**PREGUNTA 1 (25 Puntos) Punteros**

Dada la siguiente declaración de la estructura de un nodo:

struct nodo

{ int dato;

struct nodo \*next;

};

Dibuje el resultado de las siguientes instrucciones:

1. nodo \*a, \*b;
2. a = new nodo;
3. a->dato = 4;
4. b = new nodo;
5. a->next = b;
6. b->next = NULL;
7. b->dato = 5;
8. a->next = NULL;
9. b = NULL;

**PREGUNTA 2 (25 Puntos) Lista**

Mencione ventajas y desventajas de las diferentes implementaciones de listas:

1. Arreglo estático

V: No ocupa memoria adicional

D: Largo máximo predefinido/estático

D: Complejidad temporal insertar/eliminar es O(n)

D: Sólo recomendable para listas estáticas sin muchas inserciones y eliminaciones

1. Cursores

V: Complejidad temporal insertar/eliminar es O(1)

D: Ocupa memoria adicional

D: Largo máximo predefinido/estático

1. Enlazada a través de punteros

V: Complejidad temporal insertar/eliminar es O(1)

V: Largo máximo no predefinido

V: Administración de espacio libre por el sistema

D: Ocupa memoria adicional

D: Operaciones adicionales: new y delete

**PREGUNTA 3 (25 Puntos) Lista con Punteros**

*void listaPunteroInvertir(nodo \*&lista)*

*{ nodo \*tmp, \*pos;*

*pos = lista;*

*lista = NULL;*

*while(pos != NULL)*

*{ tmp = pos->next;*

*pos->next = lista;*

*lista = pos;*

*pos = tmp;*

*}*

*}*

**PREGUNTA 4 (10 + 15 = 25 Puntos) Pila**

*void push(nodo \*&p, const int x)*

*{ nodo \*nuevo;*

*nuevo = new nodo;*

*nuevo->dato = x;*

*nuevo->next = p;*

*p = nuevo;*

*}*

*int pop(nodo \*&p)*

*{ int x;*

*nodo \*eliminar;*

*if(p == NULL) error(“Error: pila esta vacia”);*

*x = p->dato;*

*eliminar = p;*

*p = p->next;*

*delete eliminar;*

*return x;*

*}*