PSEUDOCODIGOS

**EJERCICIOS RESUELTOS:**

EJERCICIO 06:

Algoritmo orden\_alfabetico\_merito

Real notas [MAX]

Carácter nombre [MAX][60]

Entero n

Ndatos (n)

Ingreso (nombre,notas,n)

Ordenalfa (nombre,notas,n)

Escribir “Datos en orden alfabetico: ”

Reporte (nombre,notas,n)

Ordemer (nombre,notas,n)

Escribir “Dtos en orden de merito ”

Reporte (nombre,notas,n)

Fin\_algortimo

Ndatos (entero n(R))

Hacer

Escribir “Numero de alumnos: ”

Leer n

Mientras (n<=0)

Fin\_ndatos

Ingreso (carácter nombre [][60], real notas [], entero n)

entero i

para i← 0 hasta n - 1 inc 1 hacer

escribir "Nombre: "

leer nombres[i]

Hacer

escribir "nota: "

leer notas[i]

mientras notas[i] < 0 o notas[i] > 20

fin\_para

fin\_ingreso

reporte (carácter nombre [][60], real notas[], entero n)

entero i

para i🡨0 hasta n inc 1 hacer

escribir ,nombre[i],“, ”,notas[i]

fin\_para

fin\_reporte

ordenalfa (carácter nombre [][60], real notas[], entero n)

entero i,j

cadena temp1 [60]

real temp2

para i ←0 hasta n - 2 inc 1 hacer

para j ← i+1 hasta n – 1 inc 1 hacer

si nombres[i] > nombres[j] entonces

temp1 ← nombres[i]

nombres[i] ← nombres[j]

nombres[j] ← temp1

temp2 ← notas[i]

notas[i] ← notas[j]

notas[j] ← temp2

fin\_si

fin\_para

fin\_para

fin\_ordenalfa

ordenmer (carácter nombre [][60], real notas[], entero n)

entero i,j

cadena temp1 [60]

real temp2

para i ←0 hasta n - 2 inc 1 hacer

para j ← i+1 hasta n – 1 inc 1 hacer

si notas[i] < notas[j] entonces

temp1 ← nombres[i]

nombres[i] ← nombres[j]

nombres[j] ← temp1

temp2 ← notas[i]

notas[i] ← notas[j]

notas[j] ← temp2

fin\_si

fin\_para

fin\_para

fin\_ordenmer

EJERICCIO 07:

Algortimo estatura\_mayor\_promedio

Entero n

Carácter nombre [MAX][60]

Real estaturas [MAX]

Ndatos

Ingreso (nombre,estaturas,n)

Escribir “datos ingresados”

Reporte (nombre,estaturas,n)

Mayoresalpromedio (nombre,estaturas,n)

Fin\_algortimo

Ndatos (entero n(R))

Hacer

Escribir “Numero de alumnos: ”

Leer n

Mientras (n<=0)

Fin\_ndatos

Ingreso (carácter nombre [][60], real notas [], entero n)

entero i

para i← 0 hasta n - 1 inc 1 hacer

escribir "Nombre: "

leer nombres[i]

Hacer

escribir "Estatura: "

leer estaturas[i]

mientras estaturas[i] < 0

fin\_para

fin\_ingreso

reporte (carácter nombre [][60], real notas[], entero n)

entero i

para i🡨0 hasta n inc 1 hacer

escribir ,nombre[i],“, ”,estatura[i]

fin\_para

fin\_reporte

real promedioest (real estaturas [], entero n)

real s

entero i

s🡨0

para i🡨0 hasta n inc 1

s🡨sestaturas [i]

fin\_para

retorno s/n

fin\_promedioest

mayoresalpromedio (carácter nombre [][60], real notas[], entero n)

real pe

entero 1

pe🡨promedioest (estaturas,n)

escribir “Promedios de las estaturas”,pe

escribir “personas con estaturas mayores al promedio: ”

para i🡨0 hasta n inc 1

si estaturas [i]>pe entonces

escribir ,nombre [i],

fin\_si

fin\_para

fin\_mayoresalpromedio

EJERCICIO 08:

Algoritmo nombre\_telefono\_fecha\_menu

Cadena nombre [MAX][60],telefono [MAX][20],fechanac [MAX][20]

Entero n,op

Hacer

Sistema salir

Escribir “AGENDA”

Escribir “[1]ingresar persona ”

Escribir “[2]consultar persona ”

Escribir “[3]modificar persona ”

Escribir “[4]eliminar persona ”

Escribir “[5]mostrar persona ”

Escribir “[6]salir ”

Escribir “ingrese una opción (1-6): ”

Leer op

Según\_sea\_op hacer

Caso 1:

Ingresarpersona (nombre,teléfono,fechanac,n)

Salir

Caso 2:

Consultarpersona (nombre,teléfono,fechanac,n)

Salir

Caso 3:

Modificarpersona(nombre,teléfono,fechanac,n)

Salir

Salir 4:

Eliminarpersona (nombre,teléfono,fechanac,n)

Salir

Caso 5:

Mostrarpersonas (nombre,teléfono,fechanac,n)

Salir

Fin\_segun\_sea

Mientras op!=6

Fin\_algoritmo

Ingresarpersona (cadena nombre [][60],cadena teléfono [][20],cadena fechanac[][20],entero n(R))

Sistema salir

Escribir “Ingrese nombre: ”

Leer nombre [n]

Escribir “Ingrese teléfono: ”

Leer teléfono [n]

Escribir “Ingrese fecha de nacimiento: ”

Leer fechanac [n]

N🡨n+1

Fin\_ingresarpersona

Entero búsqueda (cadena nombre[][60],entero n, cadena nombus)

Entero i

Para i🡨0 hasta n-1 inc 1 hacer

Si nombre [i]= nombus

Retorno i

Fin\_si

Fin\_para

Retorno -1

Fin\_busqueda

Consultarpersona (cadena nombre [][60],cadena teléfono [][20],cadena fechanac[][20],entero n(R))

Cadena nombus [60]

Entero p

Sistema salir

Escribir “nombre a buscar: ”

Leer nombus

P=búsqueda (nombre,n,nombus)

Si p!=-1 entonces

Escribir “datos de la persona”

Escribir “nombre”, nombre [p]

Escribir “teléfono ”, telefono [p]

Escribir “fecha de nacimiento ” fechanac [p]

Sino

Escribir “el nombre no se encuentra ”

Fin\_si

Sistema pausa

Fin\_consultarpersona

Modificarpersona (cadena nombre [][60],cadena teléfono [][20],cadena fechanac[][20],entero n(R))

Cadena nombus [60]

Entero p

Sistema salir

Escribir “nombre a modificar: ”

Leer nombus []

P🡨busqueda (nombre,n,nombus)

Si p!=-1 entonces

Escribir “datos de la persona”

Escribir “nombre”, nombre [p]

Escribir “teléfono ”, telefono [p]

Escribir “fecha de nacimiento ” fechanac [p]

Escribir “nuevo nombre”

Leer nombre [p]

Escribir “nuevo teléfono: ”

leer teléfono [p]

escribir “nueva fecha de nacimiento: ”

leer fechanac [p]

sino

escribir “el nombre no se encuentra”

fin\_si

sistema pausa

fin\_modificarpersona

eliminar (cadena nombre [][60],cadena teléfono [][20],cadena fechanac[][20],entero n(R))

entero i

para i🡨p hasta n-2 entonces

nombre [i]🡨nombre [i+1]

nombre [i]🡨telefono [i+1]

fechanac [i]🡨fechanac [i+1]

fin\_para

n🡨n-1

fin\_eliminar

eliminarpersona (cadena nombre [][60],cadena teléfono [][20],cadena fechanac[][20],entero n(R))

cadena nombus [60]

entero p

sistema salir

escribir “nombre a eliminar ”

leer nombus

p🡨busqueda (nombre,n,nombus)

si p!=-1 entonces

eliminar (nombre,teléfono,fechanac,n,p)

escribir “dato eliminado”

sino

escribir “dato no se encuentra”

sistema pausa

fin\_si

fin\_eliminarpersona

reporte (cadena nombre [][60],cadena teléfono [][20],cadena fechanac[][20],entero n(R))

entero i

para i🡨0 hasta n-1 inc 1 hacer

escribir ,nombre[i], “, ”,telefono [i], “, ”,fechanac [i]

fin\_para

fin\_reporte

mostrarpersonas (cadena nombre [][60],cadena teléfono [][20],cadena fechanac[][20],entero n(R))

sistema salir

escribir “personas”

reporte (nombre,teléfono,fechanac,n)

sistema pausa

fin\_mostrarpersonas

EJERCICIOS PROPUESTOS:

EJERCICIO 01:

Algoritmo numero\_dia

Entero i

Cadena nomdia [MAX]

Ndatos (dia)

Dia\_semana (dia,nomdia)

Escribir “DIA: ”

Fin\_algoritmo

Ndatos (entero dia (R))

Hacer

Escribir “INGRESE UN DIA: ”

Leer dia

Mientras dia <1 o dia >7

Fin\_ndatos

Dia\_semana (entero dia, cadena nomdia [])

Según\_sea dia hacer

Caso 1:

Nomdia [0]🡨’L’

Nomdia [1]🡨’U’

Nomdia [2]🡨’N’

Nomdia [3]🡨’E’

Nomdia [4]🡨’S’

Nomdia [5]🡨0

Salir

Caso 2:

Nomdia [0]🡨’M’

Nomdia [1]🡨’A’

Nomdia [2]🡨’R’

Nomdia [3]🡨’T’

Nomdia [4]🡨’E’

Nomdia [5]🡨’S’

Nomdia [6]🡨0

Salir

Caso 3:

Nomdia [0]🡨’M’

Nomdia [1]🡨’I’

Nomdia [2]🡨’E’

Nomdia [3]🡨’R’

Nomdia [4]🡨’C’

Nomdia [5]🡨’O’

Nomdia [6]🡨’L’

Nomdia [7]🡨’E’

Nomdia [8]🡨’S’

Nomdia [9]🡨0

Salir

Caso 4:

Nomdia [0]🡨’J’

Nomdia [1]🡨’U’

Nomdia [2]🡨’E’

Nomdia [3]🡨’V’

Nomdia [4]🡨’E’

Nomdia [5]🡨’S’

Nomdia [6]🡨0

Caso 5:

Nomdia [0]🡨’V’

Nomdia [1]🡨’I’

Nomdia [2]🡨’E’

Nomdia [3]🡨’R’

Nomdia [4]🡨’N’

Nomdia [5]🡨’E’

Nomdia [6]🡨’S’

Nomdia [7]🡨0

Salir

Caso 6:

Nomdia [0]🡨’S’

Nomdia [1]🡨’A’

Nomdia [2]🡨’B’

Nomdia [3]🡨’A’

Nomdia [4]🡨’D’

Nomdia [5]🡨’O’

Nomdia [6]🡨0

Salir

Caso 7:

Nomdia [0]🡨’D’

Nomdia [1]🡨’O’

Nomdia [2]🡨’M’

Nomdia [3]🡨’I’

Nomdia [4]🡨’N’

Nomdia [5]🡨’G’

Nomdia [6]🡨’O’

Nomdia [7]🡨0

Salir

Fin\_segun\_sea

Fin\_dia\_semana

EJERCICIO 02:

Algortimo palabra\_eco

Cadena [MAX],resp

Entero n

Hacer

Sistema salir

Ingresonum (n)

Ingreso (cadena)

Eco (cadena,n)

Resp=opcion

Mientras (resp=’s’ o resp=’S’ )

Fin\_algoritmo

Ingresonum (entero n(R))

Hacer

Escribir “ingrese el numero de repeticiones del ultimo termino:”

Leer n

Mientras n<=0

Fin\_ingresonum

Ingreso (cadena cadena [])

Escribir “ingrese la cadena: ”

Leer cadena [MAX]

Fin\_ingreso

Eco (cadena cadena [], entero n)

Entero z,i,k,x

Z🡨cadena []

Para i🡨0 hasta z-1 inc 1 hacer

Escribir cadena [i]

Fin\_para

X🡨z-1

Para k🡨0 hasta n-1 inc 1 hacer

Escribir cadena [x]

Fin\_para

Fin\_eco

EJERCICIO 03:

Algoritmo cadena\_longitud

Cadena cadena [MAX],resp

Hacer

Ingreso (cadena)

Separarpal (cadena)

Res=opción

Mientras (resp=’s’ o resp=’S’)

Fin\_algoritmo

Ingreso (cadena cadena [])

Escribir “ingrese la cadena”

Leer cadena [MAX]

Fin\_ingreso

Separarpal (cadena cadena[])

Entero i,n,j

N🡨cadena []

J🡨0

Para i🡨0 hasta n-1 inc 1 hacer

Escribir ,cadena [i],

Si i=x entonces

Escribir “ “,i-j,

Fin\_si

Si isspace(cadena [i]) entonces

Escribir “ ”, i-j

J🡨i+1

Fin\_si

Fin\_para

Fin\_separar

EJERCICIO 04:

Algoritmo palabra\_espacio

Cadena cadena [MAX]

Hacer

Sistema salir

Ingreso (cadena)

Modificacion (cadena)

Resp🡨opcion

Mientras resp=’s’ o resp=’S’

Fin\_algoritmo

Ingreso (cadena cade [])

Escribir “ingrese la cadena: ”

Leer cadena [MAX]

Fin\_ingreso

Modificacion (cadena cade [])

Cadena temp1

Entero i,n,j,b,a

B🡨0

A🡨0

N🡨cad[]

Para i🡨0 hasta 2\*i-1 inc 1 hacer

Cad [i]🡨’ ‘

Fin\_para

Cad [i]🡨0

Para j🡨0 hasta n-2 inc 1 hacer

Temp 1🡨cad [2\*n-2]

Para i🡨2\*n-3 hasta 1+b dis 1 hacer

Cad [i+1]🡨cad [i]

Fin\_para

B🡨b+2

Cad [1+a]🡨temp1

A🡨a+2

Fin\_para

Cad [2\*n+1]🡨0

Escribir “cadena modificada: “,cad

Fin\_modificacion

EJERCICIO 05:

Algoritmo cadena\_asteriscos\_Hashtag

Cadena cadena [MAX], resp

Hacer

Sistema salir

Ingreso (cadena)

Modificación (cadena)

Resp🡨opcion

Mientras resp=’s’ o resp ‘S’

Fin\_algortimo

Ingreso (cadena cadena [])

Escribir “ingrese la cadena: ”

Leer cadena[MAX]

Fin\_ingreso

Modificación (cadena cadena [])

Cadena cadena1 [MAX]🡨”\*\*\*”

Cadena cadena 2 [MAX]🡨”####”

Cadena1🡨cadena

Cadenan1🡨cadena2

Escribir “La cadena modificada: ”,cadena1

Fin\_modificacion

EJERCICIO 06:

Algoritmo cantidad\_promedio\_ponderado

Real cant[MAX],peso[MAX]

Entero n

Cadena resp

Hacer

Sistemas salir

Ndatos (n)

Ingreso (n,cant,peso)

Escribir “el promedio ponderado es: ”, calcularpropon(n,cant,peso)

Mientras resp=’s’ o resp=’S’

Fin\_algoritmo

Ndatos (entero n(R))

Hacer

Escribir “numero de datos: ”

Leer n

Mientras n<=0

Fin\_ndatos

Ingreso (real cant [],real peso [], entero n)

Entero i

Para i🡨0 hasta n-1 inc 1 hacer

Escribir “cantidad”,i+1,”: ”

Leer cant [i]

Escribir “Peso: ”

Leer peso [i]

Fin\_para

Fin\_ingreso

Calcularpropon (real cant [],real peso [], entero n)

Entero i

Real x,y,propon

X🡨0

Y🡨0

Para i🡨0 hasta n-1 inc 1 hacer

X🡨x+cant[i]\*peso

Y🡨y+peso [i]

Fin\_para

Propon🡨x/y

Retorno propon

Fin\_calcularpropon

EJERCICIO 07:

Algoritmo banco

Entero n, opcion

Cadena codigo[MAX][60],nombre[MAX][60]

Real saldo [MAX]

Sistema salir

Ndatos (n)

Ingreso (codigo,nombre,saldo,n)

Hacer

Sistema salir

Escribrir “MENU DEL BANCO”

Escribir “[1]crear una cuenta”

Escribir “[2]consultar saldo”

Escribir “[3]depositar saldo”

Escribir “[4]retirar saldo”

Escribir “[5]mostrar ceuntas ingresadas”

Escribir “[6]Salir”

Escribr “ingrese opciones del (1-6): ”

Leer opcion

Según\_sea\_opcion hacer

caso 1:

crearCuenta(codigo,nombre,saldo,n)

salir

caso 2:

consultarSaldo(codigo,nombre,saldo,n)

salir

caso 3:

depositarSaldo(codigo,nombre,saldo,n)

break

caso 4:

retirarSaldo(codigo,nombre,saldo,n)

break

caso 5:

mostrarCuentas(codigo,nombre,saldo,n)

break

fin\_segun\_sea

mientras (opcion!=6)

fin\_algoritmo

ndatos (entero n (R))

hacer

escribir “ingrese el total de clientes: ”

leer n

mientras n<=0

fin\_ndatos

busqueda (cadena codigo [][60], entero n, cadena codbus [])

entero i

para i🡨0 hasta n-1 inc 1 hacer

si codigo [i]=codbus

retorno i

fin\_si

fin\_para

retornio -1

fin\_busqueda

ingreso (cadena codigo [][60], cadena nombre [][60], real saldo [], entero n)

entero i

para i🡨0 hasta n-1 inc 1 hacer

escribir “ingrese los datos del cliente ”,i+1,”: ”

escribir “ingrese su nombre: ”

leer nombre i,[60]

escribir “ingrese codigo: ”

leer codigo i,[60]

hacer

escribir “ingrese su saldo: ”

leer saldo [i]

mientras saldo [i]<0

fin\_para

fin\_ingreso

crearcuenta (cadena codigo [][60], cadena nombre [][60], real saldo [], entero n(R))

escribir “ingrese datos del nuevo cliente”

escribir “ingrese su nombre: ”

leer nombre n,[60]

escribir “ingrese codigo: ”

leer codigo n,[60]

escribir “ingrese su saldo: ”

leer saldo [n]

n🡨n+1

fin\_crearcuenta

consultarsaldo (cadena codigo [][60], cadena nombre [][60], real saldo [], entero n)

cadena codbus [60]

entero p

sistema salir

escribri “ingrese codigo del cliente”

leer codbus

p🡨busqueda (codigo,n,codbus)

si p!=-1 entonces

escribir “DATOS DEL CLIENTE”

escribir ”Codigo: ”, codigo [p]

escribri “Nmobre”, nombre [p]

escribir “el saldo del cliente es:”, saldo [p]

sino

esribir “el codigo no se cuenta”

sistema pausa

fin\_consultarpersona

depositarsaldo (cadena codigo [][60], cadena nombre [][60], real saldo [], entero n)

cadena codbus [60]

real d

entero p

sistema salir

escribir “ingrese codigo del cliente: ”

leer codbus

p🡨 busqueda(codigo,n,codbus)

si p!=-1 entonces

escribir “DATOS DEL CLIENTE”

escribir “Codigo: ”,codigo [p]

escribir “Nombre: ”,nombre [p]

escribir “saldo: ”,saldo [p]

escribir “cuanto de saldo depositara: ”

leer d

saldo [p]🡨saldo [p]+d

escribir “el saldo nuevo es: ”,saldo [p]

fin\_si

sino

escribir “el codigo no se encuentra”

fin\_si

sistema pausa

fin\_depositarsaldo

retirarsaldo (cadena codigo [][60], cadena nombre [][60], real saldo [], entero n)

cadena codbus [60]

real r

entero p

sistema salir

escribir “ingrese codigo del cliente: ”

leer codbus

p🡨 busqueda(codigo,n,codbus)

si p!=-1 entonces

escribir “DATOS DEL CLIENTE”

escribir “Codigo: ”,codigo [p]

escribir “Nombre: ”,nombre [p]

escribir “saldo: ”,saldo [p]

escribir “cuanto de saldo va a retirar: ”

leer r

saldo [p]🡨saldo [p]-r

escribir “el saldo nuevo es: ”,saldo [p]

fin\_si

sino

escribir “el codigo no se encuentra”

sistema pausa

fin\_si

fin\_retirarsaldo

reporte (cadena codigo [][60], cadena nombre [][60], real saldo [], entero n)

entero i

para i🡨0 hasta n-1 inc 1 hacer

escribir ,nombre [i],”, “codigo [i]”, ”,saldo [i]

fin\_para

fin\_reporte

mostrarcuentas (cadena codigo [][60], cadena nombre [][60], real saldo [], entero n)

sistema salir

escribir “DATOS DE TODAS LAS PERSONAS: ”

reporte (codigo,nombre,saldo,n)

sistema pausa

fin\_mostrarcuentas

EJERCICIO 08:

Cadena codigo [MAX][20],nombre [MAX][60],op

Real valor[MAX],horastrab[MAX],t[MAX]

Entero n

Hacer

Sistema salir

Escribir “AGENDA”

Escribir “[a]Registrar trabajador”

Escribir “[b]Mostrar a trabajador”

Escribir “[c]Modificar a trabajador”

Escribir “[d]Eliminar”

Escribir “[e]mostrar trabajdores ordenados por el nombre”

Escribir “[f]Mostrar trabajadores ordenados por el sueldo”

Escribir “[g]Salir”

Escribir “Ingrese una opcion (a-g): ”

Leer op

Según\_sea\_op hacer

Caso a:

Ingresar trabajador (codigo,nombre,valor,horastrab,n,t)

Salir

Caso b:

Mostrartrabajador (codigo,nombre,valor,horastrab,n,t)

Salir

Caso c:

Modificartrabajador (codigo,nombre,valor,horastrab,n,t)

Salir

Caso d:

eliminartrabajador (codigo,nombre,valor,horastrab,n,t)

Salir

Caso e:

trabajadoresordenadospornombre(codigo,nombre,valor,horastrab,n,t)

reporte (codigo,nombre,valor,horastrab,n,t)

mostrarpersonas (codigo,nombre,valor,horastrab,n,t)

salir

caso f:

trabajadoresordenadosporsueldo (codigo,nombre,valor,horastrab,n,t)

reporte (codigo,nombre,valor,horastrab,n,t)

mostrarpersonas (codigo,nombre,valor,horastrab,n,t)

salir

fin\_segun\_sea

mientras op!=6

fin\_algoritmo

ingresartrabajador (cadena codigo[][20],cadena nombre [][60],real valor [], real horastrab [],entero n(R),real t[])

sistema salir

escribir “ingrese codigo: ”

leer codigo n,[20]

escribir “ingrese nombre: ”

leer nombre n,[60]

escribir “ingrese valor pro hora: ”

leer valor [n]

escribir “ingrese las horas trabajadas: ”

leer horastrab [n]

t[n]🡨valor [n]\*horastrab [n]

n🡨n+1

fin\_ingresartrabajador

entero búsqueda (cadena código [][20], entero n, cadena codbus [])

entero i

para i🡨0 hasta n-1 inc 1 hacer

si código[i]=codbus entonces

retorno i

fin\_si

fin\_para

retorno -1

fin\_busqueda

mostrartrabajador (cadena código [][20],código nombre [][60], real valor [], real horastrab [], entero n, real t[])

cadena codbus [20]

entero i

escribir “código a buscar: ”

leer codbus [20]

p🡨busqueda (código,n,codbus)

si p!=-1 entonces

escribir “DATOS DE LA PERSONA”

escribir “código: ”,código [p]

escribir “nombre: ”,nombre [p]

escribir “valor por hora: ”, valor [p]

escribir “horas trabajadas: ”,horastrab [p]

escribir “el salario es: ”,t[p]

sino

escribir “el nombre no se encuentra”

fin\_si

sistema pausa

fin\_mostrartrabajador

modificartrabajador (cadena código [][20],código nombre [][60], real valor [], real horastrab [], entero n, real t[])

cadena codbus [20]

entero p

escribir “código a modificar: ”

leer codbus [20]

p🡨busqueda (código,n,bodbus)

si p!=p entonces

escribir “DATOS DE LA PERSONA”

escribir “código: ”,código [p]

escribir “nombre: ”,nombre [p]

escribir “valor por hora: ”, valor [p]

escribir “horas trabajadas: ”,horastrab [p]

escribir “el salario es: ”,t[p]

escribir “NUEVOS DATOS”

escribir “ingrese el código: ”

leer código p,[20]

escribir “ingrese nombre: ”

leer nombre p,[60]

escribir “Ingrese valor por hora: ”

leer valor [p]

escribir “ingrese las horas trabajadas: ”

leer horastrab [p]

t[p]🡨valor [p]\*horastrab [p]

sino

escribir “el nombre no se encuentra”

fin\_si

sistema pausa

fin\_modificartrabajador

eliminar (cadena código [][20],cadena nombre [][60], real valor [], real horastrab [], entero n(R), real t[],entero p)

entero i

para i🡨0 hasta n-2 inc 1 hacer

código [i]🡨codigo [i+1]

nombre [i]🡨nombre [i+1]

valor [i]🡨 valor [i+1]

horastrab [i]🡨horastrab [i+1]

t[i]🡨t[i+1]

fin\_para

n🡨n-1

fin\_eliminar

eliminartrabajador (cadena código [][20], cadena nombre [][60], real valor [], real horastrab [], entero n(R), real t[])

cadena codbus [20]

entero p

escribir “código a eliminar: ”

leer codbus [20]

p🡨busqueda (código,n,codbus)

si p!=-1 entonces

eliminar (codigo,nombre,valor,horastrab,n,t,p)

escribir “dato eliminar”

sino

escribir “el dato no se encuentra ”

fin\_si

sistemas pausa

fin\_eliminartrabajador

reporte (cadena código [][20], cadena nombre [][60], real valor [], real horastrab [], entero n(R), real t[])

entero i

para i🡨0 hasta n-1 inc 1 hacer

escribir ,código [i],“, ”,nombre [i],”, vlor por hora: ”,valor [i],”, horas trabajadas: ”, horastrab [i], “, sueldo: ”,t[i]

fin\_para

fin\_reporte

trabajadoresordenadospornombre (cadena código [][20], cadena nombre [][60], real valor [], real horastrab [], entero n(R), real t[])

entero i,j

cadena temp1 [20],temp2 [60]

real temp3,temp4,tem5

para i🡨0 hasta n-2 inc 1 hacer

para j🡨i+1 hasta n-1 inc 1 hacer

si nombre [i]>nombre [j] entonces

temp1🡨codigo [j]

código [i]🡨codigo [j]

código [j] 🡨temp 1

temp2🡨nombre [j]

nombre [i]🡨nombre [j]

nombre [j] 🡨temp 2

temp3🡨valor [j]

valor [i]🡨valor [j]

valor [j] 🡨temp 3

temp4🡨horastrab [j]

horastrab [i]🡨horastrab [j]

horastrab [j] 🡨temp 4

temp5🡨t [j]

t [i]🡨t [j]

t [j] 🡨temp5

fin\_si

fin\_para

fin\_para

fin\_trabajadoresordenadospornombre

mostrarpersonas (cadena código [][20], cadena nombre [][60], real valor [], real horastrab [], entero n(R), real t[])

entero i,j

cadena temp1 [20],temp2 [60]

real temp3,temp4,tem5

para i🡨0 hasta n-2 inc 1 hacer

para j🡨i+1 hasta n-1 inc 1 hacer

si t [i]< t [j] entonces

temp1🡨codigo [j]

código [i]🡨codigo [j]

código [j] 🡨temp 1

temp2🡨nombre [j]

nombre [i]🡨nombre [j]

nombre [j] 🡨temp 2

temp3🡨valor [j]

valor [i]🡨valor [j]

valor [j] 🡨temp 3

temp4🡨horastrab [j]

horastrab [i]🡨horastrab [j]

horastrab [j] 🡨temp 4

temp5🡨t [j]

t [i]🡨t [j]

t [j] 🡨temp5

fin\_si

fin\_para

fin\_para

fin\_trabajadoresordenadosporsueldo

EJERCICIO 09:

Real e1[MAX],e2[MAX],tra[MAX],pro[MAX],proge

Entero n

Cadena nombre [MAX][60],resp

hacer

sistema salir

Ndatos (n)

Ingreso (nombre,e1,e2,tra,n)

Promedio (nombre,e1,e2,tra,n)

Ordenmer (nombre,pro,n)

Escribir “DATOS EN ORDEN DE MERITO: ”

Reporte (nombre pro,n)

Escribir “EL PROMEDIO GENERAL ES: ”,proger

Resp🡨opcion

Mientras (resp=’s’ o resp=’S’)

Fin\_algoritmo

Ndatos (entero n(R))

Hacer

Escribir “numero de alumnos: ”

Leer n

Mientras n<=0

Fin\_ndatos

Ingreso (cadena nombre[][60],real e1 [], real e2 [],real tra [],entero n)

Entero i

Para i🡨0 hasta n-1 inc 1 hacer

Escribir “Nombre: ”

Leer nombre [i]

Hacer

Escribir “nota del examen 1: ”

Leer e1 [i]

Escribir “nota del examen 2: ”

Leer e2

Escribir “nota de trabajos: ”

Leer tra [i]

Mientras e1 [i]<0 y e2 [i]<0 y tra [i]<0

Fin\_para

Fin\_ingreso

promedio (cadena nombre[][60],real e1 [], real e2 [],real tra [],real proge(R),entero n)

Entero i

Real se,s,no

Se🡨0

S🡨0

Para i🡨0 hasta n-1 inc 1 hacer

Se🡨e1 [i]\*0.25+e2[i]\*0.25

No🡨tra[i]\*0.5

Pro[i]🡨se+no

S🡨s+pro

Fin\_para

Proge🡨s/n

Fin\_promedio

Reporte (cadena nombre [][60],real pro [],entero n)

Entero i

Para i🡨0 hasta n-1 inc 1 hacer

Escribir ,nombre [i], “, el promedio ponderado es: ”,pro [i]

Fin\_para

Fin\_reporte

Ordenmer (cadena nombre[][60],real pro [], entero n)

entero i,j

cadena temp1 [60]

real temp2

para i ←0 hasta n - 2 inc 1 hacer

para j ← i+1 hasta n – 1 inc 1 hacer

si pro[i] < pro[j] entonces

temp1 ← nombre[i]

nombre[i] ← nombre[j]

nombre[j] ← temp1

temp2 ← pro[i]

pro[i] ← pro[j]

pro[j] ← temp2

fin\_si

fin\_para

fin\_para

fin\_ordenmer

EJERCICIO 10:

Algoritmo orden\_apellidos\_fecha\_inscripcion

Cadena apellidos [MAX][60],nombre [MAX][60]

Real d [MAX],a [MAX], m [MAX]

Entero n

Hacer

Sistema salir

Ndatos (n)

Ingreso (apellidos, nombre,d,m,a,n)

Ordenarapellidos (apellidos, nombre,d,m,a,n)

Escribir “DATOS ORDENADOS SEGÚN APELLIDOS: ”

Reporte (apellidos, nombre,d,m,a,n)

Ordenafecha (apellidos, nombre,d,m,a,n)

Escribir “DATOS ORDENADOS POR FECHA DE INSCRIPCION: ”

Reporte (apellidos, nombre,d,m,a,n)

Resp🡨opcion

Mientras resp=’s’ o resp=’S’

Fin\_algortimo

Ndatos (enetro n (R))

Hacer

Escribir “ingrese el numero de datos: “

Leer n

Mientras n<=0

Fin\_ndatos

Ingreso (cadena apelllidos[][60],cadena nombre [][60],real d[],real m[],real a[],entero n)

Entero i

Para i🡨0 hasta n-1 inc 1 hacer

Escribir “apellidos : ”

Leer apellidos [i]

Escribir “nombres: ”

Leer nombre [i]

Hacer

Escribir “FECHA DE INGRESO: “

Escribir “dia: ”

Leer d [i]

Escribir “mes: ”

Leer m [i]

Escribir “año: ”

Leer a [i]

Mientras d[i]<0 y m[i]<0 y a[i]

Fin\_para

Fin\_ingreso

Ordenarapellidos (cadena apelllidos[][60],cadena nombre [][60],real d[],real m[],real a[],entero n)

entero i,j

cadena temp1 [60]ntemp2 [60]

real temp3,temp4,temp5

para i ←0 hasta n - 2 inc 1 hacer

para j ← i+1 hasta n – 1 inc 1 hacer

si apellidos[i] > apellidos[j] entonces

temp1 ← apellidos[i]

apellidos[i] ← apellidos[j]

apellidos[j] ← temp1

temp2 ← nombre[i]

nombre[i] ← nombre[j]

nombre[j] ← temp2

temp3 ← d[i]

d[i] ← d[j]

d[j] ← temp3

temp4 ← m[i]

m[i] ← m[j]

m[j] ← temp4

temp5 ← a[i]

a[i] ← a[j]

a[j] ← temp5

fin\_si

fin\_para

fin\_para

fin\_ordenarrapellidos

reporte (cadena apelllidos[][60],cadena nombre [][60],real d[],real m[],real a[],entero n)

entero i

para i🡨0 hasta n-1 inc 1 hacer

escribir ,apellidos [i],”, ”,nombre [i],”, ”,d[i],”-”,m[i],”-”,a[i],”-”

fin\_para

fin\_reporte

ordenarfecha (cadena apelllidos[][60],cadena nombre [][60],real d[],real m[],real a[],entero n)

entero i,j

cadena temp1 [60]ntemp2 [60]

real temp3,temp4,temp5

para i ←0 hasta n - 2 inc 1 hacer

para j ← i+1 hasta n – 1 inc 1 hacer

si a[i]>a[j] entonces

temp1 ← apellidos[i]

apellidos[i] ← apellidos[j]

apellidos[j] ← temp1

temp2 ← nombre[i]

nombre[i] ← nombre[j]

nombre[j] ← temp2

temp3 ← d[i]

d[i] ← d[j]

d[j] ← temp3

temp4 ← m[i]

m[i] ← m[j]

m[j] ← temp4

temp5 ← a[i]

a[i] ← a[j]

a[j] ← temp5

fin\_si

si a[i]=a[j] entonces

si m[i]>m[j] entonces

temp1 ← apellidos[i]

apellidos[i] ← apellidos[j]

apellidos[j] ← temp1

temp2 ← nombre[i]

nombre[i] ← nombre[j]

nombre[j] ← temp2

temp3 ← d[i]

d[i] ← d[j]

d[j] ← temp3

temp4 ← m[i]

m[i] ← m[j]

m[j] ← temp4

temp5 ← a[i]

a[i] ← a[j]

a[j] ← temp5

fin\_si

fin\_si

si a[i] = a[j] y m[i]=m[j] entonces

si d[i]>d[i] entonces

temp1 ← apellidos[i]

apellidos[i] ← apellidos[j]

apellidos[j] ← temp1

temp2 ← nombre[i]

nombre[i] ← nombre[j]

nombre[j] ← temp2

temp3 ← d[i]

d[i] ← d[j]

d[j] ← temp3

temp4 ← m[i]

m[i] ← m[j]

m[j] ← temp4

temp5 ← a[i]

a[i] ← a[j]

a[j] ← temp5

fin\_si

fin\_si

fin\_para

fin\_para

fin\_ordenarafecha

EJERCICIO 11:

Algoritmo numero\_frecuencia

Entero a [50],b [cal+1], c[cal+1],n,j,x

Lógico re

Ndato (n)

Ingreso (a,n)

B[l]🡨a[0]

X🡨 contar ocuerrencias (b[j],n,a)

C[j]🡨x

J🡨j+1

Re🡨buscarcalificaion (a[1],j,b)

Si re=falso entonces

B[j]🡨a[1]

X🡨 contarocuerencias (b[j],n,a)

C[j]🡨x

J🡨j+1

Fin\_si

Re🡨buscarcalificacion (a[2],j,b)

Si re=falso entonces

B[j]🡨a[2]

X🡨 contarocurrencias (b[j],n,a)

C[j]🡨x

J🡨j+1

Fin\_si

Re🡨buscarcalificaion (a[3],j,b)

Si re = falso entonces

B[j]🡨a[3]

X🡨contarocurrencias (b[j],n,a)

C [j]🡨x

J🡨j+1

Fin\_si

Fin\_algoritmo

Ndato (entero n(R))

Hacer

Para i🡨0 hasta n-1 inc 1 hacer

Escribir “digite la calificación del estudiante ”,i+1,”: ”

Leer a[i]

Fin\_para

Fin\_ndato

Entero ocurrencias (entero c, entero n, entero a[])

Entero ocurrencias,i

Ocurrencias🡨0

Para i🡨0 hasta n-1 inc 1 hacer

Si c=a[i] entonces

Ocurrencias🡨ocurrencias+1

Fin\_si

Fin\_para

Fin\_ocurrencias

Lógica buscarcalificacion (entero c, entero j, entero b[])

Entero i

Para i🡨0 hasta j-1 inc 1 hacer

Si c=b[i] entonces

Retorno verdadero

Fin\_si

Fin\_para

Retorno falso

Fin\_buscarcalificacion

Imprimedatos (entero j,entero b,entero c[])

Para i🡨0 hasta j-1 inc 1 hacer

Escribir “el numero ”,b[i],”tiene una frecuencia de: ”,c[i]

Fin\_imprimedatos