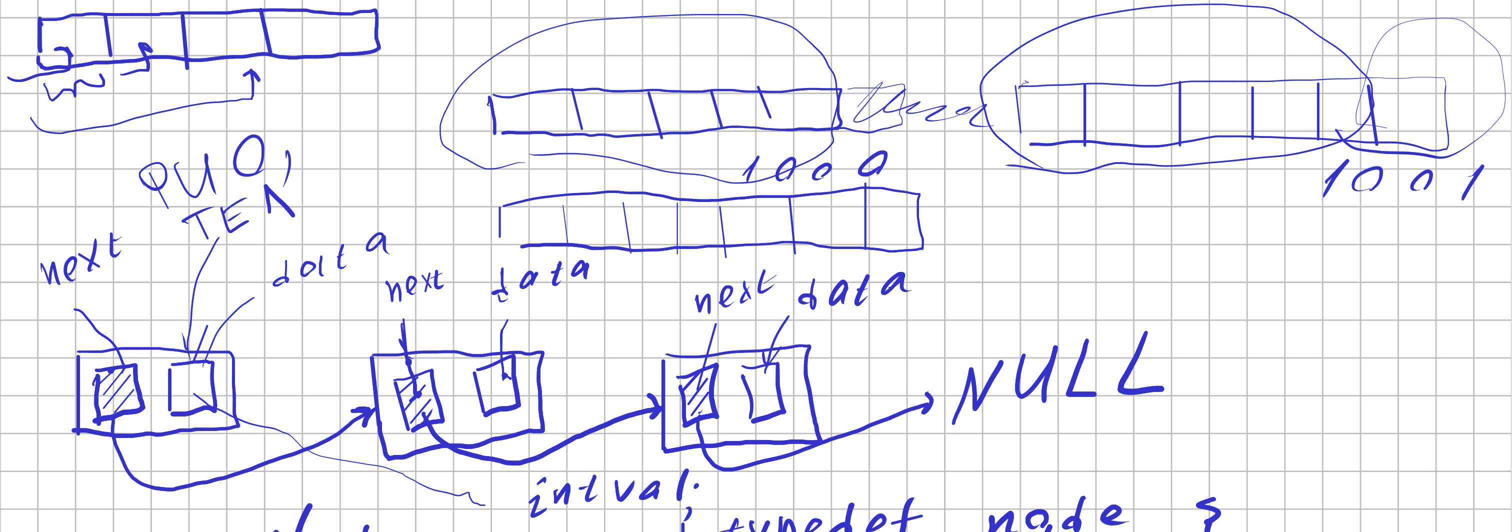
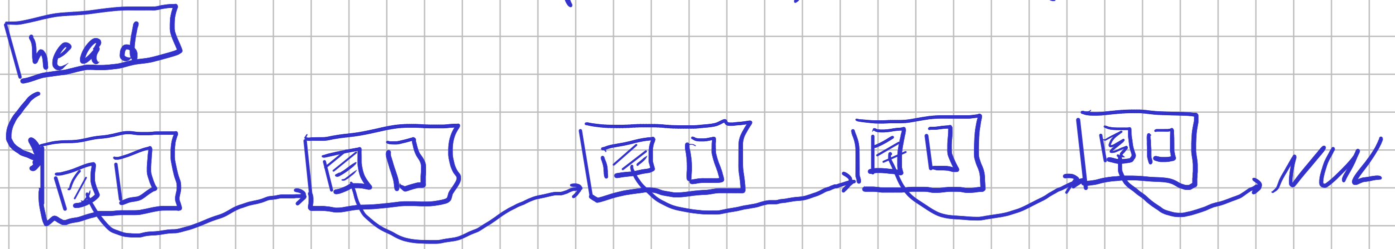


```
struct Node {
    int val;
    struct Node* next;
};
```

```
typedef node {
    int val;
    struct node* next;
} Node;
```

```
Node* head;            head = (Node*) malloc(sizeof(
head = NULL;                   Node));

head->val = 3;
head->next = NULL;
head->next = (Node*) malloc(sizeof(Node));
```

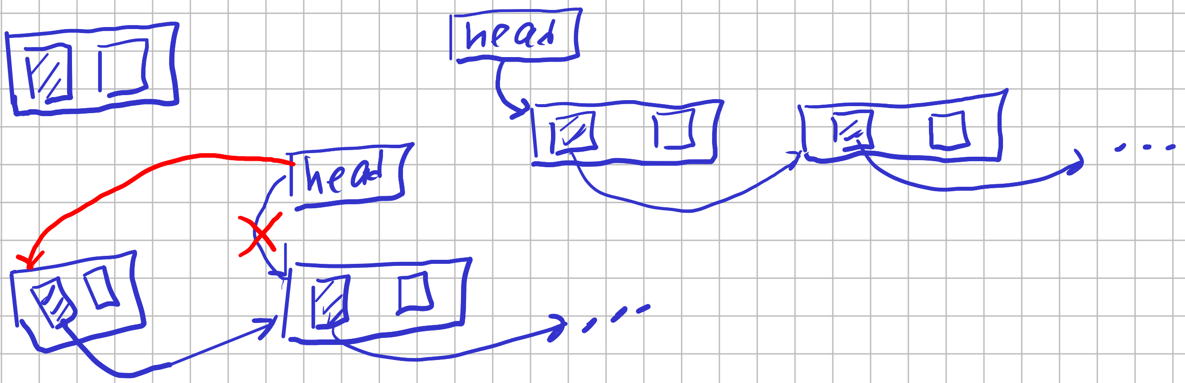
0.



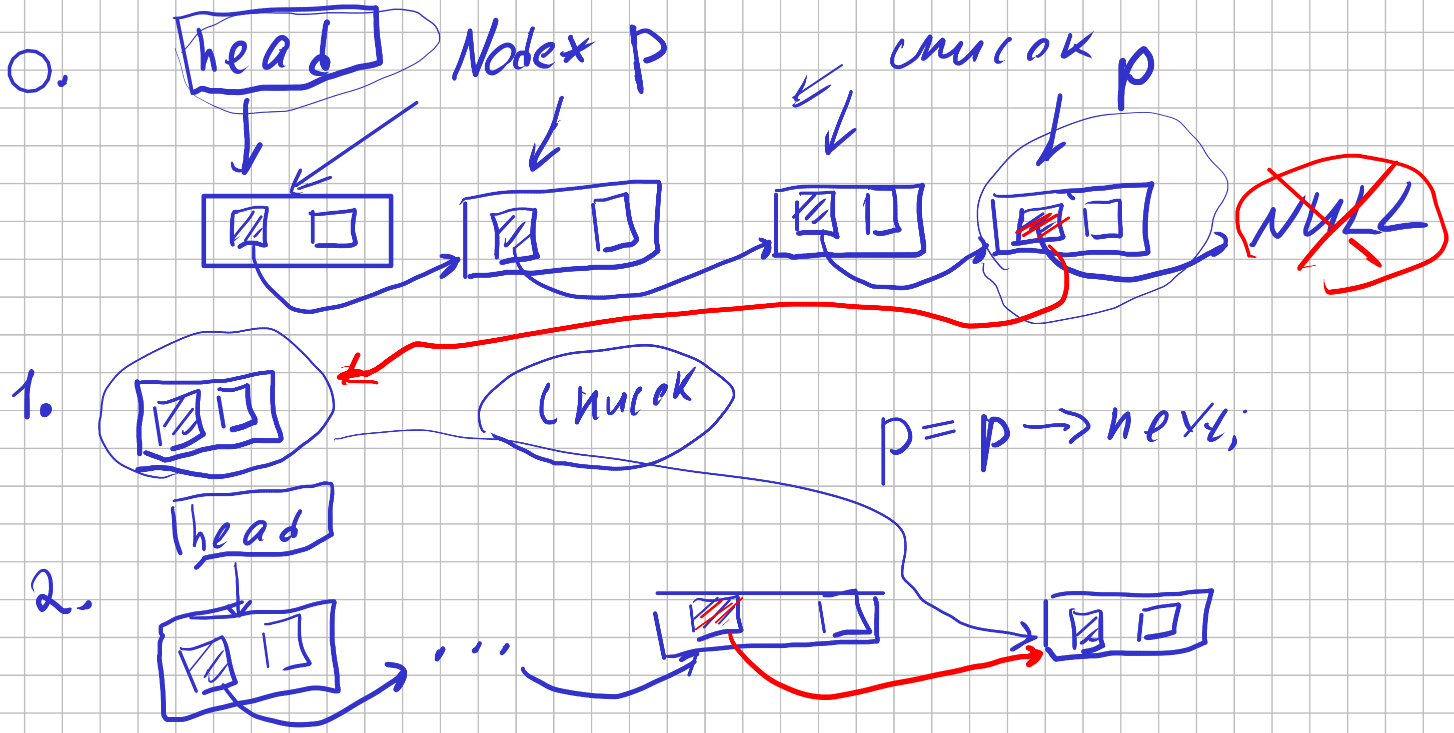
добавление в начало

1.

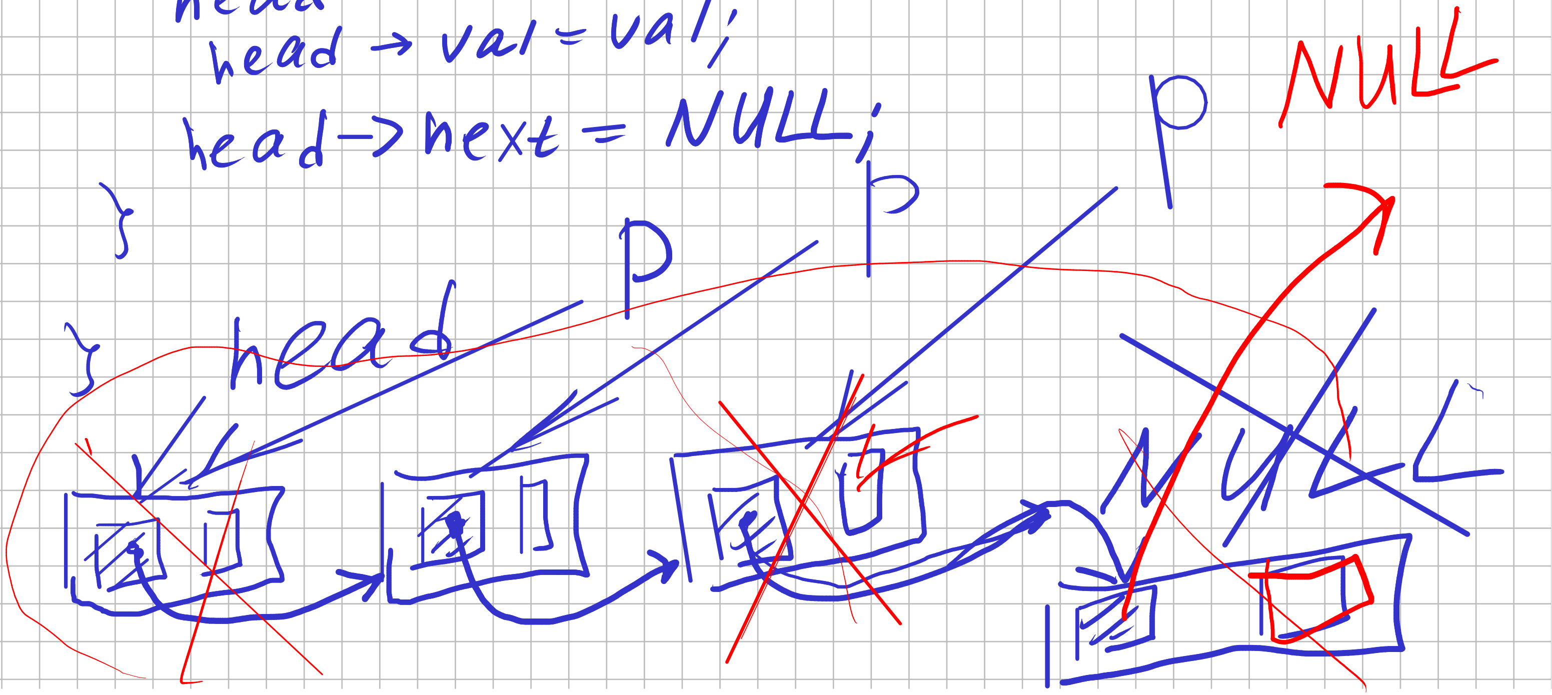
2.



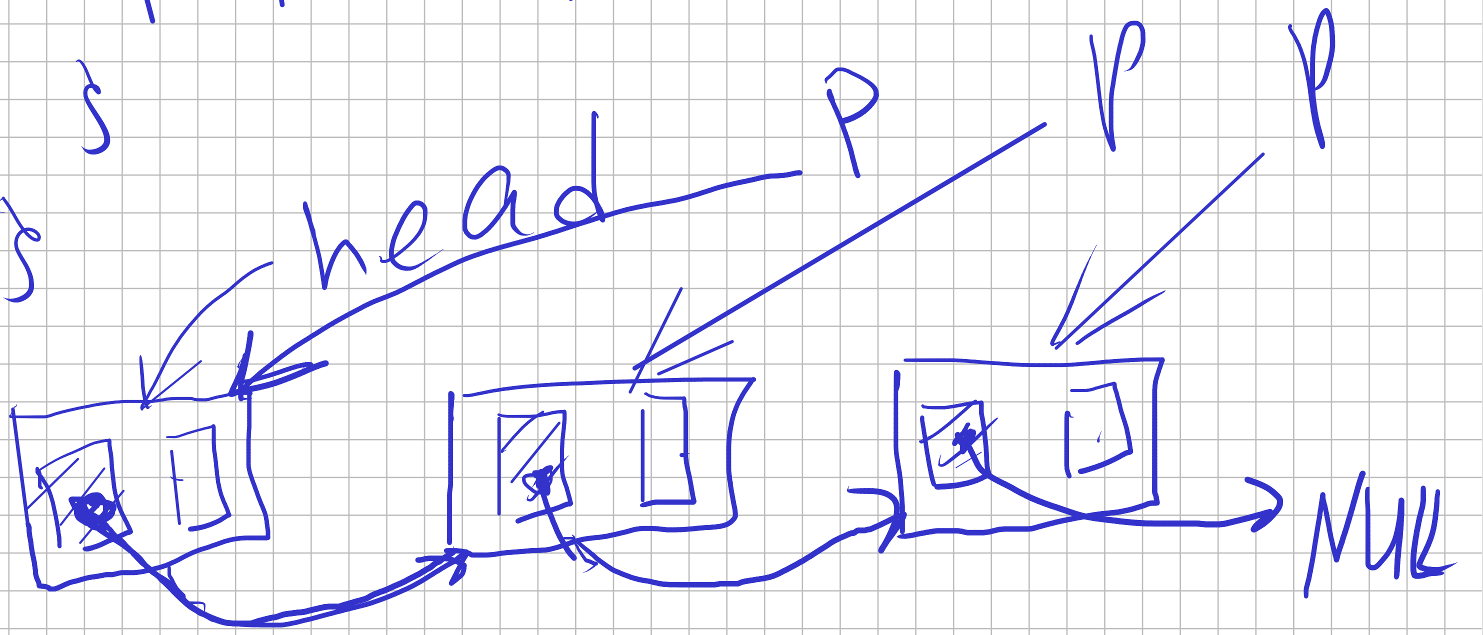
3.


0.
head
Node* p
список
p
NULL
1.
список
p=p->next;
head
2.
...

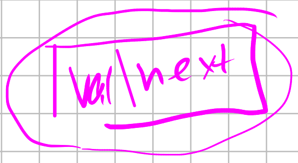
```
typedef struct node {
        int val;
        struct node* next;
      } Node;

Node* head = NULL;
void push_back(Node* head, int val) {
  if (head != NULL) {
    Node* p = head;
    while (p-> next != NULL) p = p->next;
     p->next = (Node*) malloc(sizeof (Node));
      p->next-> val = val;
      p->next->next = NULL;
  } else {
      head = (Node*) malloc(sizeof(Node));
       head -> val = val;
      head->next = NULL;
   }
}
```



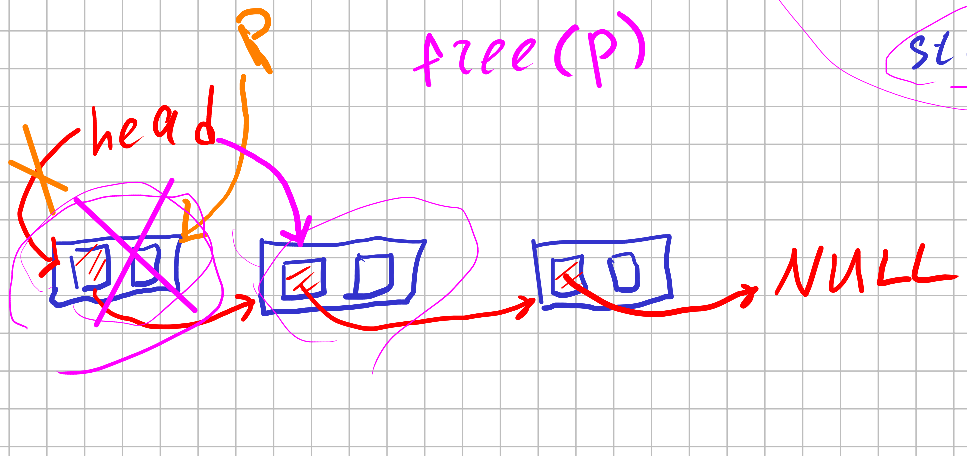
```
void print(      Node* head) {
  Node* p = head;
  while(p!=NULL) {
    printf("%d\n", p->val);
    p=p->next;
  }
}
```



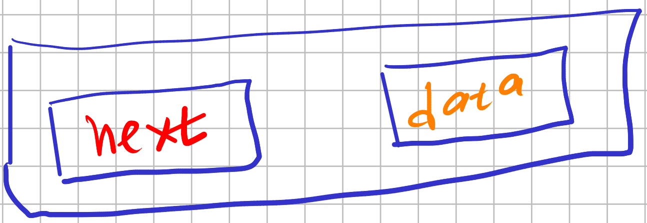
```
void del_head( Node* head) {
  ...
}
```

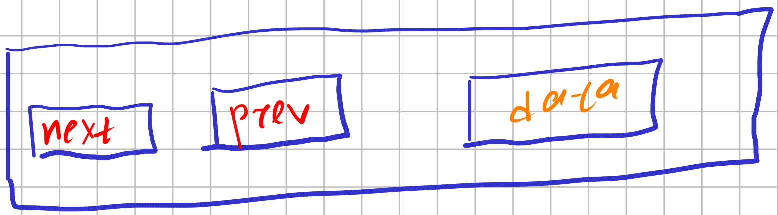


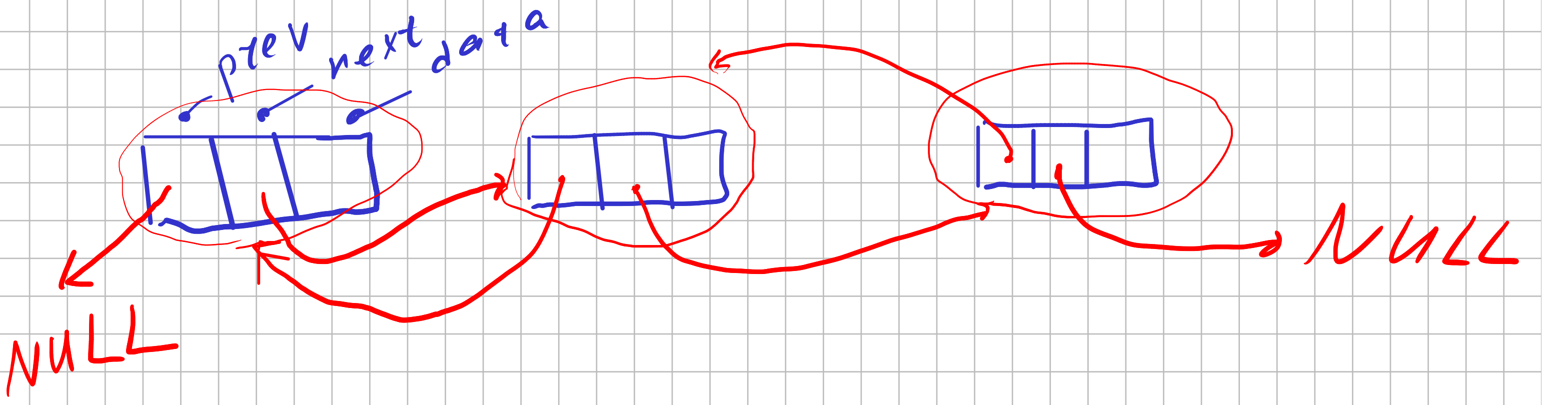
```
struct node {
  int val;
  struct node* next;
}
```

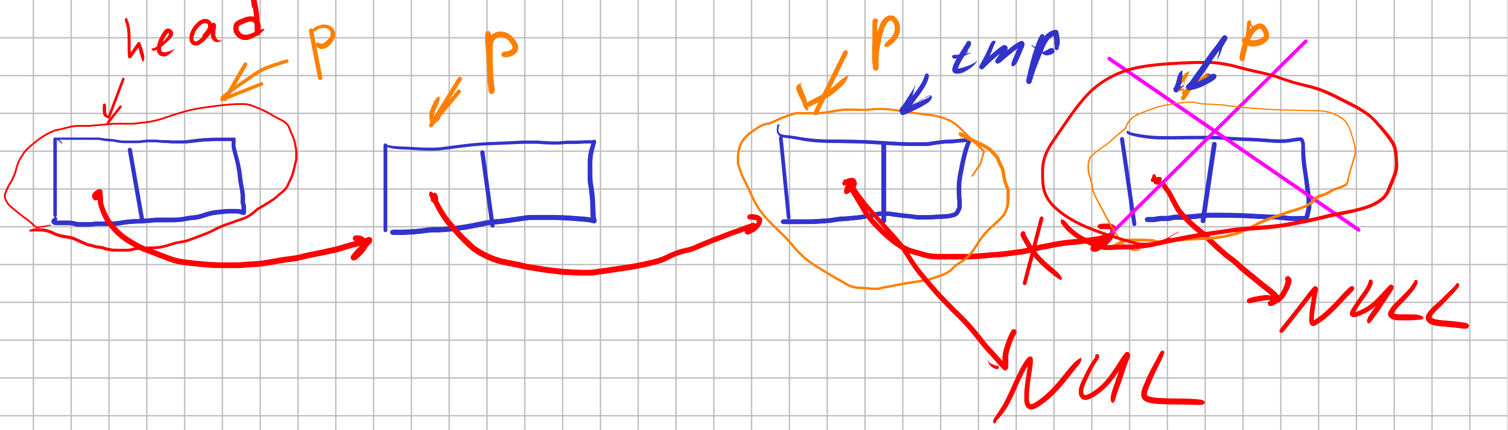


```
void delhead( Node* head) {
 Node* p = head;
 head = head -> next;
  free(p);
}
```









```
void del_tail( Node* head) {
   if( head == NULL) return;
    if( head->next == NULL) {
        free (head);
          head = NULL;
          return;
    }
    Node* p = head;
     Node * tmp;
     while ( p-> next != NULL) {
       tmp = p;
        p = p->next;
      }
      tmp->next = NULL;
       free(p);
    }
```