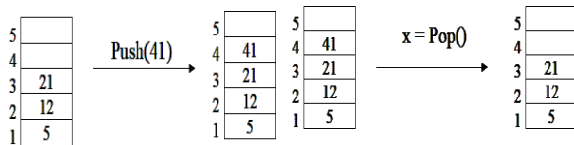


Stack, Last In First Out (LIFO) структура от данни
 Операции: `initStack()`, `push()`, `pop()`, `top()`, `isEmpty()`



функции прототипи – `stack.h` ([свали от тук](#))

```
#define STACKSIZE 50
typedef struct
{
    int item[STACKSIZE];
    int top;
} Stack;
void initStack(Stack *st);
int push(Stack *st, int elem);
int pop (Stack *st, int *elem);
int top (Stack st, int *elem);
int isEmpty(Stack st);
```

функции – код `funct.c` ([свали от тук](#))

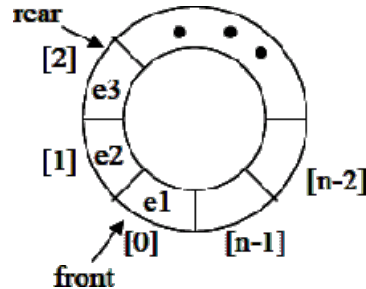
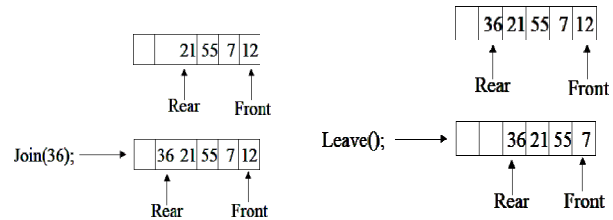
```
#ifndef STACKSIZE
#include "stack.h"
#endif
void initStack(Stack *st) { st->top=0;}
int push(Stack *st, int elem) {
    if(st->top<STACKSIZE) {
        st->item[st->top++] = elem; return 1;
    }
    return 0;
}
int pop (Stack *st, int *elem) {
    if(st->top>0) { *elem = st->item[-- st->top]; return 1; }
    return 0;
}
int top (Stack st, int *elem) {
    if(st.top>0) { *elem = st.item[st.top-1]; return 1; }
    return 0;
}
int isEmpty(Stack st) {
    if(st.top>0) return 0;
    return 1;
}
}
```

пример ([свали от тук](#)) – извеждане на текст отзад напред

```
#include <stdio.h>
#include "funct.c"
int main(int argc, char *argv[]) {
    char buf[50],*p; int chr;
    Stack st;
    printf("please input a string: ");
    fgets(buf,50,stdin); // if (!fgets...verification
    initStack(&st);
    for(p=buf; *p;p++){
        push(&st, (int)(*p)); // if(!push ...
    }
    for(; !isEmpty(st);){
        pop(&st,&chr); // if (!pop ...
        printf("%c",chr);
    }
    return 0;
}
}
```

приложения

Queue, First In First Out (FIFO) структура от данни
 Операции: `initQueue()`, `join()`, `leave()`, `int isEmpty()`



функции прототипи – `queue.h` ([свали от тук](#))

```
#define QSIZE 5
typedef struct {
    int items[QSIZE]; int front; int size;
} Queue;
void initQueue(Queue *q);
int join(Queue *q, int elem);
int leave(Queue *q, int *elem);
int isEmpty(Queue q);
```

функции – код `functq.c` ([свали от тук](#))

```
#include "queue.h"
void initQueue(Queue *q) { q->front=q->size=0;}
int join(Queue *q, int elem) {
    if(q->size<QSIZE) {
        q->items[(q->front+ q->size++)%QSIZE]=elem;
        return 1;
    }
    return 0;
}
int leave(Queue *q, int *elem) {
    if(q->size>0) {
        *elem=q->items[q->front++];
        q->size--;
        q->front%=QSIZE;
        return 1;
    }
    return 0;
}
int isEmpty(Queue q) {
    if(q.size>0) return 0;
    else return 1;
};
```