```c
#include <stdio.h>
#include <stdlib.h>
//#include "tree.h"
#include "linkqueue.h"

bitree * tree_create() {
	data_t ch;
	bitree *r;

	scanf("%c", &ch);
	if (ch == '#')
		return NULL;

	if ((r = (bitree *)malloc(sizeof(bitree))) == NULL) {
		printf("malloc failed\n");
		return NULL;
	}
	r->data = ch;
	r->left  = tree_create();
	r->right = tree_create();
	return r;
}

void preorder(bitree * r) {
	if (r == NULL) {
		return;
	}
	printf("%c", r->data);
	preorder(r->left);
	preorder(r->right);
}

void inorder(bitree * r) {
	if (r == NULL) {
		return;
	}
	inorder(r->left);
	printf("%c", r->data);
	inorder(r->right);
}

void postorder(bitree * r) {
	if (r == NULL) {
		return;
	}
	postorder(r->left);
	postorder(r->right);
	printf("%c", r->data);
}

void layerorder(bitree * r) {
	linkqueue * lq;

	if ((lq = queue_create()) == NULL)
		return;

	if (r == NULL)
		return;
```

```
	printf("%c", r->data);
	enqueue(lq, r);

	while (!queue_empty(lq)) {
		r = dequeue(lq);
		if (r->left) {
			printf("%c", r->left->data);
			enqueue(lq, r->left);
		}
		if (r->right) {
			printf("%c", r->right->data);
			enqueue(lq, r->right);
		}
	}
	puts("");
}
```