

D.S.

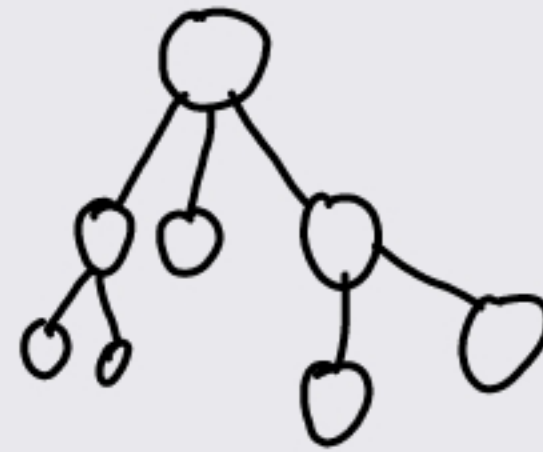
1.
  - Static (Compile-Time storage)
  - Dynamic (Runtime allocation)

int a[20];

int \*a;  
int n;

int \*a = new int[n];

2.
  - Linear (List, Queue, Stack)
  - Non-Linear (Tree, Binary Tree, Graph)



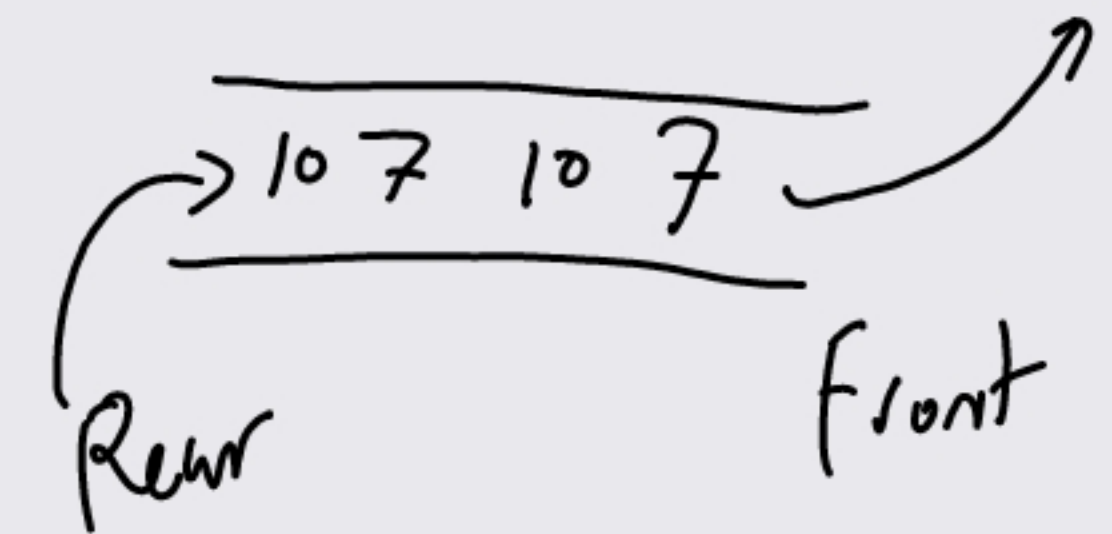
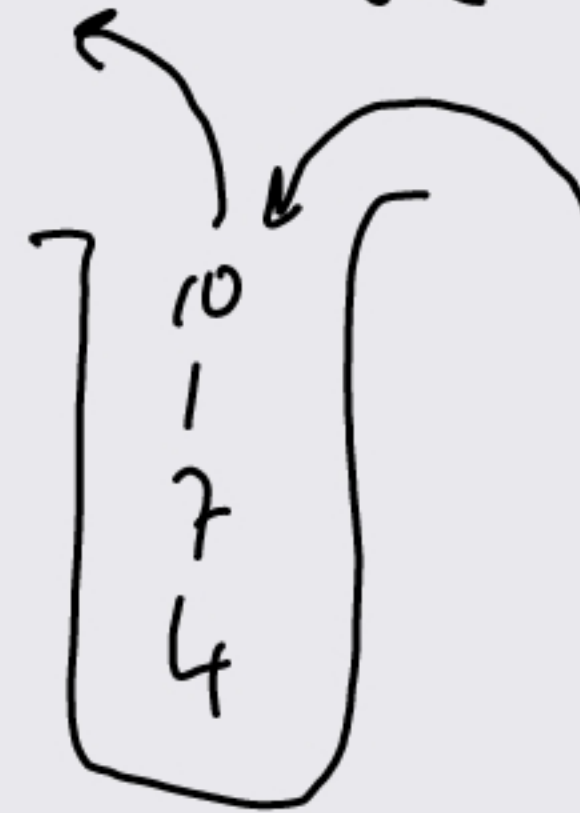
cin >> n;

if (n > 0)  
a = new int[n];

else

a = nullptr;

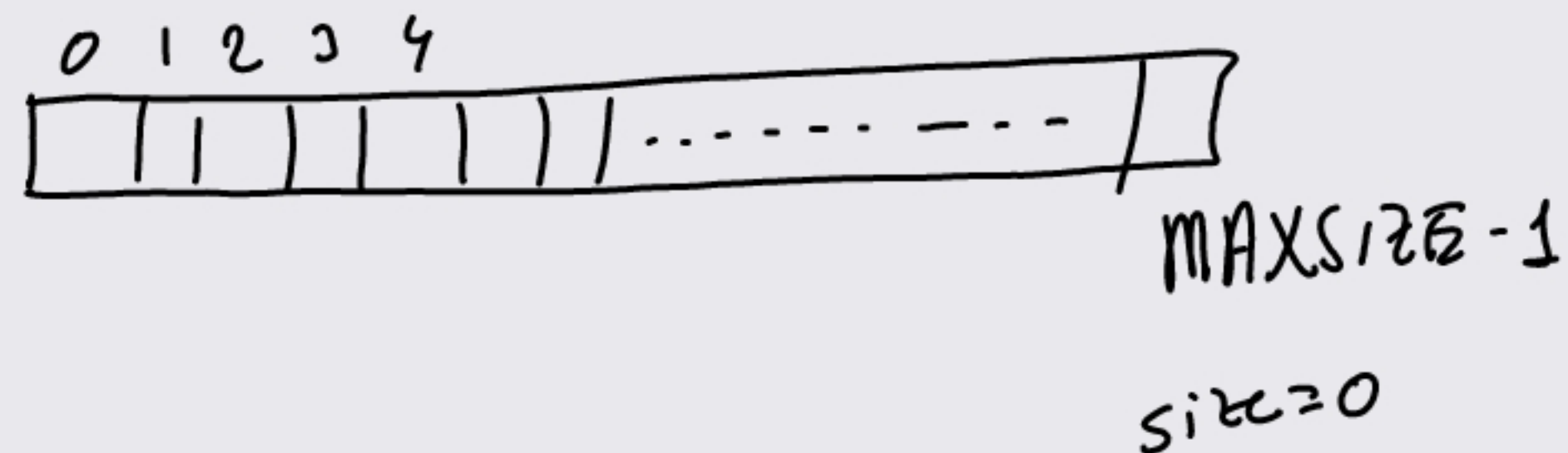
3.
  - Restricted (Stack, Queue LIFO, FIFO)
  - Non Restricted (List)



# int List ADT

- initialize empty List
- append element
- isMember
- insert At
- delete At
- delete (elem)
- clear

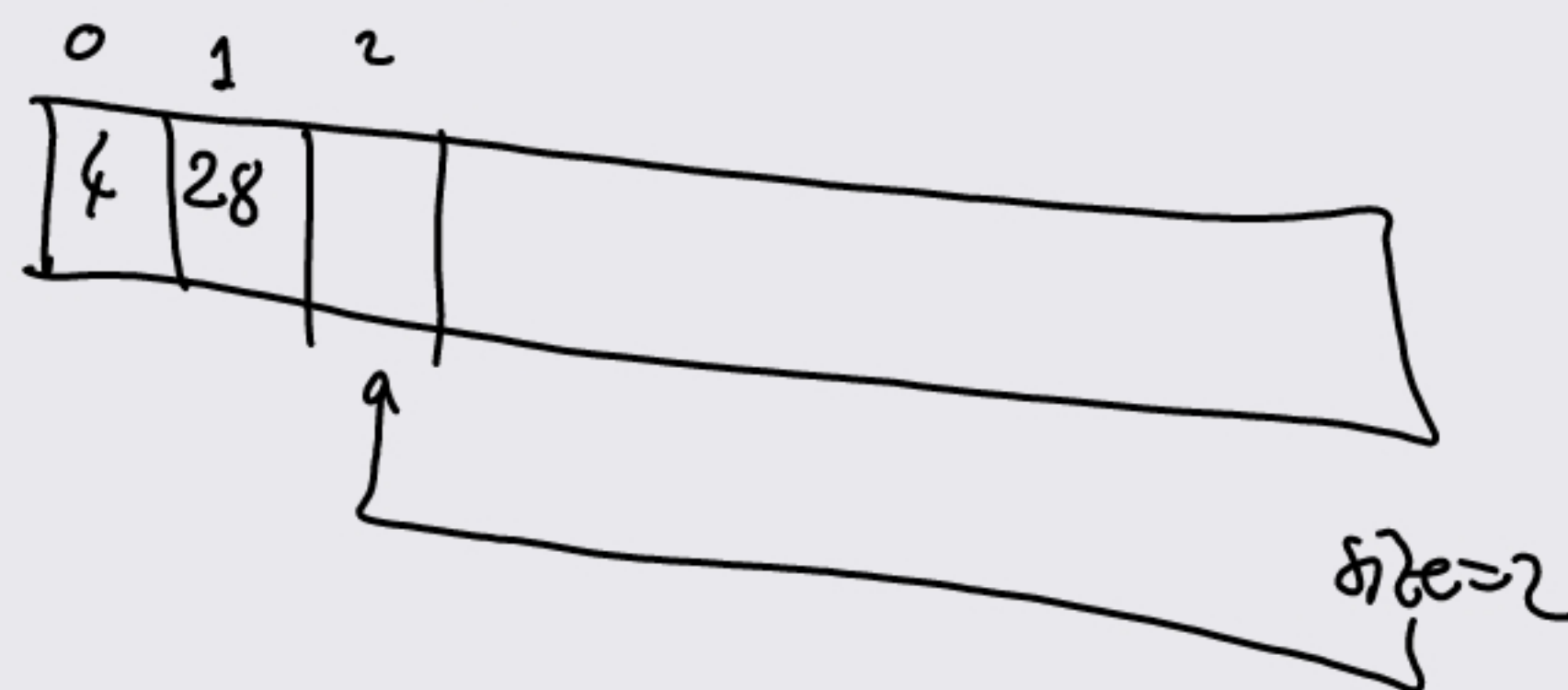
array based intlist impl.



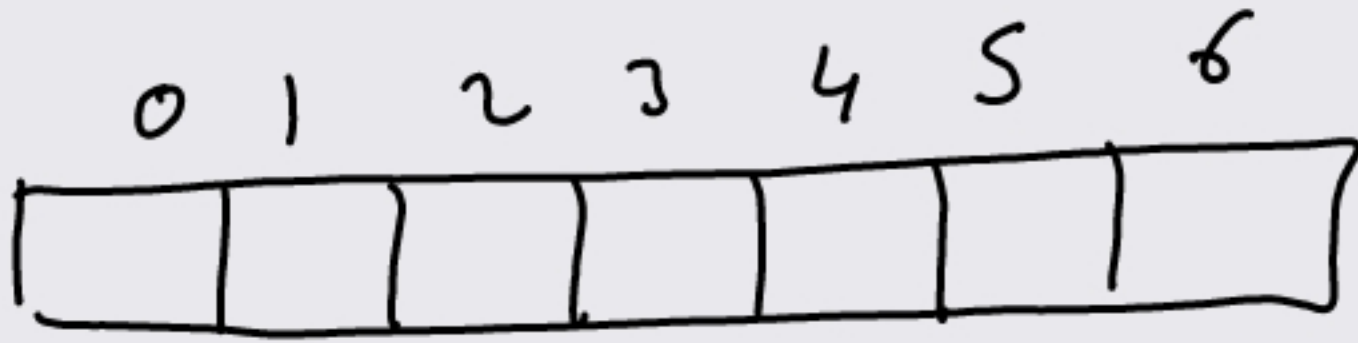
append(4)



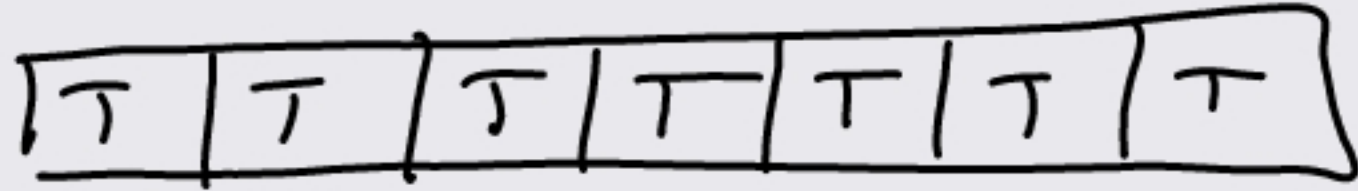
append(28)



items

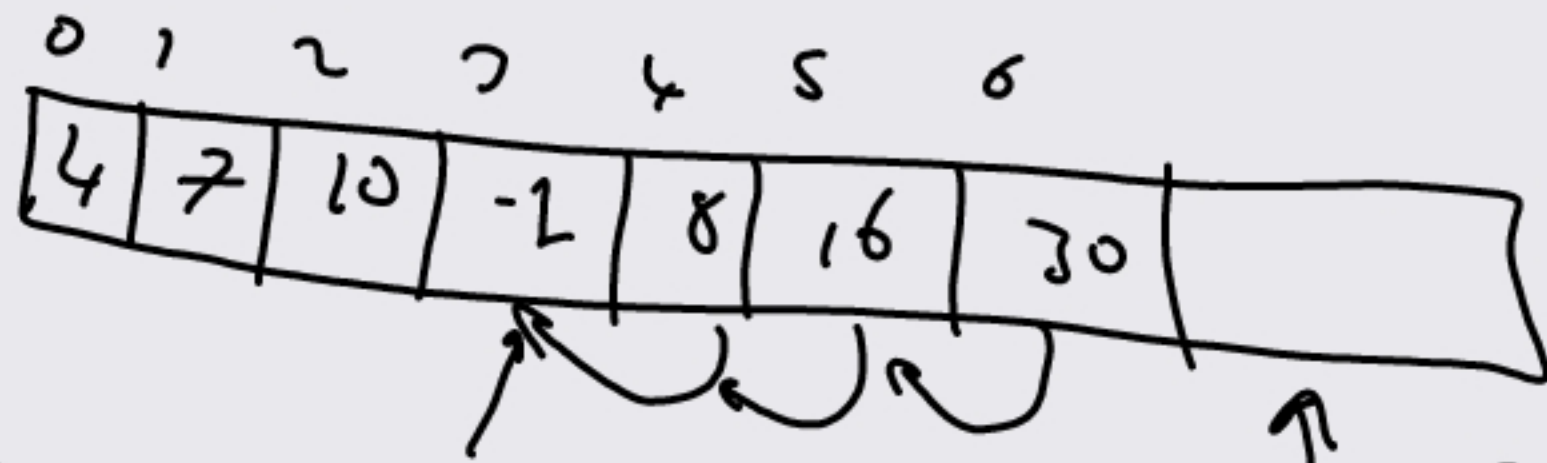


isVacant



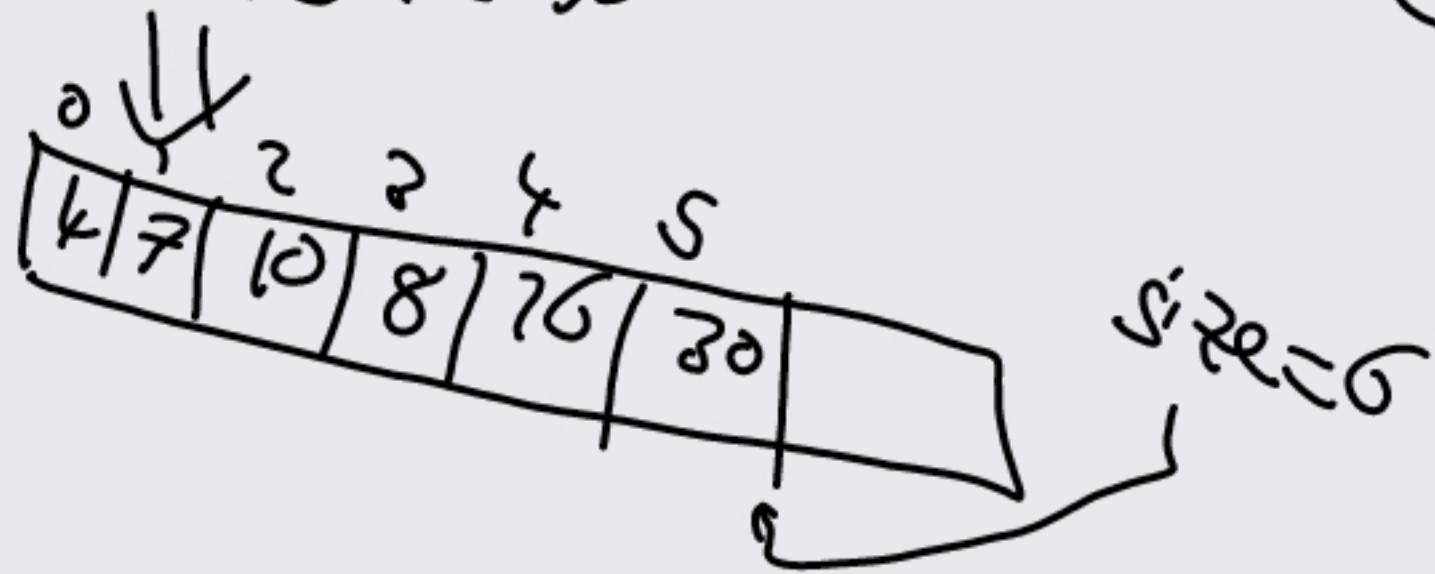
bool array

void deleteAt(int idx)



size=7

deleteAt(3);

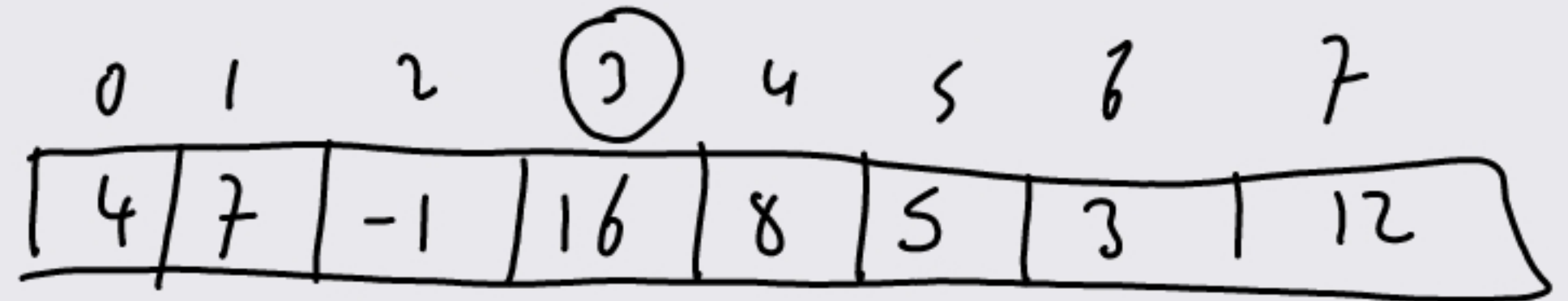


size=6

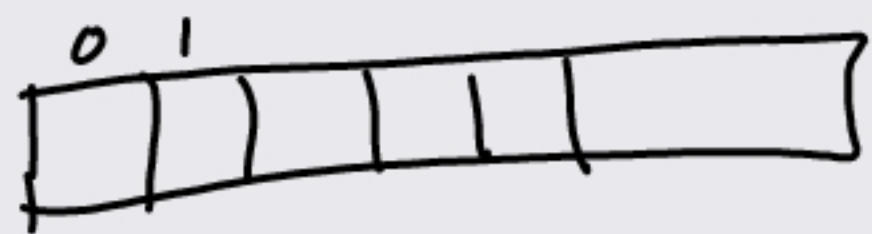
for(int i=idx; i<size-1; i++)

items[i]=items[i+1];

deleteItem(16);

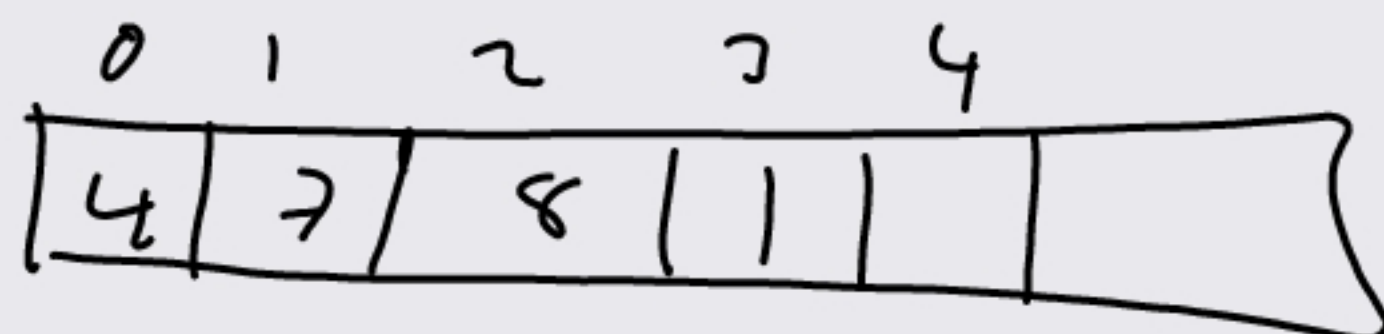


↑



Size = 0

① insertAt(0) is valid, insertAt(7) is not valid

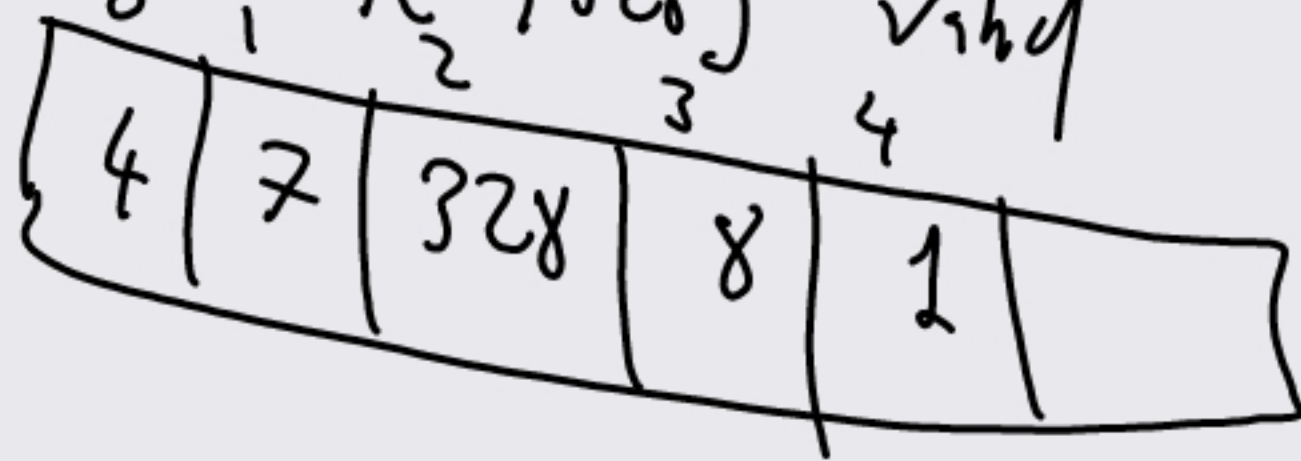


Size = 5

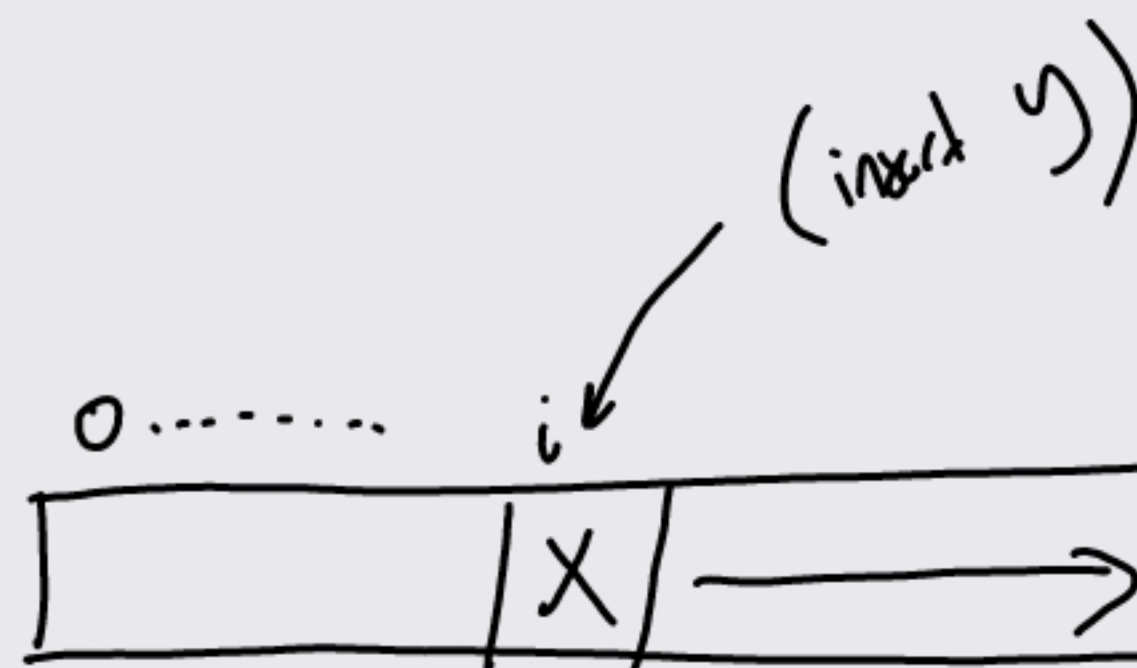
Valid insertion [0...4]

insertAt(7, 328) not valid  
 idx ↗ elem ↘

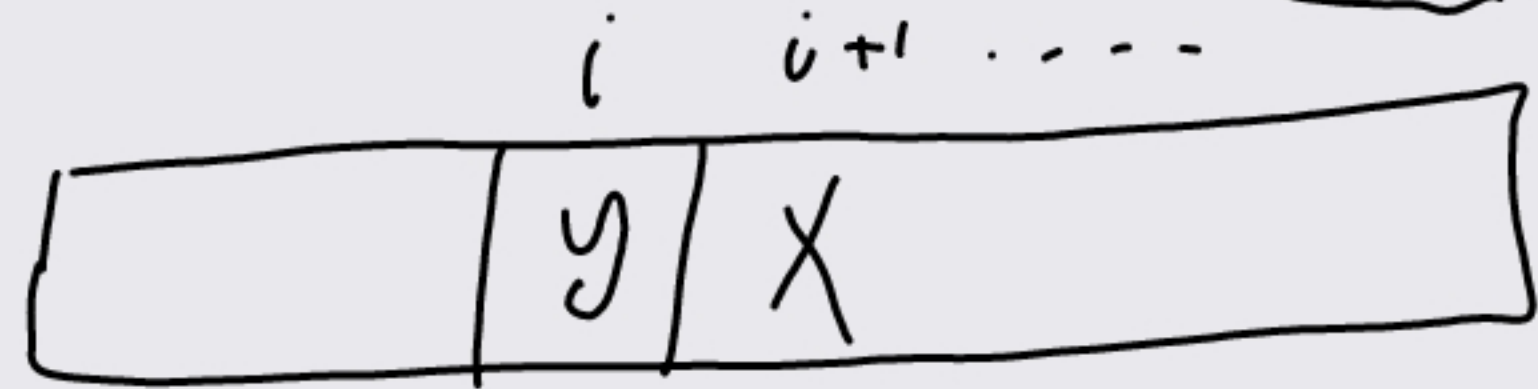
insertAt(2, 328) valid  
 0 1 2 3 4



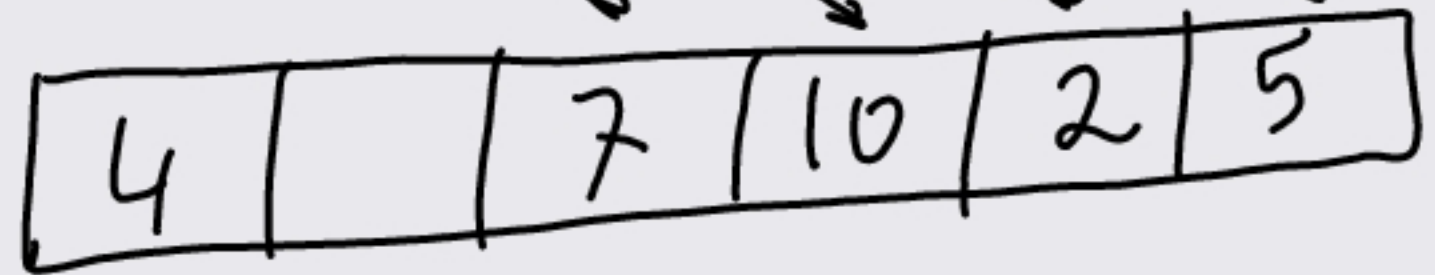
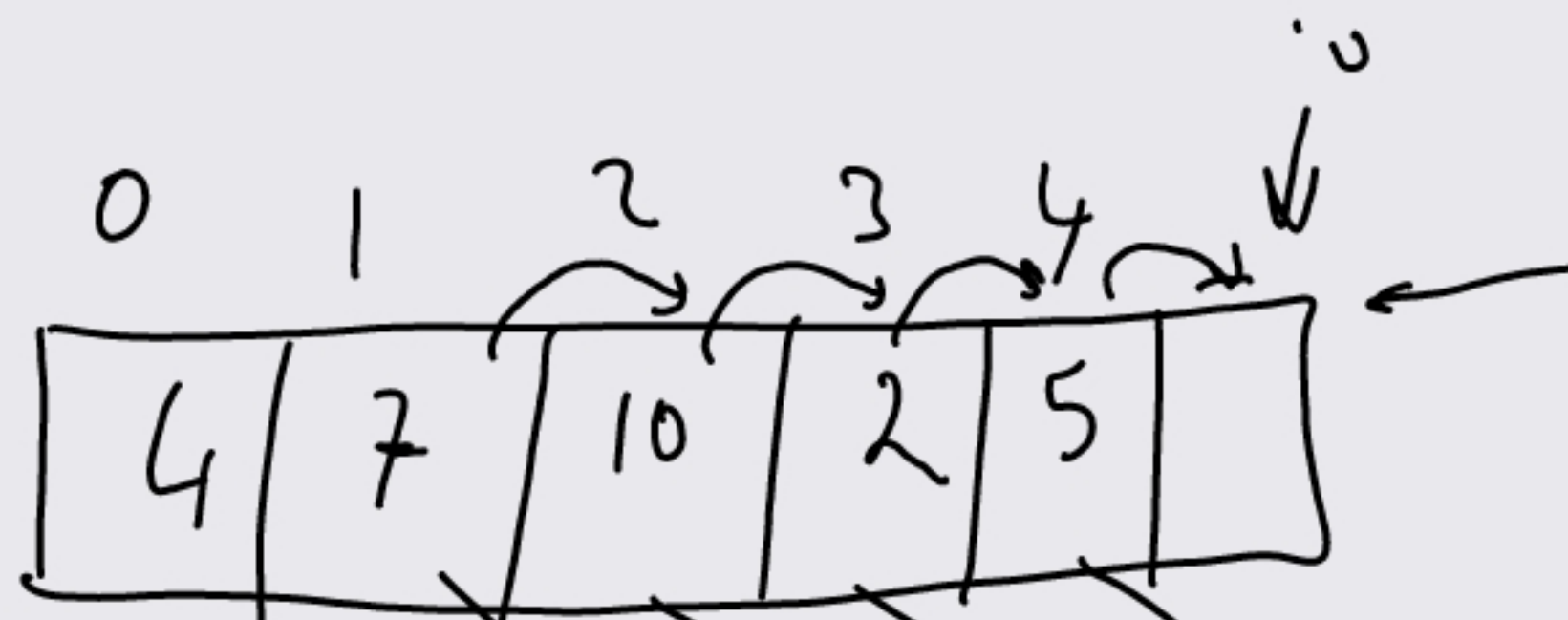
Size = 5



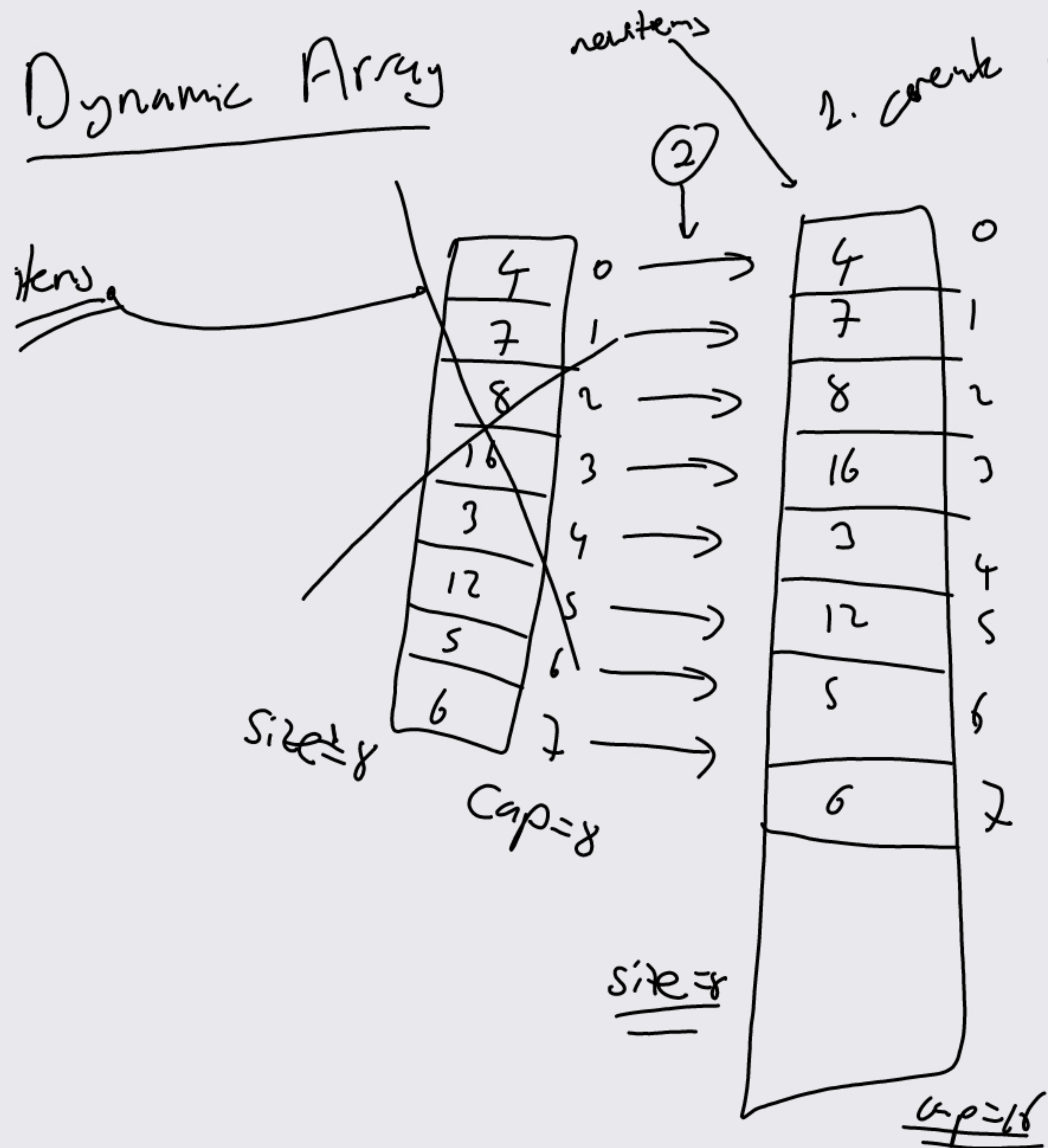
Size = n



Size = n+1



# Dynamic Array



# - Bitwise Copy

typedef struct

} Person;

char \*name;  
int id;

main() {

Person p1;  
char n[20] = "Kadir";

p1.name = n; ✓

p1.id = 2025; ✓

Person p2;

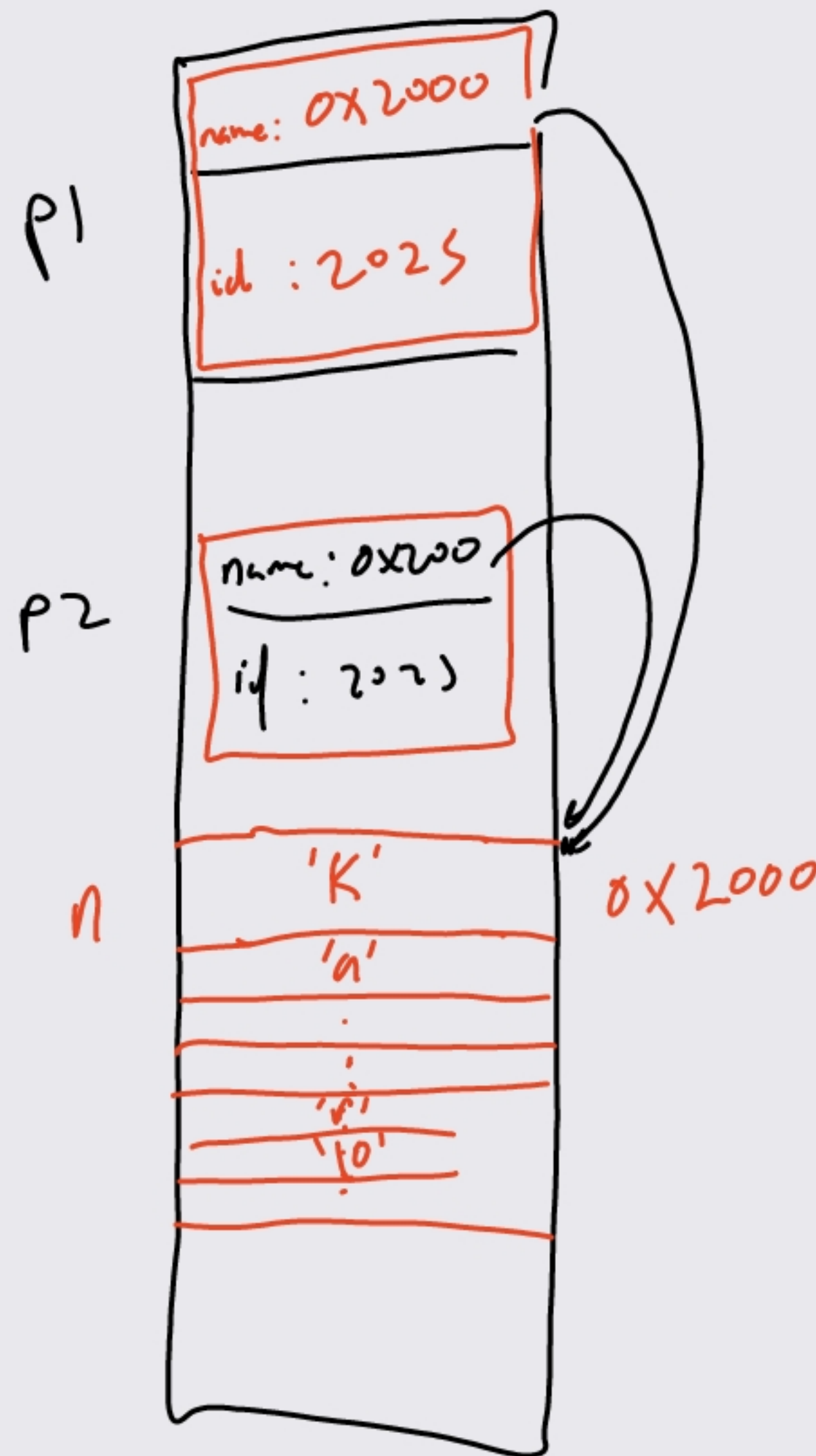
p2 = p1;

strcpy(p1.name, "Kadir");

shallow copy

to resolve  
the issue

use  
Deep Copy



A=B // A. operator (B)

int a, b, c;

a=b=c=18;

a = (b = (c = 10))

int a=8, b;  
printf(" %d", b=a);

Linked List  
Node:

array

{ 4, 7, 10, 1, ..... 28 }

```
for (int i=0; i<14; i++)  
    cout << a[next[i]];
```

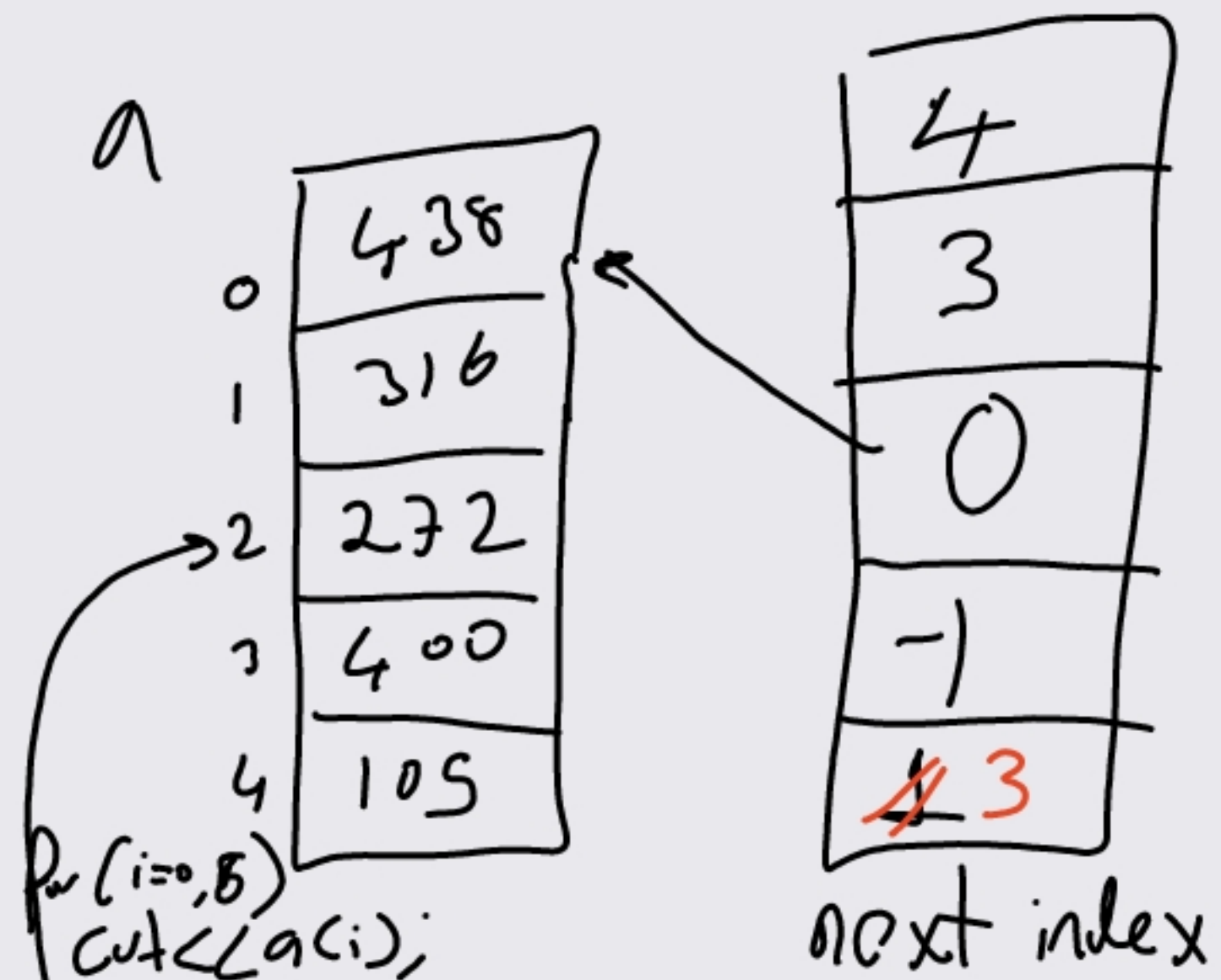
a

|           |     |
|-----------|-----|
| 0         | 4   |
| 1         | 7   |
| 2         | 10  |
| 3         | 1   |
| 4         | 8   |
| 5         | 18  |
| 6         | 22  |
| 7         | 30  |
| 8         | 40  |
| 9         | 14  |
| 10        | 17  |
| 11        | 6   |
| <u>12</u> | -15 |
| 13        | 28  |

|    |
|----|
| 12 |
| 3  |
| 0  |
| 11 |
| 1  |
| 4  |
| 2  |
| 9  |
| 10 |
| 5  |
| 6  |
| 13 |
| 7  |
| 8  |

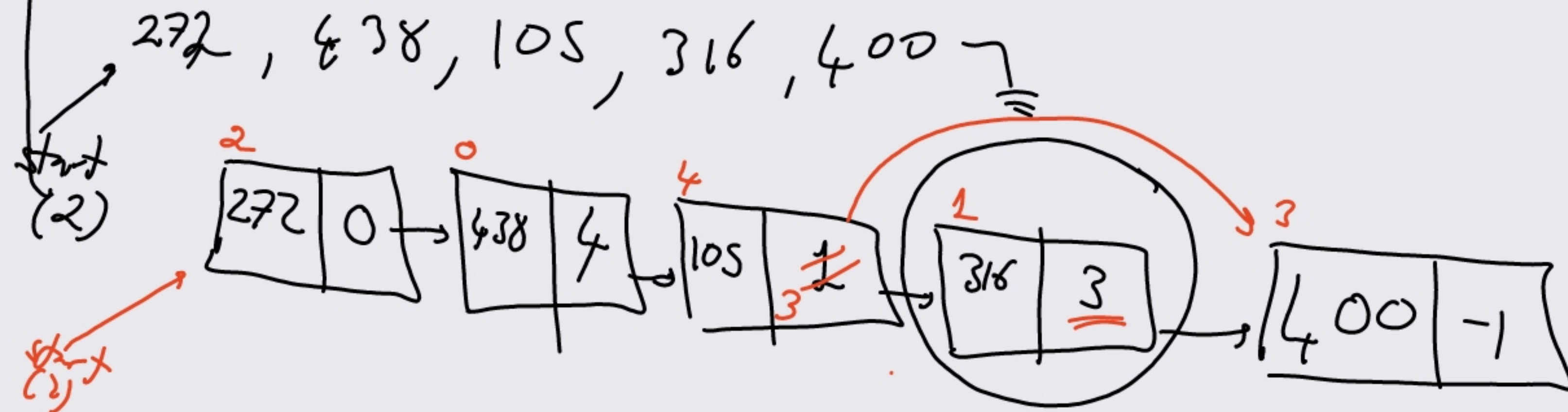
next

index value of smallest value



272, 438, 105, 400 1

438, 316, 272, 400, 105

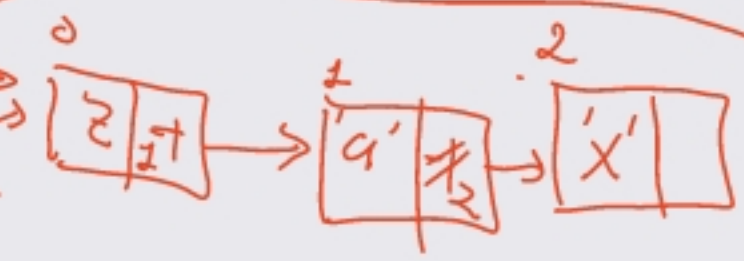
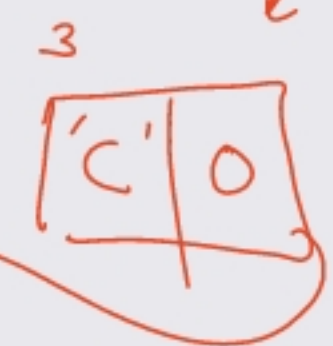


|   |     |              |              |
|---|-----|--------------|--------------|
| 0 | 'z' | <del>F</del> | <del>2</del> |
| 1 | 'a' | <del>F</del> | -1           |
| 2 | 'x' | <del>F</del> | <del>0</del> |
| 3 | 'c' | T            | -1           |
| 4 |     | T            | -1           |
| 5 |     | T            | -1           |
| 6 |     | T            | -1           |
| 7 |     | T            | -1           |
| 8 |     | T            | -1           |

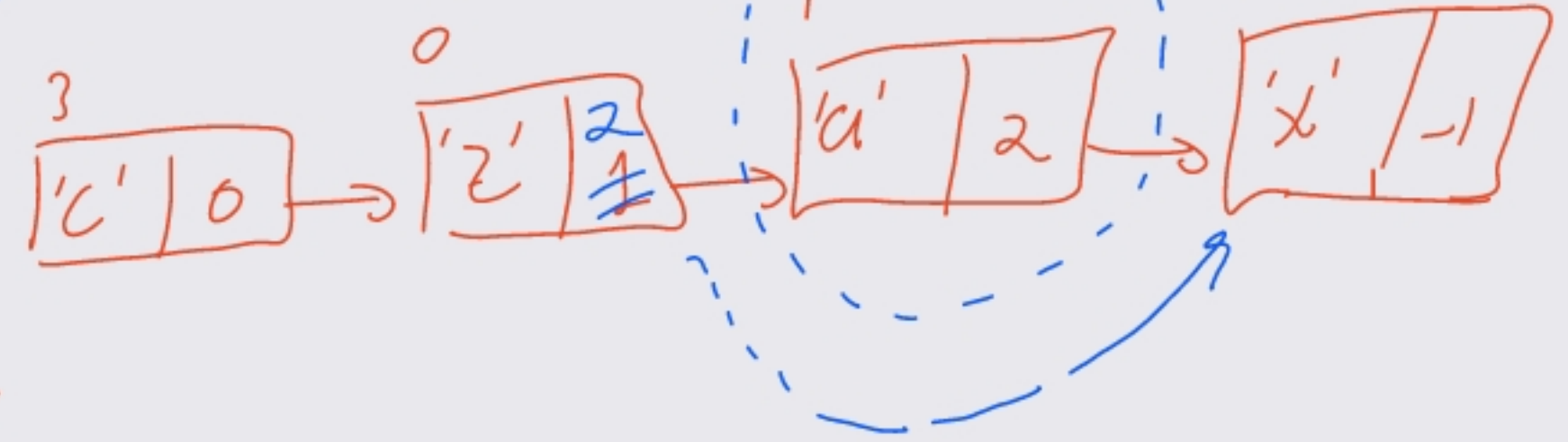
data      avail      next idx

insert('a');  
 insert('x');  
 insert(0, 'c');

start = -1  
 insert('z'); { First avail place is at 0 }  
 start = 0



logical view



start (3)

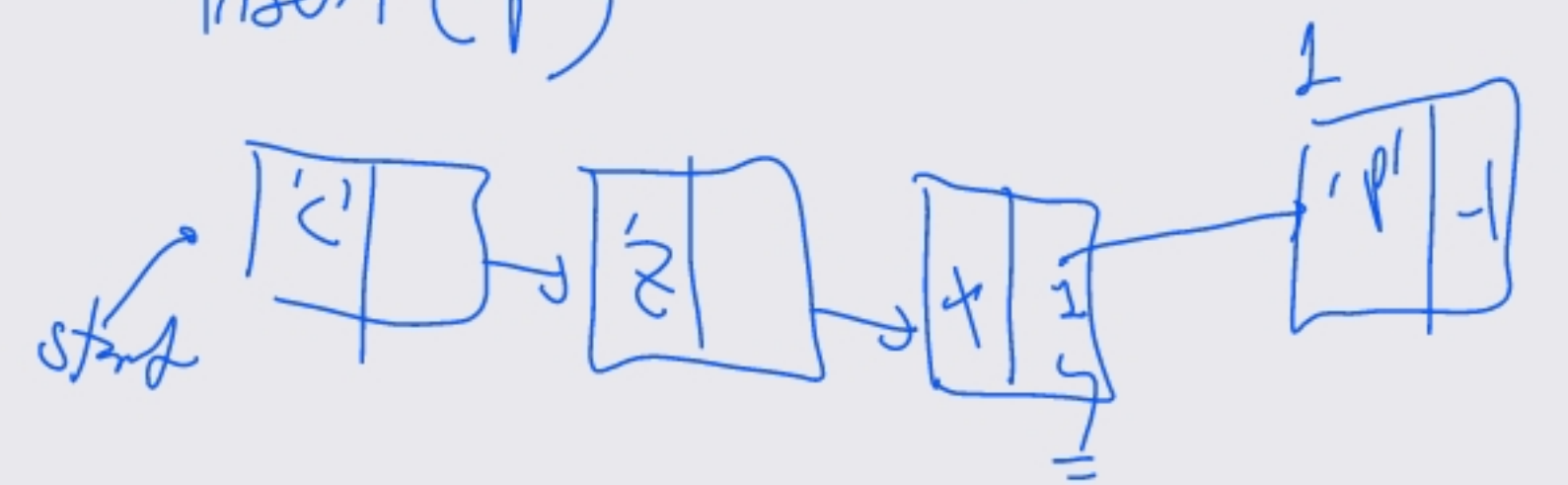
|   |                |              |               |
|---|----------------|--------------|---------------|
| 0 | 'z'            | F            | 2             |
| 1 | <del>'a'</del> | <del>F</del> | <del>-1</del> |
| 2 | 'x'            | F            | 1             |
| 3 | 'c'            | F            | 0             |
|   |                | T            |               |
|   |                | T            |               |
|   |                | ...          |               |
|   |                | ...          |               |

next index

delete('a')

start  
 c, z, x

insert('p')





FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelp

Search (Ctrl+Q)

ArrayBasedLList

Sign in

Live Share

ArrayBasedLList.cpp\*

ArrayBasedLList

append(char ch)

```
ArrayBasedLList() {
    start = -1;
    for (int i = 0; i < MAXSIZE; i++)
        avail[i] = true;
}

void append(char ch) {
    int loc = requestMemory();
    if (loc != -1) {
        data[loc] = ch;
        nextIndex[loc] = -1;
        if (start == -1)
            start = loc;
        else {
```

start

set up walk value to traverse list until we locate the last node (!)

loc

ch -1

Solution Explorer

Search Solution Explorer (Ctrl+Shift+S)

Solution 'ArrayBasedLList' (1 of 1 project)

ArrayBasedLList

References

External Dependencies

Header Files

Resource Files

Source Files

181 %

0 2

Ln: 32 Ch: 17 SPC CRLF

Error List

Entire Solution 0 Errors 2 Warnings 0 Messages Build + IntelliSense

Search Error List

|  | Code   | Description                                                                                           | Project         | File                | Line | Suppression State |
|--|--------|-------------------------------------------------------------------------------------------------------|-----------------|---------------------|------|-------------------|
|  | C26495 | Variable 'ArrayBasedLList::data' is uninitialized. Always initialize a member variable (type:6).      | ArrayBasedLList | ArrayBasedLList.cpp | 18   |                   |
|  | C26495 | Variable 'ArrayBasedLList::nextIndex' is uninitialized. Always initialize a member variable (type:6). | ArrayBasedLList | ArrayBasedLList.cpp | 18   |                   |

Ready

Add to Source Control

ArrayB... How to... Library Calcula... C:\User... C:\User... C:\User... Downl... C:\User... 2406.0... Lecture... ENG TRQ 12:03 08-Jul-25

Visual Studio IDE showing the implementation of a linked list using an array-based approach. The code is in `ArrayBasedLList.cpp`.

```
ArrayBasedLList.cpp*
ArrayBasedLList
append(char ch)

    avail[i] = true;
}

void append(char ch) {
    int loc = requestMemory();
    if (loc != -1) {
        data[loc] = ch;
        nextIndex[loc] = -1;
        if (start == -1)
            start = loc;
        else {
            int walk = start;
            while (nextIndex[walk] != -1)
                walk = nextIndex[walk];
        }
    }
}

int main()
{
    // ...
}
```

Handwritten diagram illustrating the linked list structure:

- A sequence of boxes representing nodes: `[ ]` → `[ ]` → `[ ]`.
- The last node's `nextIndex` points to a new node: `[ ]` → `[ ]`.
- The new node contains the character `ch` and its `nextIndex` is set to `-1`.
- A red arrow points from the text `nextIndex[walk] = loc;` to the new node.

Handwritten note: `nextIndex[walk] = loc;`

Visual Studio interface details:

- Search (Ctrl+Q)
- ArrayBasedLList
- Sign in
- Live Share
- Solution Explorer: ArrayBasedLList (1 of 1 project)
- Properties
- Ln: 35 Ch: 17 SPC CRLF