
        definir producto Como Caracter;
        definir precio, cantidad, apagar, adquiridos Como Entero;

        Escribir "ingresar el nombre del producto";
        leer producto;
        Escribir "ingresar el precio del producto y la cantidad de unidades que lleva";
        leer precio;
        leer cantidad;

        Mientras precio<>0 Hacer
                precio<- precio*cantidad;
                Escribir "", producto," x", cantidad, " unidades", "  $",precio;
                apagar<- apagar + precio;
```

```
            adquiridos<- adquiridos+cantidad;
            Escribir "ingresar el nombre del producto";
            leer producto;
            Escribir "ingresar el precio del producto y la cantidad de unidades
que lleva o 0 para finalizar";
            leer precio;
            leer cantidad;

        FinMientras

        Escribir "precio final $",apagar;
        Escribir "cantidad de productos: ", adquiridos;

        FinProceso
```

VERIFICACION