TEMA 1 Parcial - Introducción a la programación 1 - 9/6/2023

Dado un arreglo ***Frase*** de caracteres (de longitud MaxFrase) que está compuesto por palabras separadas por uno o más espacios en blanco. **Realizar el Diagrama de Estructuras y el código** que invierta todas las palabras cuya longitud sea mayor a ***Long*** (entero ingresado por teclado)

Notas:

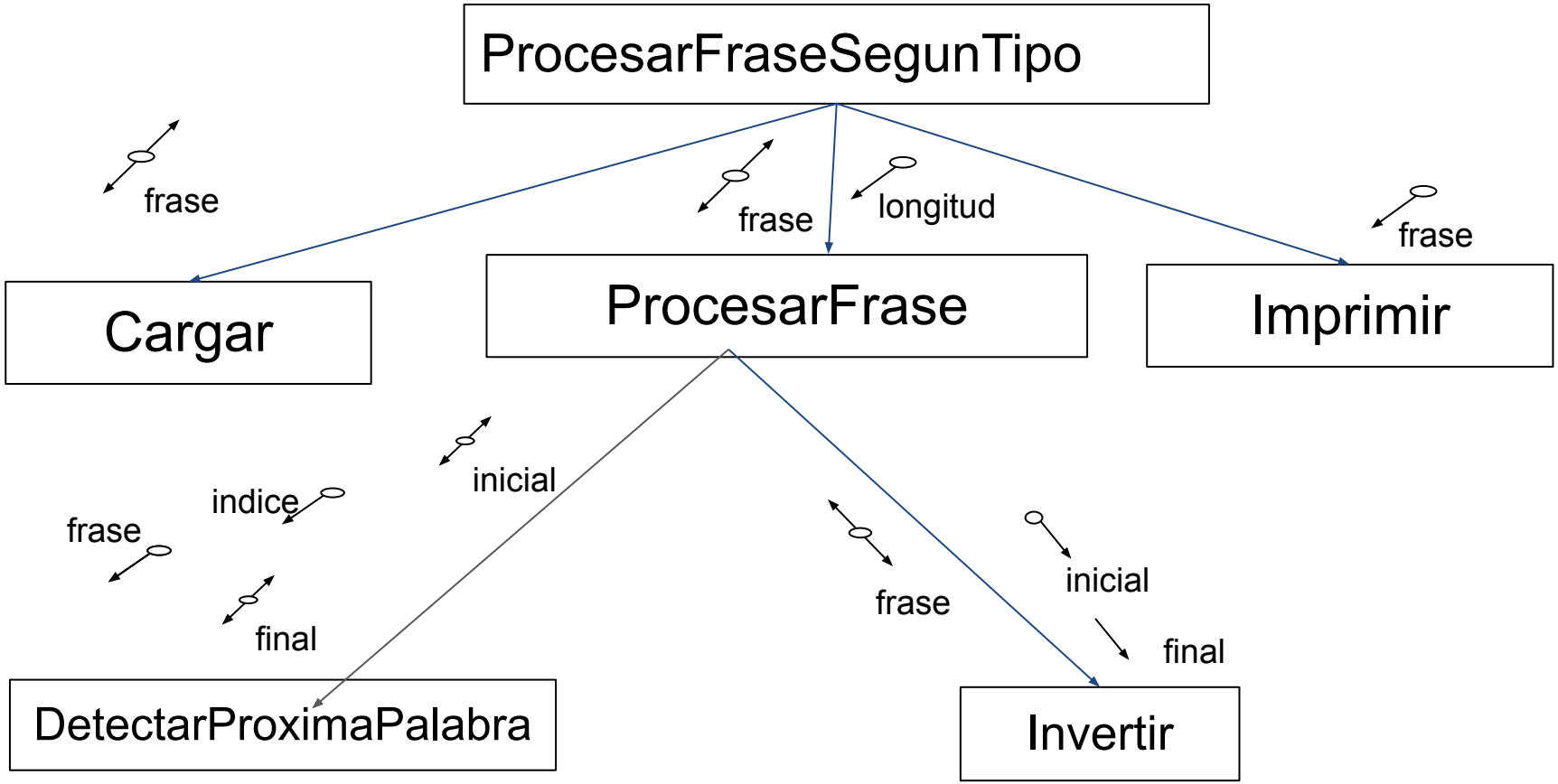
- El arreglo puede o no empezar y terminar con uno o más blancos.
- Asuma que están implementados los procedimientos *cargar(frase)* que carga el arreglo hasta su longitud máxima e *imprimir(frase)*, es decir, invoquelos pero no los implemente.
- Implemente todas las funciones y procedimientos que considere necesarios.
- No puede utilizar estructuras auxiliares.

Ejemplo:

Si se recibe esta ***frase***: " Ya falta poco para las vacaciones de invierno" y ***long***=4

Deberá quedar así: " Ya atlaf poco para las senoicacav de onreivni"

Una posible Solución



```
Program ProcesarFraseSegunTipo;
// Este programa identifica las palabras Palíndromes
conviertiendolas en * y a las restantes las invierte

const
    MinFrase = 1;
    MaxFrase = 4;
    separador = ' ';



type
    T_FRASE=array [MinFrase..MaxFrase] of char;
```

```
// Este programa identifica las palabras de longitud N  y las invierte
.....

//.....programa principal...../
var
    Frase:T_FRASE;
    long: integer;
begin
    Cargar(Frase);     //.....NO SE DEBE IMPLEMENTAR ..//
    writeln('ingrese longitud');
    readln(long);
    ProcesarFrase(Frase, long);
    Imprimir(Frase);   //.....NO SE DEBE IMPLEMENTAR ..//

end.
```

```
Procedure Cargar(var frase:T_FRASE);
// este procedimiento se encarga de pedir al usuario que cargue un
arreglo
var
    Indice:Integer;
begin
    WriteIn('Para separar las palabras use', separador);
    For Indice:= MinFrase to MaxFrase do begin
        Write('Ingrese el valor del arreglo en la posicion ',Indice,':');
        ReadIn(frase[Indice]);
    end;
end;
```

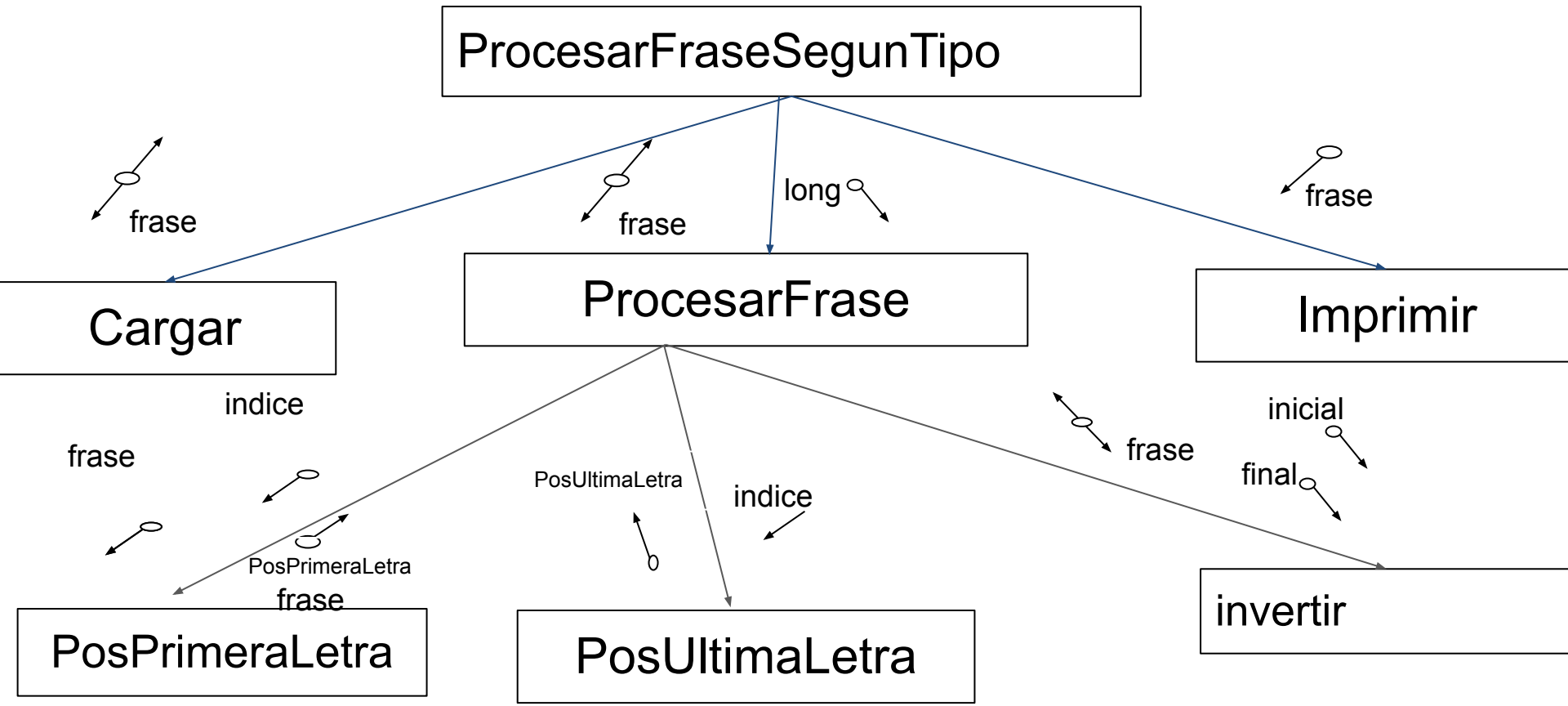
```
procedure Imprimir(frase:T_FRASE);
// este procedimiento muestra el arreglo que carga el usuario
var
    Indice:Integer;
begin
    WriteIn('El arreglo quedó cargado de la siguiente manera:');
    For Indice:= MinFrase to MaxFrase do
          Write(frase[Indice]);
    WriteIn();  // para saltar de linea
end;
```

```
procedure ProcesarFrase(var frase:T_FRASE; longitud: integer);
var
    inicial, final: Integer;
begin
   inicial:= MinFrase;
   final:= MinFrase;
   DetectarProximaPalabra(frase, MinFrase,inicial, final);
    while final<= MaxFrase do
      begin
        if  (final-inicial+1) = longitud
            then
                Invertir(frase, inicial, final)
        DetectarProximaPalabra(frase, final+1,inicial, final);
     end;
end;
```

```
Procedure DetectarProximaPalabra(FRASE:T_FRASE; indice:integer; var posi, posf: integer);
begin
  while (indice <= maxFrase)   and (FRASE[indice]=  separador) do indice:= indice + 1;
   if indice <= maxFrase
    then
       begin
        posi:= indice;
        while (indice <= maxFrase)  and (FRASE[indice] <> separador) do indice:= indice + 1;
        posf:= indice-1;
        end
    else
       posf:= indice;
end;
```

```
procedure Invertir(var frase:T_FRASE; Inicial, final:Integer);
var
    Final: Integer;
    Aux: char;
begin
    while (Inicial < Final) do begin
        Aux := frase[Inicial];
        frase[Inicial] := frase[Final];
        frase[Final] := Aux;
        Inicial := Inicial + 1;
        Final := Final - 1;
    end;
end;
```

Otra posible solución: implementar dos funciones para detectar inicio y fin de palabras


ProcesarFraseSegunTipo
frase
frase
long
frase
Cargar
ProcesarFrase
Imprimir
indice
frase
PosUltimaLetra
indice
frase
inicial
final
PosPrimeraLetra
frase
PosPrimeraLetra
PosUltimaLetra
invertir

```
Program ProcesarFraseSegunTipo;
// Este programa identifica las palabras Palíndromes
convirtiendolas en * y a las restantes las invierte

const
    MinFrase = 1;
    MaxFrase = 4;
    separador = ' ';
    simbolo  = ' *';

type
    T_FRASE=array [MinFrase..MaxFrase] of char;
```

```
Program ProcesarFrase;
// Este programa identifica las palabras de longitud N  y las invierte
..............

//.....programa principal.....//
var
    Frase:T_FRASE;
    long: integer;
begin
    Cargar(Frase);     //.....NO SE DEBE IMPLEMENTAR ..//
    writeln('ingrese longitud');
    readln(long);
    ProcesarFrase(Frase, long);
    Imprimir(Frase);   //.....NO SE DEBE IMPLEMENTAR ..//
end.
```

```
procedure ProcesarFrase(var frase:T_FRASE, longitud: integer);
var
   iniPal, finPal: Integer;
begin
   iniPal:= PosPrimerLetra(FRASE, MinFrase);
   while iniPal<= MaxFrase do
     begin
       finPal:= PosUltimaLetra(frase,iniPal);
       if  (final-inicial+1) = longitud
            then
                 Invertir(frase, iniPal, finPal)
       iniPal:= PosPrimerLetra(FRASE,finPal+ 1);
    end;
end;
```

```
function PosPrimerLetra(frase:T_FRASE; Indice:Integer):Integer;
begin
  while (Indice <= MaxFrase) AND (frase[Indice] = separador) do
              Indice := Indice + 1;
   PosPrimerLetra := Indice;
end;

function PosUltimaLetra(frase:T_FRASE; Indice:Integer):Integer;
begin
  while (Indice <= MaxFrase) AND (frase[Indice] <> separador) do
              Indice := Indice + 1;
   PosUltimaLetra := Indice - 1;
end;
```

Otra posible solución: implementar dos funciones, una para detectar inicio y la otra para la longitud