

28.4.2011 Някои функции за работа със списък

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct {
    char name [30];
    int n;
}Person;
typedef struct node {
    Person prs;
    struct node *next;
}node;

node * push(node * head, Person p){
    node *hlp;
    if(!(hlp= (node *)malloc(sizeof(node)))) { /*...*/ }
    hlp->prs=p;
    hlp->next=head;
    head=hlp;
    return head;
}

node * pull(node *head, Person *p){
    node *hlp=head;
    if(!head) { /* ... */ }
    *p=head->prs;
    head=head->next;
    free(hlp);
    return head;
}

Person peek(node *head){
    if(!head) { /* ... */ }
    return head->prs;
}

node * last(node *head, Person *p){
    node *hlp=head;
    if(!head) { /* ... */ }
    if(!(head->next)){
        *p=head->prs;
        head=head->next;
        free(hlp);
    }
    else{
        for(hlp=head;hlp->next->next;hlp=hlp->next);
        *p=hlp->next->prs;
        free(hlp->next);
        hlp->next=NULL;
    }
    return head;
}

void prt(node *crnt){
    printf("\nthe list is:\n");
    while(crnt){
        printf("%s has %d\n",crnt->prs.name,crnt->prs.n);
        crnt=crnt->next;
    }
}
```

```
int main(){
    FILE *f;
    node *head=NULL;
    Person p;
    if (!(f=fopen("D:\\Work\\person.dat","rb"))){ return 1; }
    do{
        if(!fread(&p,sizeof(p),1,f)) break;
        head=push(head,p);
    }while(1);
    prt(head);
    head = pull(head,&p);
    printf("\n pulled: %s has %d\n",p.name,p.n);
    prt(head);
    p=peek(head);
    printf("\n peeked: %s has %d\n",p.name,p.n);
    prt(head);
    head = last(head,&p);
    printf("\n the last element: %s has %d\n",p.name,p.n);
    prt(head); return 0;
}
```

Побитови операции:

&	Bitwise AND	Побитово И
	Bitwise OR	Побитово ИЛИ
^	Bitwise Exclusive-OR	Побитова сума по модул 2 (побитово изключващо ИЛИ)
<<	Bitwise left shift	Побитово местене в ляво
>>	Bitwise right shift	Побитово местене в дясно
~	Bitwise complement	Побитово отрицание

01001011	01001011	01001011	01001011
00010101	00010101	00010101	2
&		^	<<
-----	-----	-----	-----
00000001	01011111	01011110	00101100
10001011	10001011	01001011	~01001011
2	2	2	-----
>>	>>	>>	10110100
-----	-----	-----	
00100010	11100010	00010010	
unsigned	signed	signed	

<pre>#include <stdio.h> #include <string.h> int main(){ char buf[] = "abraKADABRA"; char buf1[] = "abraKADABRA", *p; unsigned mask = 'A'^'a'; for(p=buf1;*p;p++) *p &= ~mask; printf("%s\n",buf1); strcpy(buf1,buf); for(p=buf1;*p;p++) *p = mask; printf("%s\n",buf1); strcpy(buf1,buf); for(p=buf1;*p;p++) *p ^= mask; printf("%s\n",buf1); return 0; }</pre>	<pre>ABRAKADABRA abrakadabra ABRAkadabra</pre>
--	--