

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
Laboratorio 4
(Segundo semestre de 2013)

Horario 0582: prof. Fernando Alva

Duración: **1.5 horas**

Observaciones:

- Se permite el uso de material de consulta.

Puntaje total: 0 puntos

PARTE DIRIGIDA

Pregunta 1 (0 puntos)

El siguiente programa es un ejemplo de implementación en ANSI C de listas simplemente enlazadas. Digite el programa en el entorno NetBeans, ejecutando cada una de las funciones decretas en él.

```

1  /* Archivo: ListaSE.h */
2
3  #ifndef LISTASE_H
4  #define LISTASE_H
5
6  typedef int TElemento;
7
8  typedef struct nodo{
9      TElemento elem;
10     struct nodo *sig;
11 }TNodo;
12
13 typedef struct lista{
14     TNodo *inicio;
15     TNodo *fin;
16     int numElem;
17 }TLista;
18
19 void ListaSE_inicializar(TLista *);
20 void ListaSE_insertarInicio(TLista *, TElemento);
21 void ListaSE_insertarFinal(TLista *, TElemento);
22 void ListaSE_imprimirIterativo(TLista *);
23 TNodo* ListaSE_buscarNodo(TLista *, TElemento);
24 void ListaSE_eliminarNodo(TLista *, TElemento);
25 void ListaSE_finalizar(TLista *);
26
27 #endif /* LISTASE_H */

```

```

1  /* Archivo: ListaSE.c */
2
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include "ListaSE.h"
6
7  void ListaSE_inicializar(TLista *listaSE){

```

```

8      listaSE->inicio = NULL;
9      listaSE->fin = NULL;
10     listaSE->numElem = 0;
11 }
12
13 void ListaSE_insertarInicio(TLista *listaSE, TElemento nuevoElem){
14     TNode *ptr_nuevoNode;
15     ptr_nuevoNode = (TNode*) malloc(sizeof(TNode));
16
17     if (ptr_nuevoNode != NULL){
18         ptr_nuevoNode->elem = nuevoElem;
19         ptr_nuevoNode->sig = NULL;
20
21         if (listaSE->fin == NULL)
22             listaSE->fin = ptr_nuevoNode;
23         else
24             ptr_nuevoNode->sig = listaSE->inicio;
25
26         listaSE->inicio = ptr_nuevoNode;
27         listaSE->numElem++;
28     }
29 }
30
31 void ListaSE_insertarFinal(TLista *listaSE, TElemento nuevoElem){
32     TNode *ptr_nuevoNode;
33     ptr_nuevoNode = (TNode*) malloc(sizeof(TNode));
34
35     if (ptr_nuevoNode != NULL){
36         ptr_nuevoNode->elem = nuevoElem;
37         ptr_nuevoNode->sig = NULL;
38
39         if (listaSE->inicio == NULL)
40             listaSE->inicio = ptr_nuevoNode;
41         else
42             listaSE->fin->sig = ptr_nuevoNode;
43
44         listaSE->fin = ptr_nuevoNode;
45         listaSE->numElem++;
46     }
47 }
48
49 void ListaSE_imprimirIterativo(TLista *listaSE){
50     TNode *ptr_rec;
51     ptr_rec = listaSE->inicio;
52     while(ptr_rec != NULL){
53         printf("%03d\n", ptr_rec->elem);
54         ptr_rec = ptr_rec->sig;
55     }
56     printf("NULL\n");
57 }
58
59 TNode* ListaSE_buscarNode(TLista *listaSE, TElemento e){
60     TNode *ptr_rec;
61     ptr_rec = listaSE->inicio;
62     while(ptr_rec != NULL){
63         if (ptr_rec->elem == e)
64             return ptr_rec;
65         ptr_rec = ptr_rec->sig;
66     }
67     return NULL;
68 }
69
70 void ListaSE_eliminarNode(TLista *listaSE, TElemento e){
71     TNode *ptr_rec, *ptr_ant;
72     int termino = 0;
73     ptr_rec = listaSE->inicio;
74     while (ptr_rec != NULL && !termino){

```

```

75     if (ptr_rec->elem == e){
76         if (ptr_rec == listaSE->inicio)
77             listaSE->inicio = listaSE->inicio->sig;
78         else
79             if (ptr_rec == listaSE->fin){
80                 listaSE->fin = ptr_ant;
81                 listaSE->fin->sig = NULL;
82             }else
83                 ptr_ant->sig = ptr_rec->sig;
84         free(ptr_rec);
85         termino = 1;
86         listaSE->numElem--;
87     }else{
88         ptr_ant = ptr_rec;
89         ptr_rec = ptr_rec->sig;
90     }
91 }
92 }
93
94 void ListaSE_finalizar(TLista *listaSE){
95     TNode *ptr_nodo;
96     ptr_nodo = listaSE->inicio;
97     while (ptr_nodo != NULL){
98         listaSE->inicio = listaSE->inicio->sig;
99         free(ptr_nodo);
100        ptr_nodo = listaSE->inicio;
101    }
102    listaSE->fin = NULL;
103 }

```

```

1  /* Archivo: main.c */
2
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include "ListaSE.h"
6
7  int main(int argc, char** argv) {
8      TLista listaSE;
9
10     ListaSE_inicializar(&listaSE);
11
12     ListaSE_insertarFinal(&listaSE, 50);
13     ListaSE_insertarFinal(&listaSE, 34);
14     ListaSE_insertarFinal(&listaSE, 45);
15     ListaSE_insertarFinal(&listaSE, 17);
16
17     ListaSE_imprimirIterativo(&listaSE);
18
19     ListaSE_eliminarNodo(&listaSE, 34);
20     ListaSE_imprimirIterativo(&listaSE);
21
22     ListaSE_finalizar(&listaSE);
23
24     return (EXIT_SUCCESS);
25 }

```

Pando, 30 de octubre de 2013