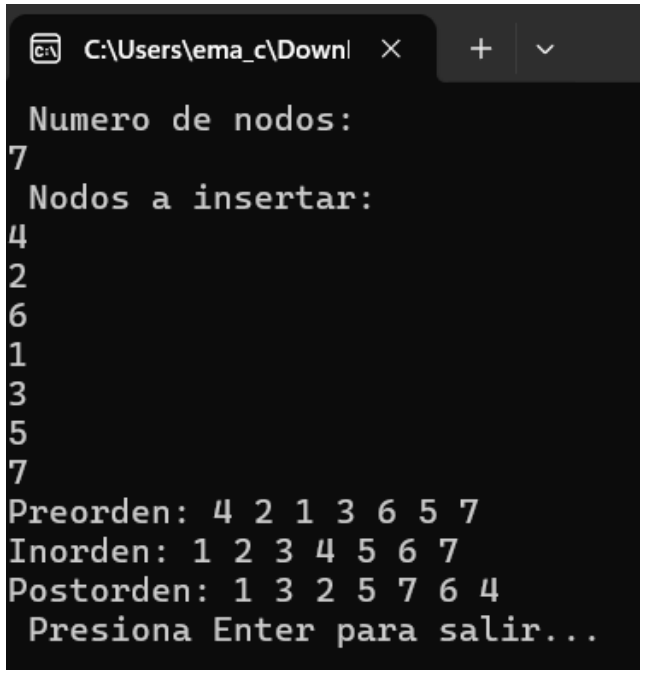


Reporte de Examen práctico.

Problema 20: Insertar nodos a un árbol binario y recorrer en preorden, inorden y postorden.

Nombre del alumno(a):	Claudio Emmanuel Ramírez González	Fecha:	05 - 09 - 2025
-----------------------	-----------------------------------	--------	----------------

Código en el lenguaje Fortran	Ejecución
<pre> 1 program arbolbinario 2 implicit none 3 integer, parameter :: MAXN=100 4 integer :: key(MAXN), left(MAXN), right(MAXN) 5 integer :: n, i, x, root, sz 6 do i=1,MAXN 7 left(i)=0 8 right(i)=0 9 enddo 10 root=0 11 sz=0 12 print *, 'Numero de nodos:' 13 read(*,*) n 14 print *, 'Nodos a insertar:' 15 do i=1,n 16 read(*,*) x 17 call insert_bst(x, root, key, left, right, sz) 18 enddo 19 call print_pre(root, key, left, right) 20 call print_in(root, key, left, right) 21 call print_post(root, key, left, right) 22 print *, 'Presiona Enter para salir...' 23 read(*,*) 24 stop 25 contains 26 subroutine insert_bst(x, root, key, left, right, sz) 27 integer x, root, key(*), left(*), right(*), sz 28 integer cur, parent 29 if (root .eq. 0) then 30 sz = sz + 1 31 key(sz)=x 32 left(sz)=0 33 right(sz)=0 34 root = sz 35 return 36 endif 37 parent = 0 38 cur = root 39 10 continue 40 if (cur .ne. 0) then 41 parent = cur 42 if (x .lt. key(cur)) then 43 cur = left(cur) 44 else 45 cur = right(cur) 46 endif 47 goto 10 48 endif </pre>	 <p>The screenshot shows the execution of the Fortran program. It prompts for the number of nodes (7) and the nodes to insert (4, 2, 1, 3, 6, 5, 7). It then displays the pre-order, in-order, and post-order traversals of the resulting binary tree.</p> <pre> Numero de nodos: 7 Nodos a insertar: 4 2 6 1 3 5 7 Preorden: 4 2 1 3 6 5 7 Inorden: 1 2 3 4 5 6 7 Postorden: 1 3 2 5 7 6 4 Presiona Enter para salir... </pre>

```

49     sz = sz + 1
50     key(sz)=x
51     left(sz)=0
52     right(sz)=0
53     if (x .lt. key(parent)) then
54         left(parent)=sz
55     else
56         right(parent)=sz
57     endif
58 end subroutine
59 subroutine print_pre(root, key, left, right)
60     integer root, key(*), left(*), right(*)
61     integer stack(200), top, v
62     write(*,'(A)',advance='no') 'Preorden: '
63     if (root .eq. 0) then
64         print *; return
65     endif
66     top=1; stack(top)=root
67     do while (top .gt. 0)
68         v=stack(top); top=top-1
69         write(*,'(I0,IX)',advance='no') key(v)
70         if (right(v) .ne. 0) then
71             top=top+1; stack(top)=right(v)
72         endif
73         if (left(v) .ne. 0) then
74             top=top+1; stack(top)=left(v)
75         endif
76     end do
77     print *
78 end subroutine
79 subroutine print_in(root, key, left, right)
80     integer root, key(*), left(*), right(*)
81     integer stack(200), top, cur
82     write(*,'(A)',advance='no') 'Inorden: '
83     top=0; cur=root
84     do while (cur .ne. 0 .or. top .gt. 0)
85         if (cur .ne. 0) then
86             top=top+1; stack(top)=cur
87             cur=left(cur)
88         else
89             cur=stack(top); top=top-1
90             write(*,'(I0,IX)',advance='no') key(cur)
91             cur=right(cur)
92         endif
93     end do
94     print *
95 end subroutine
96 subroutine print_post(root, key, left, right)
97     integer root, key(*), left(*), right(*)
98     integer s1(200), s2(200), t1, t2, v
99     write(*,'(A)',advance='no') 'Postorden: '
100     if (root .eq. 0) then
101         print *; return
102     endif
103     t1=1; s1(t1)=root; t2=0
104     do while (t1 .gt. 0)
105         v=s1(t1); t1=t1-1
106         t2=t2+1; s2(t2)=v
107         if (left(v) .ne. 0) then
108             t1=t1+1; s1(t1)=left(v)
109         endif
110         if (right(v) .ne. 0) then
111             t1=t1+1; s1(t1)=right(v)
112         endif
113     end do
114     do while (t2 .gt. 0)
115         v=s2(t2); t2=t2-1
116         write(*,'(I0,IX)',advance='no') key(v)
117     end do
118     print *
119 end subroutine
120 end program arbolbinario

```

C:\Users\ema_c\Downl × + ▾ —

Numero de nodos:
9

Nodos a insertar:
50
30
70
20
40
60
80
35
65

Preorden: 50 30 20 40 35 70 60 65 80
Inorden: 20 30 35 40 50 60 65 70 80
Postorden: 20 35 40 30 65 60 80 70 50
Presiona Enter para salir...

Código en el lenguaje Pascal

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TURBO
File Edit Search Run Compile Debug Tools Options Window Help
[*] \TPMDB\DOC\WELCOME.TXT 1
[*] \TPMDB\SOURCE\ARBOLB*1.PAS 2-[1]-
program arbolbinario;
type
  PNode = ^TNode;
  TNode = record
    key: LongInt;
    L, R: PNode;
  end;
procedure Insert(var root: PNode; x: LongInt);
begin
  if root = nil then begin
    New(root); root^.key := x; root^.L := nil; root^.R := nil;
  end
  else if x < root^.key then Insert(root^.L, x)
  else Insert(root^.R, x);
end;
procedure PreOrder(t: PNode);
begin
  if t = nil then exit;

```

```

  Write(t^.key, ' ');
  PreOrder(t^.L);
  PreOrder(t^.R);
end;
procedure InOrder(t: PNode);
begin
  if t = nil then exit;
  InOrder(t^.L);
  Write(t^.key, ' ');
  InOrder(t^.R);
end;
procedure PostOrder(t: PNode);
begin
  if t = nil then exit;
  PostOrder(t^.L);
  PostOrder(t^.R);
  Write(t^.key, ' ');
end;
end.
40:27 S_
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TURBO
File Edit Search Run Compile Debug Tools Options Window Help
[*] \TPMDB\DOC\WELCOME.TXT 1
[*] \TPMDB\SOURCE\ARBOLB*1.PAS 2-[1]-
procedure PostOrder(t: PNode);
begin
  if t = nil then exit;
  PostOrder(t^.L);
  PostOrder(t^.R);
  Write(t^.key, ' ');
end;
var
  n, i, x: LongInt; root: PNode;
begin
  root := nil;
  Write('Numero de nodos: '); ReadLn(n);
  WriteLn('Nodos a insertar:');
  for i := 1 to n do begin Read(x); Insert(root, x); end;

  WriteLn; Write('Preorden: '); PreOrder(root); WriteLn;
  Write('Inorden: '); InOrder(root); WriteLn;
  Write('Postorden: '); PostOrder(root); WriteLn;
end.
53:26 S_
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```

Ejecución

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TURBO
Turbo Pascal (With DOSBox) 7.3.4
Copyright (C) 2017 - 2020 Luu Nguyen Thien Hau
Free and open-source under the terms of MIT License.

Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Numero de nodos: 7
Nodos a insertar:
4
2
6
1
3
5
7

Preorden: 4 2 1 3 6 5 7
Inorden: 1 2 3 4 5 6 7
Postorden: 1 3 2 5 7 6 4
Numero de nodos:

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TURBO
Turbo Pascal (With DOSBox) 7.3.4
Copyright (C) 2017 - 2020 Luu Nguyen Thien Hau
Free and open-source under the terms of MIT License.

Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Numero de nodos: 9
Nodos a insertar:
50
30
70
20
40
50
30
35
65

Preorden: 50 30 20 40 35 70 60 65 80
Inorden: 20 30 35 40 50 60 65 70 80
Postorden: 20 35 40 30 65 60 80 70 50
Numero de nodos:

```

Código en el lenguaje C/C++	Ejecución
<pre> arbolbinario.c 1 #include <stdio.h> 2 #include <stdlib.h> 3 4 typedef struct Node { 5 int key; 6 struct Node *L, *R; 7 } Node; 8 9 Node* newNode(int x){ 10 Node* n = (Node*)malloc(sizeof(Node)); 11 n->key = x; n->L = n->R = NULL; return n; 12 } 13 14 void insert(Node** root, int x){ 15 if(*root==NULL){ *root = newNode(x); return; } 16 if(x < (*root)->key) insert(&(*root)->L, x); 17 else insert(&(*root)->R, x); 18 } 19 20 void pre(Node* t){ if(!t) return; printf("%d ", t->key); pre(t->L); pre(t->R); } 21 void in (Node* t){ if(!t) return; in(t->L); printf("%d ", t->key); in(t->R); } 22 void post(Node* t){ if(!t) return; post(t->L); post(t->R); printf("%d ", t->key); } 23 24 int main(void){ 25 int n,x; Node* root=NULL; 26 printf("Numero de nodos:\n"); 27 if(scanf("%d",&n)!=1) return 0; 28 printf("Nodos a insertar:\n"); 29 for(int i=0;i<n;i++){ scanf("%d",&x); insert(&root,x); } 30 printf("Preorden: "); pre(root); puts(""); 31 printf("Inorden: "); in(root); puts(""); 32 printf("Postorden: "); post(root); puts(""); 33 return 0; 34 } 35 </pre>	<pre> Numero de nodos: 7 Nodos a insertar: 4 2 6 1 3 5 7 Preorden: 4 2 1 3 6 5 7 Inorden: 1 2 3 4 5 6 7 Postorden: 1 3 2 5 7 6 4 ...Program finished with exit code 0 Press ENTER to exit console. </pre> <pre> Numero de nodos: 9 Nodos a insertar: 50 30 70 20 40 60 80 35 65 Preorden: 50 30 20 40 35 70 60 65 80 Inorden: 20 30 35 40 50 60 65 70 80 Postorden: 20 35 40 30 65 60 80 70 50 ...Program finished with exit code 0 Press ENTER to exit console. </pre>

Código en el lenguaje Java

```

J arbolbinario.java 1 X
J arbolbinario.java > Language Support for Java(TM) by Red Hat > arbolbinario > main(String[])
1 import java.util.*;
2
3 public class arbolbinario {
4     static class Node {
5         int key; Node L, R;
6         Node(int k){ key=k; }
7     }
8     static Node insert(Node t, int x){
9         if(t==null) return new Node(x);
10        if(x < t.key) t.L = insert(t.L, x);
11        else t.R = insert(t.R, x);
12        return t;
13    }
14    static void pre(Node t){ if(t==null) return; System.out.print(t.key+" "); pre(t.L); pre(t.R); }
15    static void in (Node t){ if(t==null) return; in(t.L); System.out.print(t.key+" "); in(t.R); }
16    static void post(Node t){ if(t==null) return; post(t.L); post(t.R); System.out.print(t.key+" "); }
17
18    Run | Debug | Run main | Debug main
19    public static void main(String[] args){
20        Scanner sc=new Scanner(System.in);
21        System.out.println(x:"Numero de nodos:");
22        int n=sc.nextInt();
23        System.out.println(x:"Nodos a insertar:");
24        Node root=null;
25        for(int i=0;i<n;i++) root = insert(root, sc.nextInt());
26        System.out.print(s:"Preorden: "); pre(root); System.out.println();
27        System.out.print(s:"Inorden: "); in(root); System.out.println();
28        System.out.print(s:"Postorden: "); post(root); System.out.println();
29    }
30

```

Ejecución

```

sages' '-cp' 'C:\Users\ema_c\AppData\Roaming\O
Numero de nodos:
7
Nodos a insertar:
4
2
6
1
3
5
7
Preorden: 4 2 1 3 6 5 7
Inorden: 1 2 3 4 5 6 7
Postorden: 1 3 2 5 7 6 4
PS C:\Users\ema_c\OneDrive\Documentos\Java>

```

```

PS C:\Users\ema_c\OneDrive\Documentos\Java> & 'C
va_d48d8ff8\bin' 'arbolbinario'
Numero de nodos:
9
Nodos a insertar:
50
30
70
20
40
60
80
35
65
Preorden: 50 30 20 40 35 70 60 65 80
Inorden: 20 30 35 40 50 60 65 70 80
Postorden: 20 35 40 30 65 60 80 70 50
PS C:\Users\ema_c\OneDrive\Documentos\Java>

```