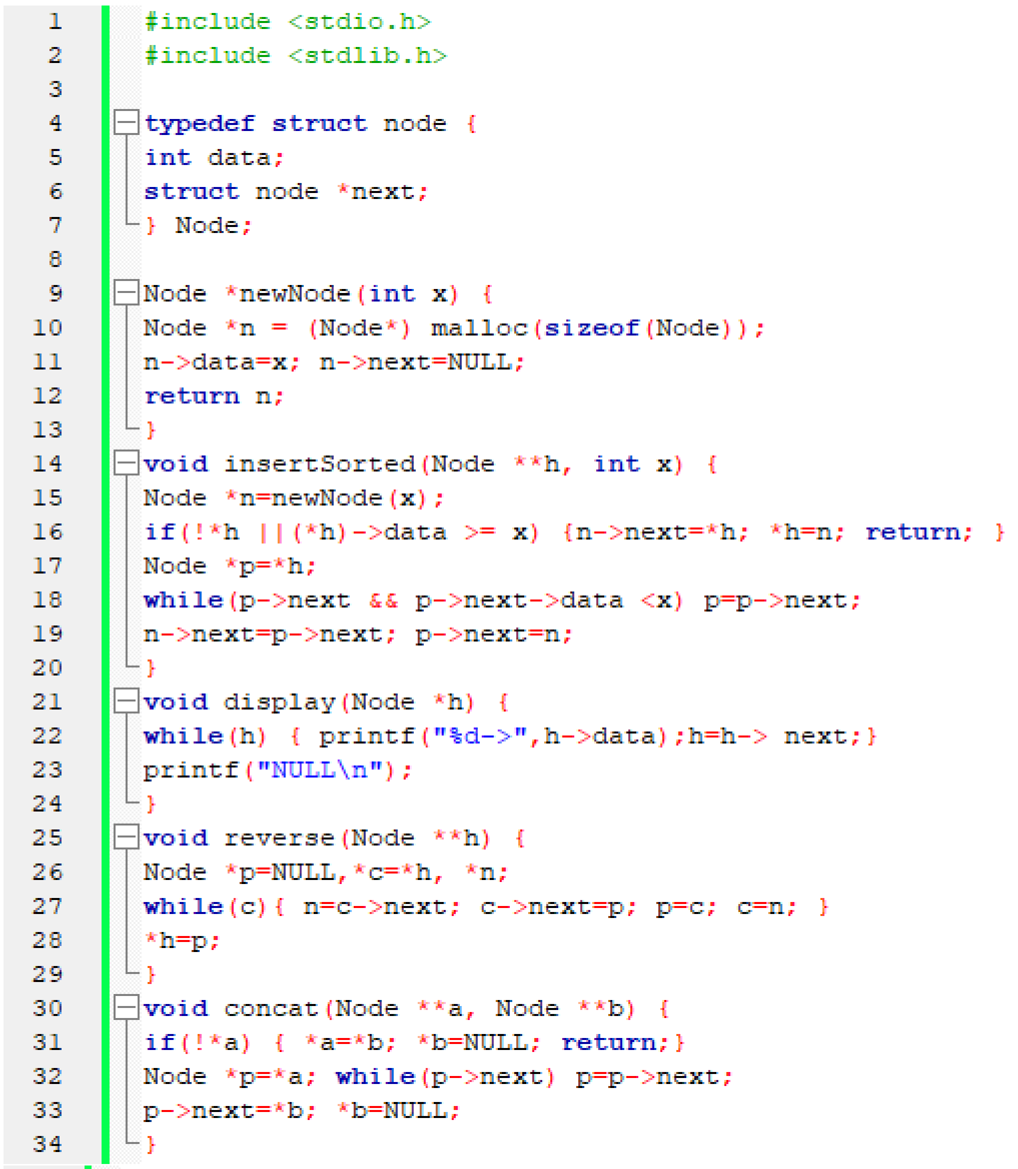
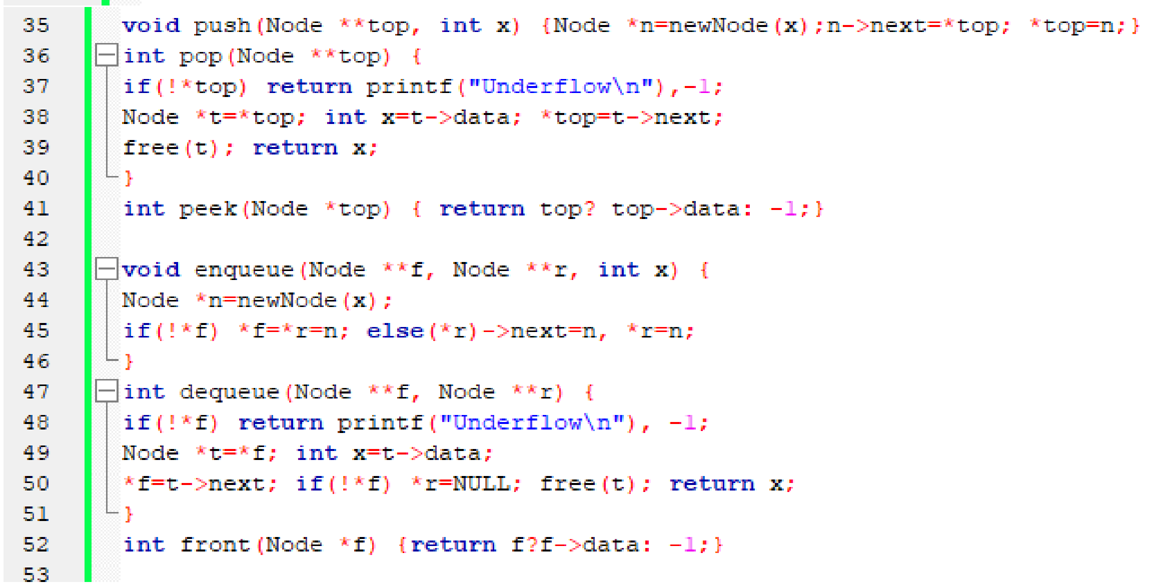
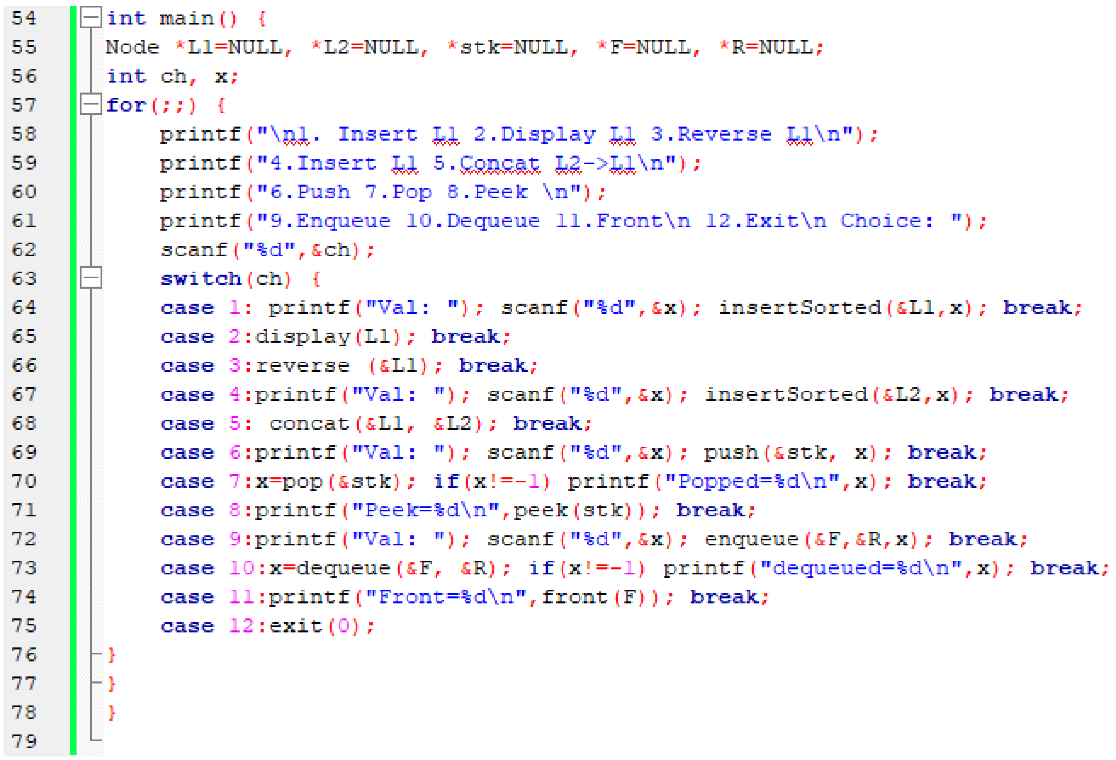
LAB 6-
Operations on singly linked list using stacks and queue



```c
#include <stdio.h>
#include <stdlib.h>

typedef struct node {
int data;
struct node *next;
} Node;

Node *newNode(int x) {
Node *n = (Node*) malloc(sizeof(Node));
n->data=x; n->next=NULL;
return n;
}
void insertSorted(Node **h, int x) {
Node *n=newNode(x);
if(!*h || (*h)->data >= x) {n->next=*h; *h=n; return; }
Node *p=*h;
while(p->next && p->next->data <x) p=p->next;
n->next=p->next; p->next=n;
}
void display(Node *h) {
while(h) { printf("%d->",h->data);h=h-> next;}
printf("NULL\n");
}
void reverse(Node **h) {
Node *p=NULL,*c=*h, *n;
while(c){ n=c->next; c->next=p; p=c; c=n; }
*h=p;
}
void concat(Node **a, Node **b) {
if(!*a) { *a=*b; *b=NULL; return;}
Node *p=*a; while(p->next) p=p->next;
p->next=*b; *b=NULL;
}
```

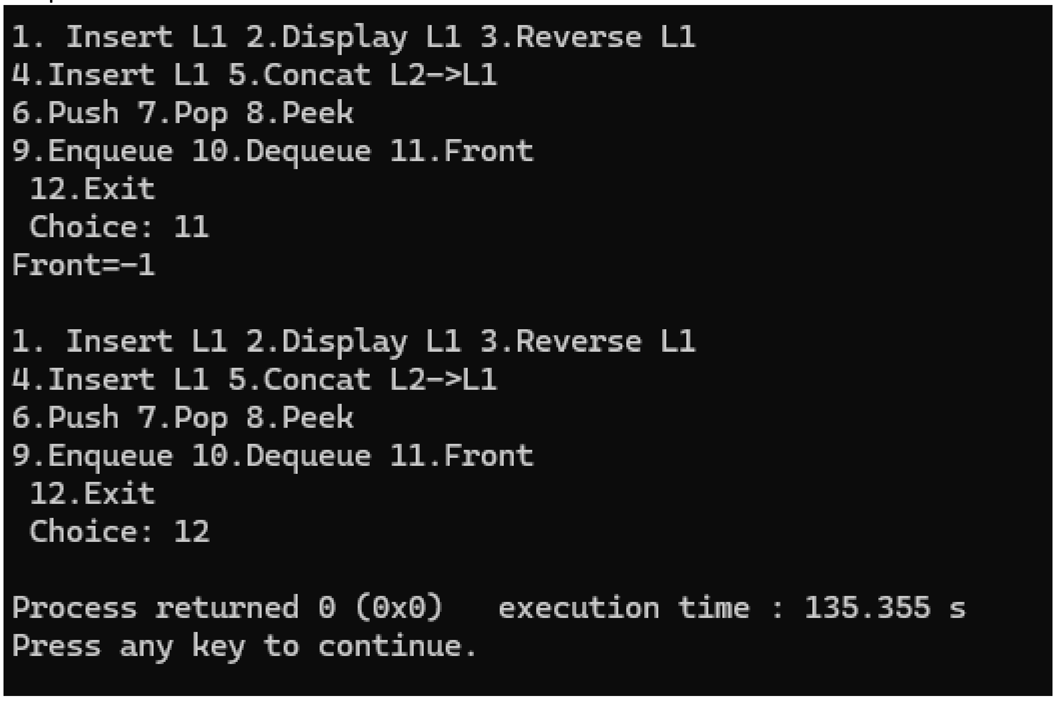


```c
void push(Node **top, int x) {Node *n=newNode(x);n->next=*top; *top=n;}
int pop(Node **top) {
if(!*top) return printf("Underflow\n"),-1;
Node *t=*top; int x=t->data; *top=t->next;
free(t); return x;
}
int peek(Node *top) { return top? top->data: -1;}

void enqueue(Node **f, Node **r, int x) {
Node *n=newNode(x);
if(!*f) *f=*r=n; else(*r)->next=n, *r=n;
}
int dequeue(Node **f, Node **r) {
if(!*f) return printf("Underflow\n"), -1;
Node *t=*f; int x=t->data;
*f=t->next; if(!*f) *r=NULL; free(t); return x;
}
int front(Node *f) {return f?f->data: -1;}

```



```c
int main() {
Node *L1=NULL, *L2=NULL, *stk=NULL, *F=NULL, *R=NULL;
int ch, x;
for(;;) {
    printf("\nl. Insert L1 2.Display L1 3.Reverse L1\n");
    printf("4.Insert L1 5.Concat L2->L1\n");
    printf("6.Push 7.Pop 8.Peek \n");
    printf("9.Enqueue 10.Dequeue 11.Front\n 12.Exit\n Choice: ");
    scanf("%d",&ch);
    switch(ch) {
    case 1: printf("Val: "); scanf("%d",&x); insertSorted(&L1,x); break;
    case 2:display(L1); break;
    case 3:reverse (&L1); break;
    case 4:printf("Val: "); scanf("%d",&x); insertSorted(&L2,x); break;
    case 5: concat(&L1, &L2); break;
    case 6:printf("Val: "); scanf("%d",&x); push(&stk, x); break;
    case 7:x=pop(&stk); if(x!=-1) printf("Popped=%d\n",x); break;
    case 8:printf("Peek=%d\n",peek(stk)); break;
    case 9:printf("Val: "); scanf("%d",&x); enqueue(&F,&R,x); break;
    case 10:x=dequeue(&F, &R); if(x!=-1) printf("dequeued=%d\n",x); break;
    case 11:printf("Front=%d\n",front(F)); break;
    case 12:exit(0);
}
}
}
```

Output-



```
1. Insert L1 2.Display L1 3.Reverse L1
4.Insert L1 5.Concat L2->L1
6.Push 7.Pop 8.Peek
9.Enqueue 10.Dequeue 11.Front
 12.Exit
 Choice: 11
Front=-1

1. Insert L1 2.Display L1 3.Reverse L1
4.Insert L1 5.Concat L2->L1
6.Push 7.Pop 8.Peek
9.Enqueue 10.Dequeue 11.Front
 12.Exit
 Choice: 12

Process returned 0 (0x0)   execution time : 135.355 s
Press any key to continue.
```