**PREGUNTA 1 (10 + 15 = 25 Puntos) Árbol Binario - Recorrido**

1. *void rordpre(struct nodo \*a)*

*{ if(a != NULL)*

*{ cout << a->dato;*

*if(a->izq != NULL) rordpre(a->izq);*

*if(a->der != NULL) rordpre(a->der);*

*}*

*}*

1. *void ordpre(struct nodo \*a)*

*{ pila p;*

*if(a != NULL)*

*{ crearPila(p);*

*push(a, p);*

*while(!vacia(p))*

*{ a = pop(p);*

*cout << a->dato;*

*if(a->der != NULL) push(a->der, p);*

*if(a->izq != NULL) push(a->izq, p);*

*}*

*}*

**PREGUNTA 2 (10 + 15 = 25 Puntos) Árbol Binario - AVL**

10

10-N-15

10-3-15

10-3-15, 3-N-7

10-3-15, 3-N-7, 7-N-9

Rot Izq por nodo 3: 10-7-15, 7-3-9

10-7-15, 7-3-9, 9-8-N

Rot Izq por nodo 7: 9-7-N, 7-3-8

Rot Der por noso 10: 9-7-10, 7-3-8, 10-N-15

**PREGUNTA 3 (25 Puntos) Árbol - Representación**

1. void escribeEtiquetasHojas(struct nodo \*n)

{ struct nodo \*ni;

for(ni = n; ni != NULL; ni = ni->next)

if(ni->hijo == NULL) cout << ni->etiqueta << endl;

}

1. // Se recorre la lista de nodos hasta el nodo con la etiqueta que se busca (nodoEtiq)

// n1 es la raíz del árbol

for(ni = n1; ni->etiqueta != nodoEtiq; ni = ni->next);

// Si el nodo que se busca tiene hijos, se recorre la lista de hijos y se escriben las etiquetas

if(ni->hijo != NULL)

do { hi = ni->hijo;

cout << hi->nodo->etiqueta << " es hijo de " << nodoEtiq << endl << endl;

hi = hi->next;

}

while(hi != NULL);

}

**PREGUNTA 4 (25 Puntos) Hashing**

|  |  |
| --- | --- |
| a) | b) |
| 0--| | 0 – 19 |
| 1🡪 15🡪43 | 1 – 15 |
| 2🡪 2 | 2 – 2 |
| 3--| | 3 – 5 |
| 4🡪53 | 4 – 53 |
| 5🡪12🡪5🡪19 | 5 – 12 |
| 6--| | 6 - 43 |