

Proyecto Integrador

Ejercicio propuesto de práctica 9 “Arreglo de registros”

Una institución necesita administrar una lista de personas, para ello se necesita desarrollar un algoritmo que permita realizar la carga de información de las personas en un arreglo unidimensional.

Para almacenar los nombres se utilizará un registro que contiene dos campos, uno con un arreglo donde se guardan los datos de las personas y otro con la cantidad actual de personas almacenadas en el arreglo (Utilice un registro con un campo arreglo y otro campo con la cantidad de datos que se han cargado al arreglo).

La cantidad máxima que se podrá almacenar será de 150 personas.

El algoritmo debe ofrecer al usuario un menú de opciones que le permita acceder a las siguientes:

1. Alta de Persona 2. Baja de Persona 3. Listado de Personas 4. Salir

Para poder administrar la lista de personas se deben desarrollar las siguientes acciones y/o funciones:

a) Una función llamada Vacía que reciba como parámetro el registro (con el arreglo de personas y la cantidad cargadas), y retorne verdadero si la lista está vacía y sino debe retornar falso.

b) Una función llamada Llena que reciba como parámetro el registro (con el arreglo de personas y la cantidad cargadas), y retorne Verdadero si el arreglo está completamente lleno (es decir si ya tiene 150 personas cargadas) y sino debe retornar Falso.

c) Una acción que permita insertar una persona en el arreglo. Para ello se pasará como parámetros el registro (con el arreglo de personas y la cantidad cargadas), y los datos de la nueva persona a insertar. La nueva persona se inserta en el arreglo de modo tal que deba quedar siempre ordenado de modo creciente según nuestro alfabeto. Después de ejecutada la acción, la cantidad debe quedar incrementada en una unidad.

d) Una acción que permita suprimir la última persona de la lista, no importa cuál es. Simplemente suprime el último cada vez que la acción es ejecutada. El parámetro que debe recibir es el registro (con el arreglo de personas y la cantidad cargada). Después de ejecutada la acción, la cantidad debe quedar decrementada en una unidad.

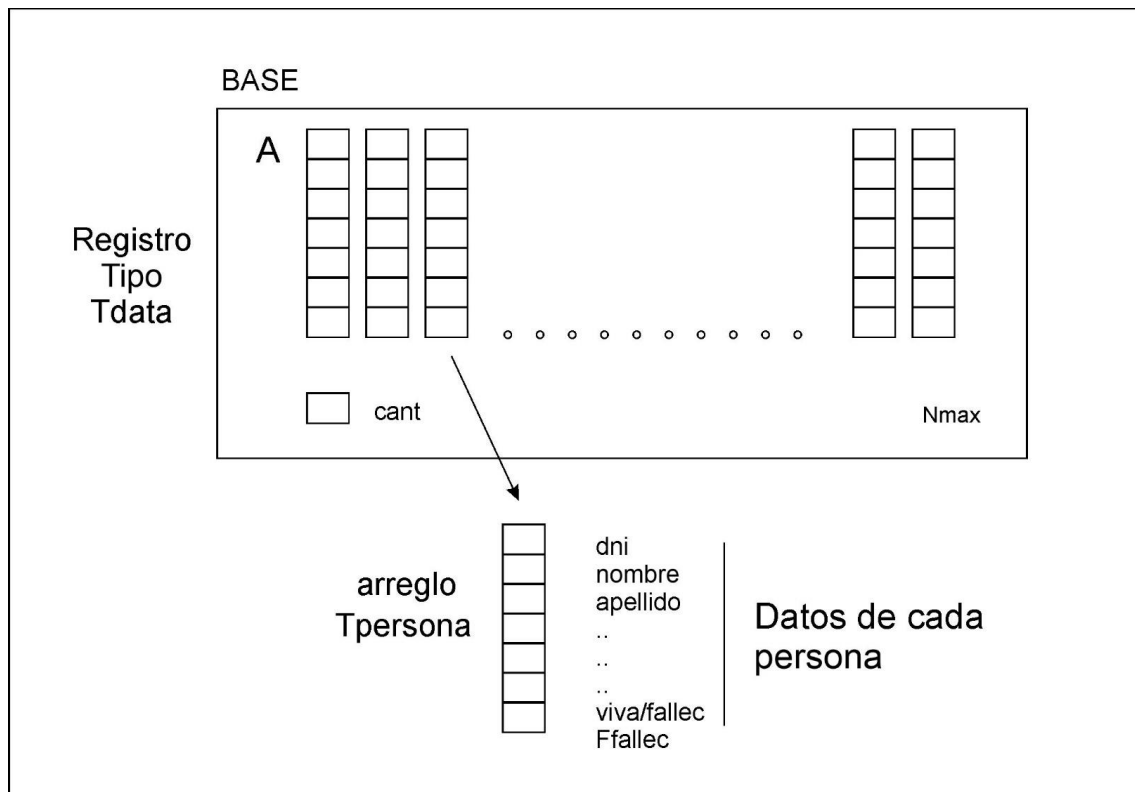
e) Una acción que permita mostrar todas las personas de la lista. Los parámetros que debe recibir son: el arreglo y la cantidad cargada.

Por cada persona, debe almacenarse la siguiente información:

- DNI
- Nombre
- Apellido
- Fecha de Nacimiento
- Edad actual (se calcula con la fecha de nacimiento)

- Ciudad de residencia:
- Dirección Postal de su vivienda
- Correo Electrónico
- Teléfono
- Vive/falleció:
- Fecha de fallecimiento (para los casos de fallecidos):

ANALISIS:



Datos:

Cantidad de personas que almacenara como máximo el registro es 150

Registro con un campo arreglo y otro campo con la cantidad de personas cargadas en el arreglo.

Menú de opciones para el usuario para poder administrar la lista

1. Alta de Persona
2. Baja de Persona
3. Listado de Personas
4. Salir

Resultados:

Lista de personas con todos sus datos

Relaciones:

Para administrar la lista de personas de la institución se crearán las siguientes funciones y acciones:

Función VACIA, que recibirá como parámetro el registro BASE y devolverá un valor booleano VERDADERO si la lista está vacía o FALSO si no lo está.

Función LLENA, que recibirá como parámetro el registro BASE y devolverá un valor booleano VERDADERO si la lista está llena (ósea que tienen 150 personas cargadas) o FALSO si no lo está.

Acción INSERTAR, que recibirá como parámetro el registro BASE y los datos de la nueva persona a insertar en la lista y devolverá el registro modificado con la nueva persona agregada en orden de modo creciente según el alfabeto y la cantidad (variable cant) incrementada en 1.

Acción SUPRIMIR, que recibirá como parámetro el registro BASE y devolverá el registro modificado con la última persona borrada y la cantidad (variable cant) decrementada en 1.

Acción MOSTRAR, que recibirá como parámetro el registro BASE y devolverá por pantalla el listado de todas las personas cargadas en el registro con sus datos respectivos.

ALGORITMO

Algoritmo administrarListaPersonas

Léxico

Opción E {1,2,3,4}

Nmax = 150

Tfecha = (día, mes, año E Z)

Tpersona = (dni E Z, nombre E cadena, apellido E cadena, Fnac E Tfecha,
edad E Z, ciudad E cadena, dirección E cadena, correo E cadena
telef E cadena, estaViva E log, Ffallec E Tfecha)

TarregloPers = arreglo [0 ...Nmax] de Tpersona

Tdata = (A E TarregloPers, cant E {0...Nmax})

base E Tdata

nuevo E Tpersona

Función **vacía** (dato b E Tdata)→log

Inicio

$\leftarrow b.cant = 0$

Ffuncion

Función **llena** (dato l E Tdata)→log

Inicio

$\leftarrow l.cant = 150$

Ffuncion

Acción **insertar** (dato r E Tdata, dato nuevo E Tpersona)

Léxico local

i E N

j E N

inicio

si NO(llena(r)) entonces

$i \leftarrow r.cant - 1$

mientras (i >=0 y (r.a[i].apellido > nuevo.apellido) >0) hacer

$i \leftarrow i - 1$

Fmientras

// en este punto i apunta al índice donde debe insertarse la nueva persona. En c usar
//La función STRCMP que compara dos cadenas pasadas como parámetros (cad1,
//cad2) y determina su orden alfabético devolviendo un num negativo si cad1<cad2,
//un num positivo si cad1>cad2 y 0 si cad1=cad2

//ahora necesitamos mover hacia delante elementos mayores a ese índice en el
//registro

para ($j \leftarrow r.cant-1$, $j > i$, $j \leftarrow j - 1$) hacer

$r.a[j+i] \leftarrow r.a[j]$

fpara

$r.a[i+1] \leftarrow nuevo$

$r.cant \leftarrow r.cant + 1$

salida: mje “ persona insertada correctamente a la lista”

sino

salida: mje “la lista esta llena”

facción

Acción **suprimir** (dato s E Tdata)

Inicio

Si (NO(vacia(s))) entonces

$s.cant \leftarrow s.cant - 1$

sino

salida: mje “la lista esta vacia”

fsi

facción

Acción **mostrar** (dato p E tdata)

Léxico local

i E N

inicio

si (NO(vacia(p))) entonces

salida: mje “mostrar listado”

para (i \leftarrow 0, i \leq p.cant, i \leftarrow i+1) hacer

salida: p.a[i].dni

salida: p.a[i].nombre

salida: p.a[i].apellido

salida: p.a[i].Fnac.dia , p.a[i].Fnac.mes, p.a[i].Fnac.año

salida: p.a[i].edad

salida: p.a[i].ciudad

salida: p.a[i].direccion

salida: p.a[i].correo

salida: p.a[i].telefono

si p.a[i].estaViva entonces

salida: mje "estado: viva"

sino

salida: mje "estado: fallecida"

s salida: p.a[i].Ffallec.dia , p.a[i].Ffallec.mes, p.a[i].Ffallec.año

fsi

salida: mje "-----"

fpara

sino

salida: mje "la lista esta vacia"

fsi

facción

función **calculoEdad** (dato: fechaNac E Tfecha) →N

hoy ← obtenerFechaActual()

edad ← hoy.año – fechaNac.año

si (hoy.mes < fechaNac.mes o (hoy.mes = fechaNac.mes y
hoy.dia < fechaNac.dia)) entonces

edad ← edad – 1

fsi

← edad

ffunción

inicio principal

base.cant ← 0

repetir

salida: mje "MENU:"

salida: mje "1 – ALTA DE PERSONA"

salida: mje "2 – BAJA DE PERSONA"

salida: mje "3 – LISTAR PERSONAS"

salida: mje "4 – SALIR"

salida: mje "INGRESE SU OPCION"

entrada: opción

según opción hacer

caso 1 entrada: nuevo.dni

entrada: nuevo.nombre

entrada: nuevo.apellido

entrada: nuevo.Fnac.dia , p.Fnac.mes, p.Fnac.año

nuevo.edad \leftarrow calculoEdad (nuevo.Fnac.dia , p.Fnac.mes,
p.Fnac.año)

entrada: nuevo.ciudad

entrada: nuevo.direccion

entrada: nuevo.correo

entrada: nuevo.telefono

entrada: nuevo.estaViva // SI o NO

si (NO(nuevo.estaViva)) entonces

entrada: nuevo.Ffallec.dia , p.Ffallec.mes, p.Ffallec.año

fsi

insertar (base, nuevo)

caso 2 suprimir (base)

caso 3 mostrar (base)

caso 4 salida: mje "saliendo del programa"

fsegun

hasta opción ==4

fin