

to create Node type:

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

```
typedef NodeObj * Node;
```

can also reverse order. See
Queue example.

what if an ADT op. is called in violation of one of its stated Preconditions.

Policy.

- (1) All ADT ops state their Pre-cond in a comment block, in terms of access function calls.
- (2) All ADT ops. must check its Pre cond. (s) before nominal execution.
- (3) If a Pre cond. is violated, the ADT op. must Print an error msg. to stderr, giving
name of ADT, name of op, precond.
Then exit Program.

Note: in C, we have one additional precond. on every ADT op.

Ret var != NULL

test this first!!!

RMK.

Test each precond. in its own stand-alone if stmt, then nominal execution should begin at top level of fun.

Rule.

When testing an ADT, exercise every logical pathway within the code, in each ADT op.

Read: P.T Naming conventions.

especially how to write a destructor in C.

Pa1 commente :

LS

• Variable length arrays (VLA) in C

its possible to do :

```
int n;
```

```
// get n from user input
```

```
int A[n]; // VLA
```

```
for (int i=0; i<n; i++){
```

```
    A[i] = .....  
}
```

Don't Do this .

Instead do this:

```
int n;
```

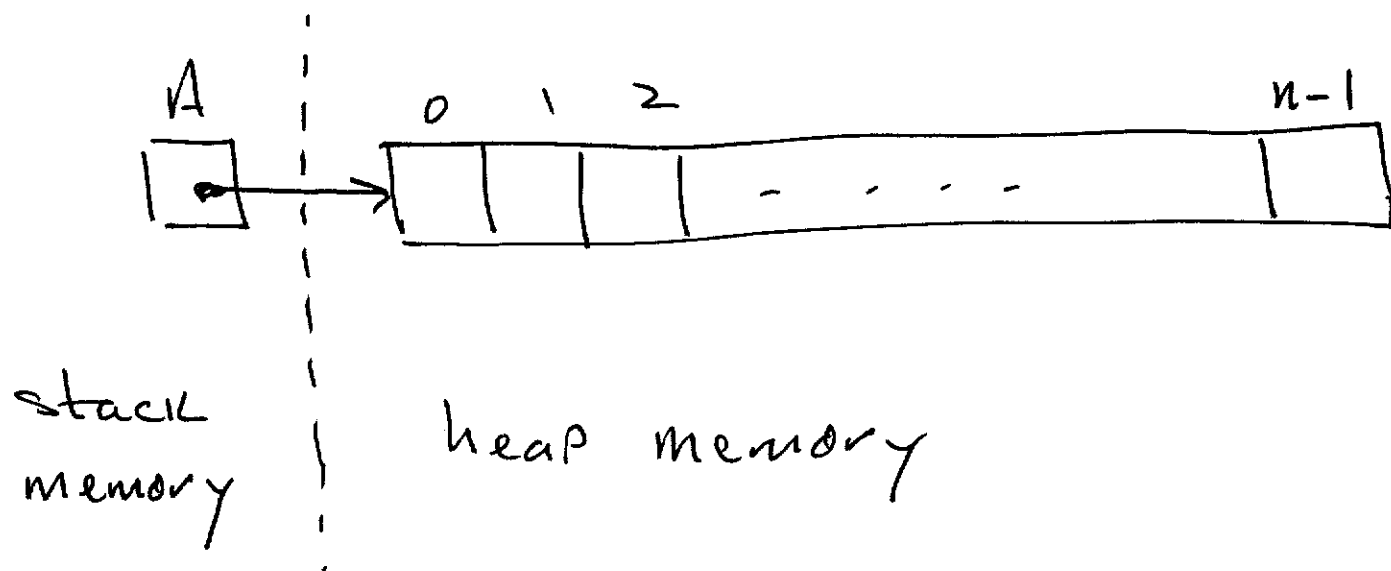
```
// get n
```

```
int * A = calloc(n, sizeof(int));
```

or

```
int * A = malloc(n * sizeof(int))
```

Picture



→ List ADT

client view : $(2, \underline{-1}, 5)$

inside view:

