

```

Някои функции за работа със списък (свали програмата от тук)
#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;

/* въвеждане нов възел в началото на списъка */
node * push(node * head, Person p){
    node *hlp;
    if(!(hlp=(node *)malloc(sizeof(node)))) { /*...*/}
    hlp->prs=p;
    hlp->next=head;
    head=hlp;
    return head; /* резултат - ново начало на списъка */
}

/* извеждане възел от началото на списъка-в 2-я параметър*/
node * pull(node *head, Person *p){
    node *hlp=head;
    if(!head) { /* ... */}
    *p=head->prs;
    head=head->next;
    free(hlp);
    return head; /* резултат - ново начало на списъка */
}

/* връща първия елемент от списъка */
Person peek(node *head){
    if(!head) { /* ... */}
    return head->prs;
}

/* извеждане на последен елемент */
node * last(node *head, Person *p){
    node *hlp=head;
    if(!head) { /* ... */} /* ако в списъка няма елементи */
    if(!(head->next)){ /* ако в списъка има само 1 елемент */
        *p=head->prs;
        head=head->next;
        free(hlp);
    }
    else { /* hlp се насочва към предпоследния елемент */
        for(hlp=head;hlp->next->next;hlp=hlp->next);
        *p=hlp->next->prs; /* последният елемент се извежда */
        free(hlp->next);
        hlp->next=NULL;
    }
    return head; /* резултат - ново начало на списъка */
}

/* отпечатване на списъка */
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;
    }
}

/* извеждане на възел от позиция index в списъка */
node * delInd(node *head,int index,Person *p){
    node * prev=NULL,*crnt=head;
    int i=0; /* насочване на crnt към елемента */
    for(;crnt&&(i<index);i++,prev=crnt,crnt=crnt->next);
    if(crnt){ /* достигнат index */
        *p=crnt->prs;
        if(crnt == head){
            head=crnt->next;
        }
        else {
            prev->next=crnt->next;
        }
        free(crnt);
    }
    else { /* достигнат края на списъка преди index */
        printf("no such index in the list");
    }
    return head;
}

int main(){
    FILE *f;
    node *head=NULL;
    int index;
    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\n",p.name,p.n);
    prt(head);
    p=peek(head);
    printf("\n peeked: %s has %d\n\n",p.name,p.n);
    prt(head);
    printf("\n removing the last element");
    head = last(head,&p);
    printf("\n the last element: %s has %d\n\n",p.name,p.n);
    prt(head);
    printf("\n give a index to remove: ");
    scanf("%d",&index);
    head = delInd(head,index,&p);
    printf("\n the removed element: %s has %d\n\n",p.name,p.n);
    prt(head);
    return 0;
}

```

Примерен резултат при стартиране:

the list is:

Kiril has 43
Kira has 34
Sima has 19
Penda has 22
Pijo has 27

pulled: Kiril has 43

the list is:

Kira has 34
Sima has 19
Penda has 22
Pijo has 27

peeked: Kira has 34

the list is:

Kira has 34
Sima has 19
Penda has 22
Pijo has 27

removing the last element
the last element: Pijo has 27

the list is:

Kira has 34
Sima has 19
Penda has 22

give a index to remove: 1

the removed element: Sima has 19

the list is:

Kira has 34
Penda has 22

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	

(свали програмата от тук)

```
#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;
}
```

ABRAKADABRA
abrakadabra
ABRAkadabra