

# 输入输出样例

样例 1：（层序遍历求深度）

用例序号	输入	输入含义	输出	输出含义
1	29		0	不是空的二叉树
	2	选择第 2 棵二叉树	4	树的深度为 4
	1	初始化	A	根结点为 A
	3	创建二叉树	ADBECFG	层序遍历的结果
	AD^EF^^G^^B^C^^	含空二叉树的格式		
	5	判断是否为空的二叉树		
	6	求树的深度		
	7	求根结点		
	20	按层序遍历二叉树		
	2	销毁二叉树		
	0	退出程序		

样例 2：（访问左右孩子）

用例序号	输入	输入含义	输出	输出含义
1	29		A	根结点为 A
	3	选择第 3 棵二叉树	D	左孩子 D
	1	初始化	E	右孩子 E
	3	创建二叉树	F	左孩子 F
	AD^EF^^G^^B^C^^	含空二叉树的格式	^	右孩子空
	7	求根结点		
	11	求左孩子		
	12	求右孩子		
	11	求左孩子		
	12	求右孩子		
	0	退出程序		

样例 3：（访问兄弟/双亲结点/赋值结点）

用例序号	输入	输入含义	输出	输出含义
1	29		A	根结点为 A
	2	选择第 2 棵二叉树	B	右孩子 B
	1	初始化	D	左兄弟 D
	3	创建二叉树	E	右孩子 E
	AD^EF^^G^^B^C^^	含空二叉树的格式	F	左孩子 F
	7	求根结点	G	右兄弟 G

	12	求右孩子	S	左兄弟 S
	13	求左兄弟	E	双亲 E
	9		J	双亲 J
	J	将结点 D 改成 J	A	双亲 A
	12	求右孩子	^	双亲空
	11	求左孩子		
	9			
	S	将结点 F 改为 S		
	14	求右兄弟		
	13	求左兄弟		
	10	求双亲		
	10	求双亲		
	10	求双亲		
	10	求双亲		
	0	退出程序		

样例 4：（递归前序）

用例序号	输入	输入含义	输出	输出含义
1	29		0	不是空的二叉树
	2	选择第 2 棵二叉树	4	树的深度为 4
	1	初始化	A	根结点为 A
	3	创建二叉树	ADEFGBC	前序遍历的结果
	AD^EF^^G^^B^C^^	含空二叉树的格式		
	5	判断是否为空的二叉树		
	6	求树的深度		
	7	求根结点		
	17	按前序遍历二叉树		
	2	销毁二叉树		
	0	退出程序		

样例 5：（非递归中序）

用例序号	输入	输入含义	输出	输出含义
1	29		0	不是空的二叉树
	2	选择第 2 棵二叉树	4	树的深度为 4
	1	初始化	A	根结点为 A
	3	创建二叉树	DFEGABC	中序遍历的结果
	AD^EF^^G^^B^C^^	含空二叉树的格式		
	5	判断是否为空的二叉树		
	6	求树的深度		
	7	求根结点		
	18	按中序遍历二叉树		
	2	销毁二叉树		

	0	退出程序		
--	---	------	--	--

样例 6：（哈夫曼树输入/后序遍历）

用例序号	输入	输入含义	输出	输出含义
1	29 2 1 31 akghiC^^U^^H^^j R^^S^^IE^^mI^^N ^^znopqB^^G^^rP ^^sV^^tuvJ^^Q^^ wX^^Z^^K^^O^^x A^^yD^^L^^bcdeY ^^W^^fM^^F^^T^ ^ ^^ 6 19 2 0	选择第 2 棵二叉树 初始化 创建哈夫曼树 含空二叉树的格式        求树的深度 按后序遍历二叉树 销毁二叉树 退出程序	11 CUiHhRSj gEINmlkB GqPVJQv XZwuKtsr pOoADLy xnYWeMF fdTc bza	树的深度为 11 后序遍历的结果

样例 7：（编码）

用例序号	输入	输入含义	输出	输出含义
1	29 2 1 31 akghiC^^U^^H^^j R^^S^^IE^^mI^^N ^^znopqB^^G^^rP ^^sV^^tuvJ^^Q^^ wX^^Z^^K^^O^^x A^^yD^^L^^bcdeY ^^W^^fM^^F^^T^ ^ ^^ 6 32 THIS PROGRAM IS MY FAVORITE 2 0	选择第 2 棵二叉树 初始化 创建哈夫曼树 含空二叉树的格式        求树的深度 编码 需要编码的字符串  销毁二叉树 退出程序	11 11010001 01100011 11110001 00010100 11000010 01010101 10010111 01100011 11111001 01100001 11110011 10101000 11010010 01001101 101010	树的深度为 11 编码结果

样例 8：（解码）

用例序号	输入	输入含义	输出	输出含义
1	29 2 1 31 akghiC^^U^^H^^j R^^S^^IE^^mI^^N ^^znopqB^^G^^rP ^^sV^^tuvJ^^Q^^ wX^^Z^^K^^O^^x A^^yD^^L^^bcdeY ^^W^^fM^^F^^T^ ^ ^^ 6 33 11000100011100001 11101000100101111 10001010111110100 10110000011001111 00001111100001100 11111101100111111 0010101000100011 2 0	选择第 2 棵二叉树 初始化 创建哈夫曼树 含空二叉树的格式        求树的深度 解码 需要解码的编码串        销毁二叉树 退出程序	11 WHY ARE WE TRYING GO TO MARS	树的深度为 11 解码结果

样例 9：（解码编码自校验）

用例序号	输入	输入含义	输出	输出含义
1	29 2 1 31 akghiC^^U^^H^^j R^^S^^IE^^mI^^N ^^znopqB^^G^^rP ^^sV^^tuvJ^^Q^^ wX^^Z^^K^^O^^x A^^yD^^L^^bcdeY ^^W^^fM^^F^^T^ ^ ^^ 6	选择第 2 棵二叉树 初始化 创建哈夫曼树 含空二叉树的格式        求树的深度	11 1 1	树的深度为 11 校验正确 校验正确

34	THIS PROGRAM IS MY FAVORITE	编码解码校验 字符串		
35	11000100011100001 11101000100101111 10001010111110100 10110000011001111 00001111100001100 11111101100111111 0010101000100011	解码编码校验 编码串		
2		销毁二叉树		
0		退出程序		

样例 10：（插入/删除子树/自由访问孩子、双亲、兄弟）

用例序号	输入	输入含义	输出	输出含义
1	29		1	是空的二叉树
	2	选择第 2 棵二叉树	4	树的深度为 4
	1	初始化	A	根结点为 A
	5	是否空二叉树	D	左孩子为 D
	3	创建二叉树	ADBECFG	层序遍历 2 的结果
	AD^EF^^G^^B^C^^	含空二叉树的格式	KMNW	层序遍历 5 的结果
	6	求树的深度	K	根结点为 K
	7	求根结点	^	右孩子为 ^
	11	访问左孩子	RSTUV	层序遍历 6 的结果
	20	按层序遍历二叉树	R	根结点为 R
	29		^	右孩子为 ^
	5	选择第 5 棵二叉树	^	D 的左孩子为 ^
	1	初始化	E	右孩子为 E
	3	创建二叉树	D	双亲为 D
	KMN^^W^^^	含空二叉树的格式	ADBKECM	层序遍历
	20	层序遍历	FGNW	
	7	访问根结点	ADBKRCM	层序遍历
	12	检查右孩子	SENWTFG	
	29		UV	
	6	选择第 6 棵二叉树	6	树的深度为 6
	1	初始化	B	右兄弟为 B
	3	创建二叉树	A	双亲结点为 A
	RS^TU^^V^^^	含空二叉树的格式	ADKRMSE	层序遍历
	20	层序遍历	NWTFGU	
	7	访问根结点	V	
	12	检查右孩子	NWMKUV	后序遍历
	29		TSFGERDA	

	2	选择第 2 棵二叉树		
	11	从结点 D 访问左孩子		
	12	访问右孩子		
	10	访问双亲		
	15			
	L 5	将子树 5 插入左子树		
	20	层序遍历		
	15			
	R 6	将子树 6 插入右子树		
	20	层序遍历		
	6	求树的深度		
	14	访问右兄弟		
	10	访问双亲		
	16			
	R	删除右子树		
	20	层序遍历		
	19	后序遍历		
	2	销毁二叉树		
	0	退出程序		