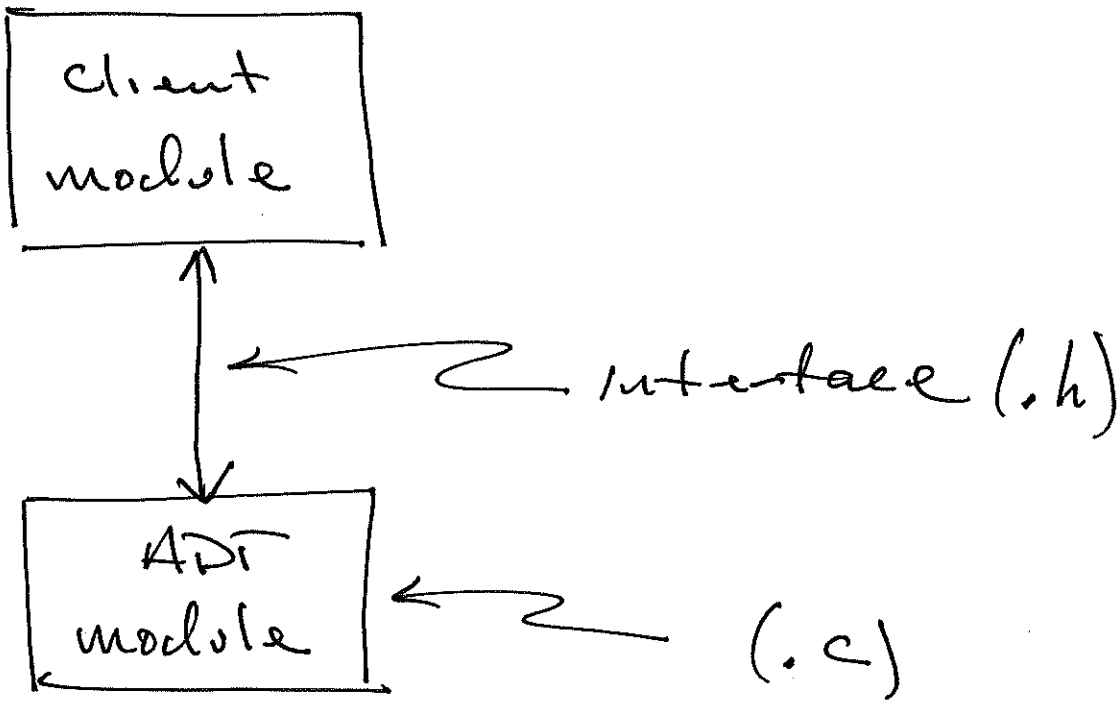
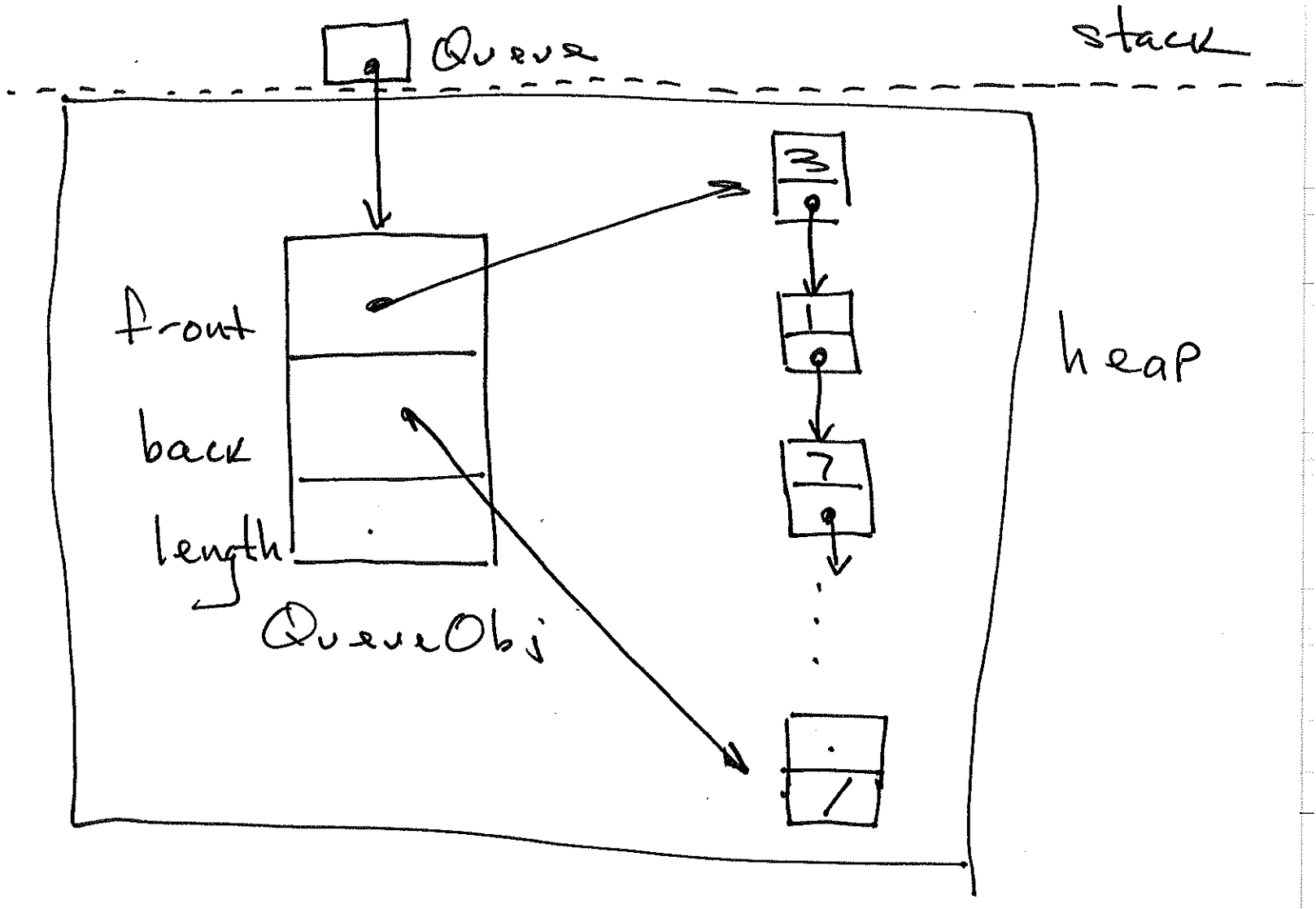
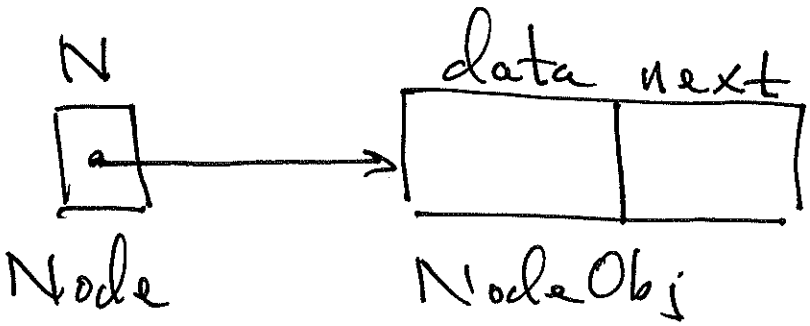
CSE 101-02 1-12-23

In C



Client view (3, 1, 7, ...)





How to create new types:

```
typedef A B;
```

↑ now an alias for A.

```
typedef struct NodeObj {
    int data;
    struct NodeObj* next;
} NodeObj;
```

A = `struct NodeObj {...}`, B = `NodeObj`

```
typedef NodeObj* Node;
```

A = `NodeObj*`, B = `Node`

Or

```
typedef struct NodeObj* Node;

typedef struct NodeObj{
    int data
    Node next;
} NodeObj;
```

in Queue.h

```
// types
typedef int QueueElement;
typedef struct QueueObj* Queue;

// constructors & destructors

Queue newQueue();
Void freeQueue(Queue* pQ);

// main proc.

// access fcn.

// other fcns.
```

in Queue.c

```
#....

#include "Queue.h"

// types

typedef struct NodeObj* Node

typedef
   :
   :

} NodeObj;

typedef struct                {

    Node front;
    Node back;
    int length;

} QueueObj;
```

```
Node newNode(QueueElement x){
    Node N = malloc(sizeof(NodeObj));
    N->data = x;          assert(N!=NULL);
    N->next = NULL;
    return(N);
}
```

note: in C and C++

```
N->data
```

same as

```
(*N).data
```

↑ Pointer dereference (value at)

↑ member selection

* Operator

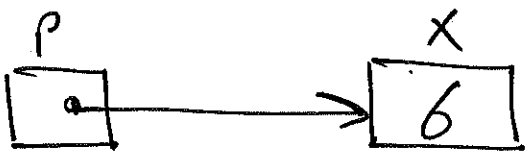
(1) 3 * 5 multiplication

(2) int * Pointer-to int

Ex. int* p;

int x = 6;

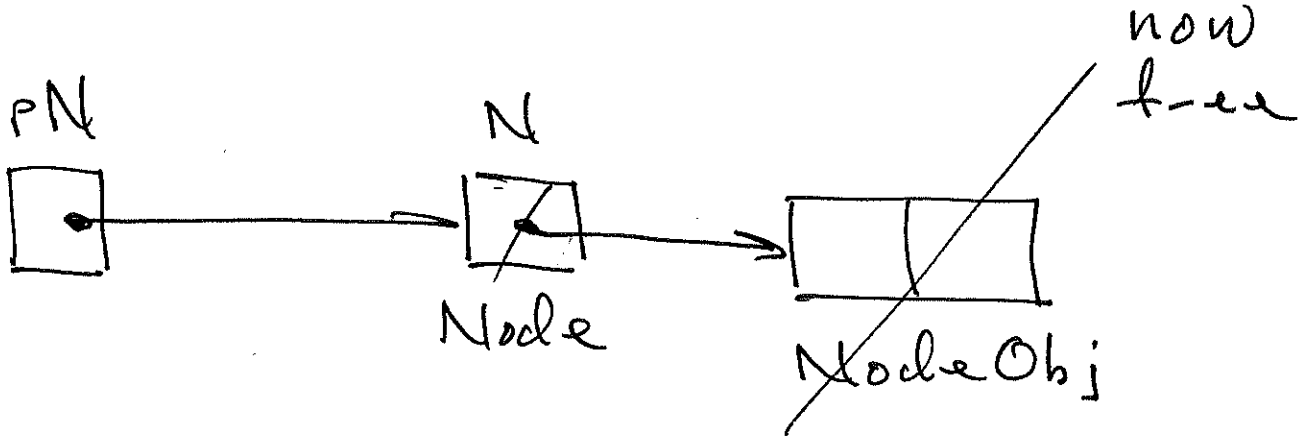
p = &x; address of



(3) *p value at or dereference.

Ex printf("%d", *p)

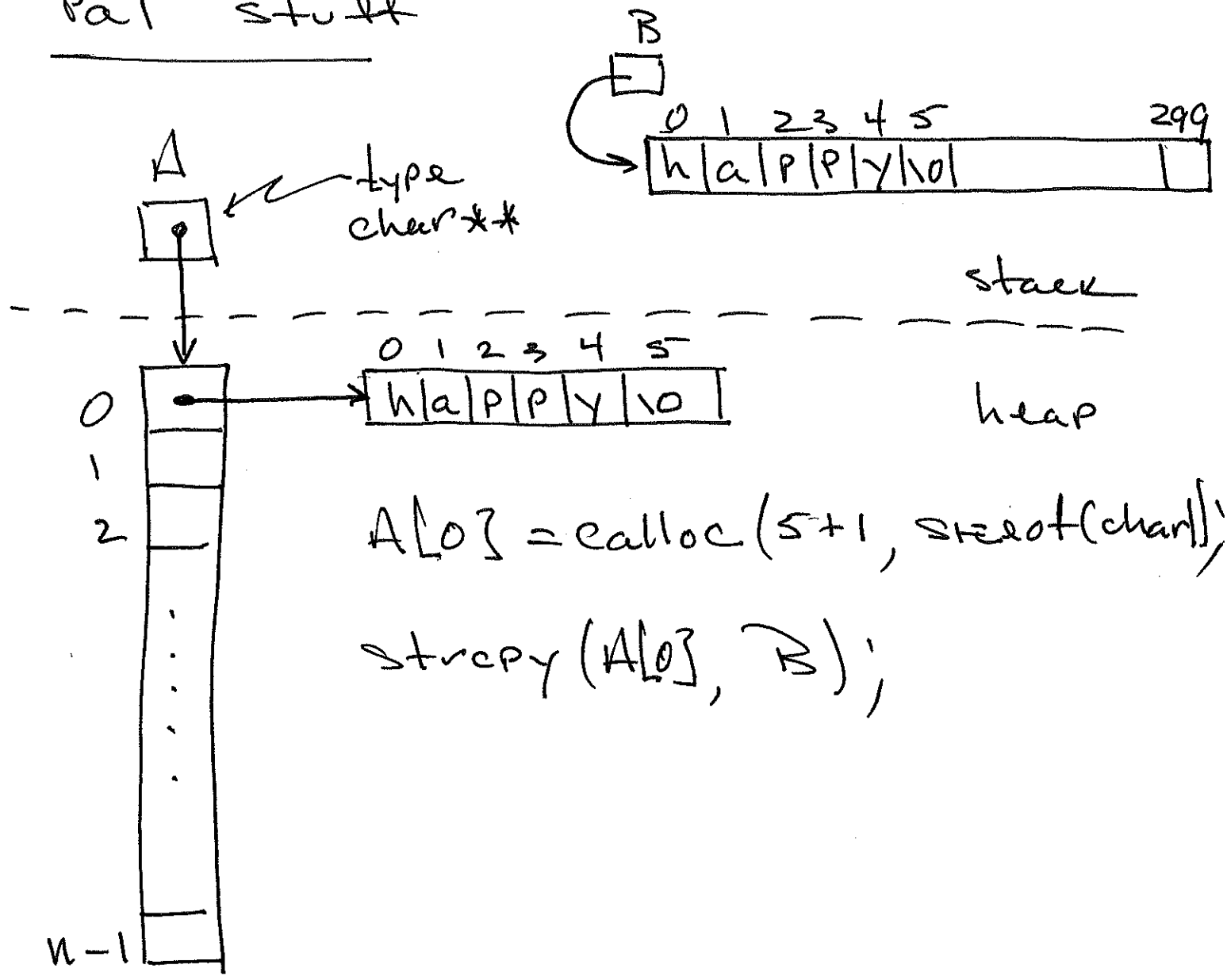
```
void freeNode(Node * pN){
    free (*pN);
    *pN = NULL;
}
```



Call :

```
freeNode(&N);
```

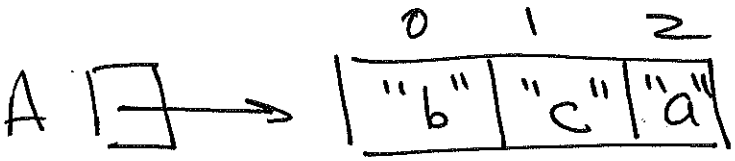
Pal stuff



A[0] = calloc(5+1, sizeof(char));

strcpy(A[0], B);

A = calloc(n, sizeof(char*));



L = (2, 0, 1)