

# Lecture 5

## Contents

**Definition**

**Applications**

**The difference between the list and the array**

**Design of list**

**Detailed design:**

**Central list**

**List.h**

**List.cpp**

في حاجات بقى الدكتور قال يصححها من المحاضرة اللي فاتت

المحاضرة دي هنبداً موضوع جديد و هو الlists

## Definition

**Lists:** a container data type that contains items of some sort.

## Applications

الapplications بتاعتها كتير جدا infinite

For example:

نظام شئون الطلبة.... List of student in FECU

الرقم القومى..... List of citizen in Egypt

List of drivers in Egypt

List of files on a hard disk

List of programs (processors)

List of friends on FB

List of contacts on phone

وكتييير جدا تانى.

## Operations:

طبيب ايه ال operations اللي نقدر نعملها على ال list

اقدر ازود item في ال list

### 1.insert:

Add item in the list.

اقدر مثلا استرجع او استدعي حاجة من ال list زي مثلا اما بنكتب كام حرف من الاسم اللي عايزين نتصل بيه في ال list contacts.. فيجيب الرقم

### 2.retrieve:

دى في منها نوعين

**\*first:** retrieve the first item from the list.

يعنى نستدعي اول item في ال list

**\*next:** retrieve the next item from the list.

هنفهمها بالضبط كمان شوية

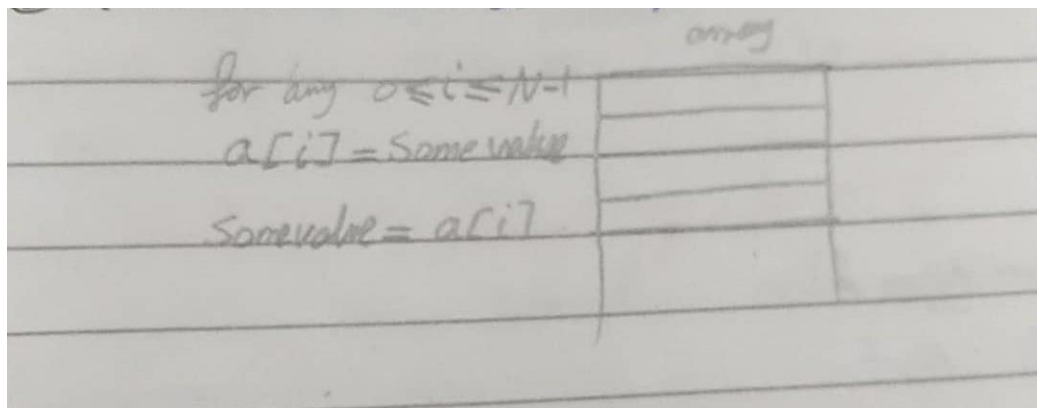
.... المهم ممكن كمان زي ما عملنا insert نعمل delete

### 3.delete:

Remove items from the list.

## The difference between the list and the array

سؤال قبل ما نكمل ... ايه الفرق بين ال list وال array؟



ال array بيبقى بيت access بطريقة ال loop دى .. غير ال list اللي نقدر نعملها ب insert و delete وكده ..  
ال array احسن استخدام ليه .. بيبقى في ال matrix او ال vector.

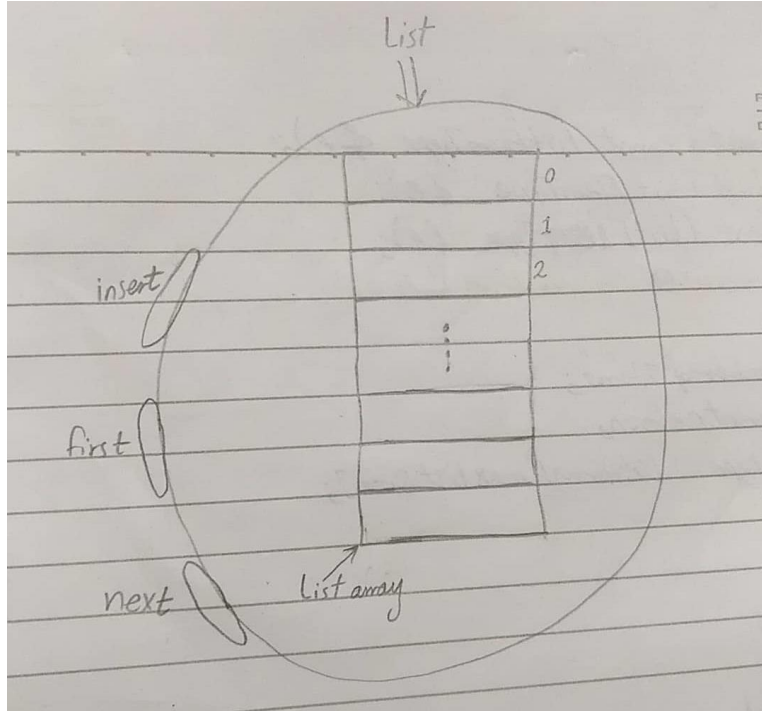
ولو عايزين non numerical applications ييقى نستخدم ال lists.

## Design of list

نكمل بقى موضوع ال list عايزين نعمل design لل list عشان نبدأ نكتب الكود.

فهنعمل list باستخدام ال array وهنحطه في ال black box بتاعنا ونحط ال interfaces اللي هما insert ... first و... next.

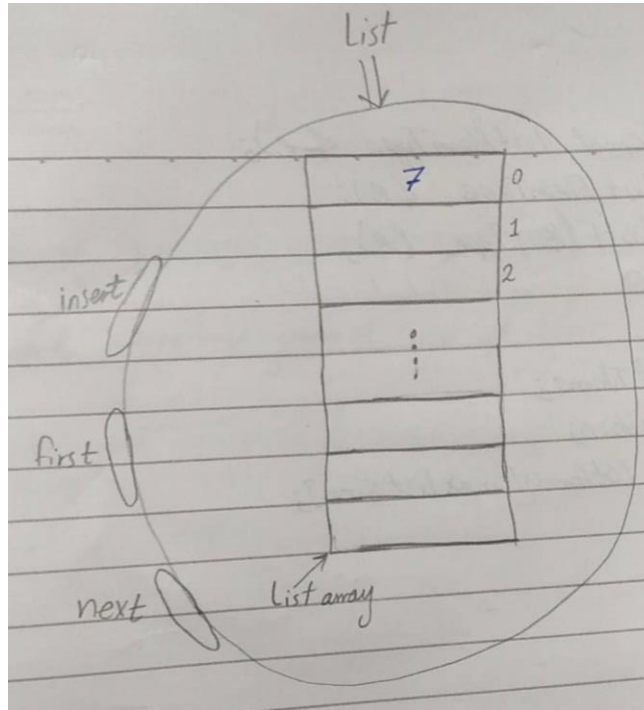
فين delete ؟ .. delete الدكتور قال اعملوها في البيت.



فاحنا في الكود ممكن نكتب

Insert(7)

يعنى هنحط 7 في المكان رقم 0



نكتب مثلا

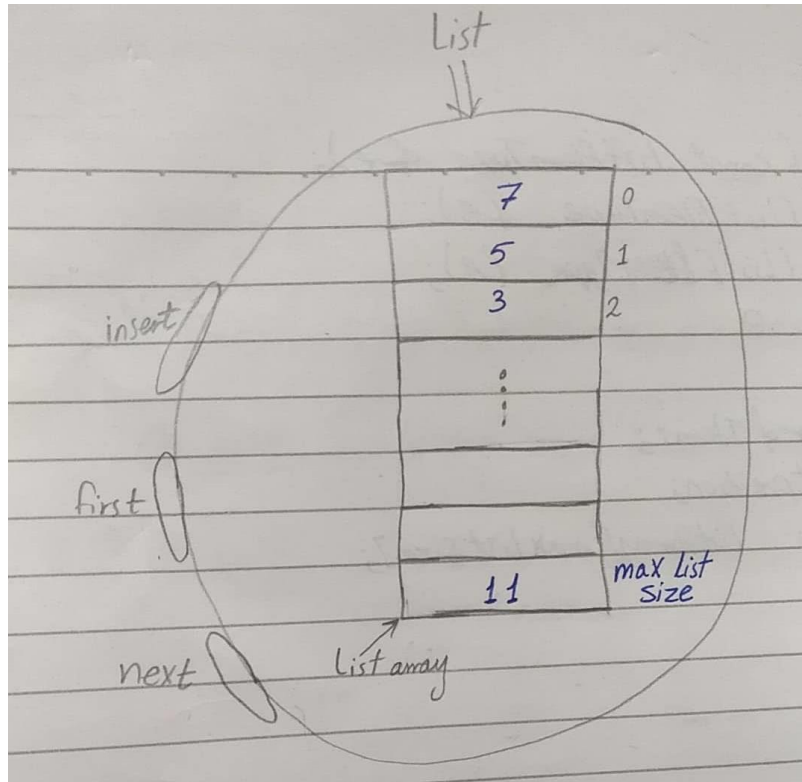
Insert(5)

Insert(3)

Insert(11)

وهكذا لحد ما نكتب في اخر خانه

الى هنسميها max list size



لو كتبنا كمان

Insert(18)

failت

إذا احنا محتاجين counter هنسميه number of items

1. يبقى الinsert محتاجين فيه number of items ومع اول insert .. الnumber of items هيبقى بواحد ومع كل insert هيزيد واحد.

2. first مش هحتاج حاجة بس ان الnumber of items يبقى على الأقل ب1

3. next: مين الnext دى؟؟ .. الnext ده المفروض يبقى في مؤشر واقف في حنة عشان الnext يجيب اللى بعده .. يبقى محتاجين حاجة اسمها الcurrent position.

ولو عملت first .. هنبدأ الcurrent position ويفضل ينزل مع كل next.

نعمل detailed design زى اللى عملناهم تانى محاضرة.

### Detailed design:

Bool insert (ListElemType &e)

Insert item e into list.

- Preconditions:

لازم ال list يكون فيها مكان فاضى .. إذاً

List is not full  $\text{number of items} < \text{max list size}$  يعنى

يعنى لو ال max list size ب50 .. لازم ال number of items على الأقل ب49 عشان ده معناه ان في خانة واحدة فاضية .. اللى هي اللى هعمل فيها insert دلوقتى و خد بالك ان ال max list size ده رقم ثابت .. وعدد كل الخانات الفاضية والمليانة .. لكن ال number of items هو عدد الخانات المليانة يعنى لو بس اول خانة ميانة هيبقى بواحد وهكذا و ده بيزيد مع كل insert.

- Post conditions:

-number of items is incremented.

-item e is inserted in the list.

- Returns:

-True if insert succeeds.

-False otherwise.

الدكتور قال خدوا بالكم ان ال current position ملوش اى علاقة بال insert .. عشان الناس بتتلخبط في الحته دى.

واحد سأل ليه ال insert مترجعش ال e بدل ال bool الدكتور قال ممكن بس انا اللى بعمل ال design فاننا مختاره كده ... والاحسن انك تعملها وكده وتبقى تعمل function تانية بترجع ال e يبقى اسمها gets e مثلاً.

Bool first (listElemType &e)

Retrieve the first item from the list.

- Preconditions:

-list is not empty  $\text{number of items} > 0$  يعنى

عشان يلاقى ال first ده عشان يقرأه.

- Postconditions:

-current position is set to zero.

عشان هنبدأ نقرأ بقى ونقول next فيفضل ال current position يزيد

-e will get the value stored in list array[0].

هم بيتعملوا كده بالترتيب

يعنى مينفعش ال current position يتغير الأول لان كده ال e هتاخذ قيمة المكان اللى بعد اللى انا اقصد.

- Returns:

-true if first succeeds.

-false otherwise.

### Bool next (listElem type &e)

Retrieve item stored in next position in the list.

- Preconditions:

- first has been executed ..  $\text{current position} \geq 0$  يعنى

- يعنى لازم ابقى ندهت على ال first الأول .. مينفعش اعمل next قبل ال first عشان كمان ال current position.

- there is "next" item in the list ( $\text{current position} \leq \text{number of items} - 2$ )

بمعنى انى لو مثلا عملت insert خمس مرات .. فانا كده عندى خمس خانات مليونين .. يعنى ال number of items=5 .. عملت first فى ال current position=0 قعد اعمل next بس اخر next هعرف اعملها اللى هو عند الخانة اللى قبل الأخيرة .. اللى عندها ال current position=3

- Post condition:

- current position is incremented.

- e gets the value stored in list array [current position].

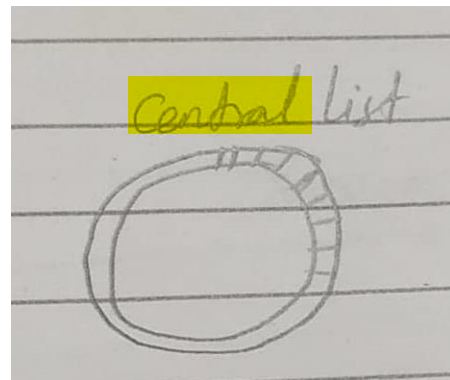
- Returns:

- true if next succeeds.

- false otherwise.

### Central list

ال list اللى خدناها دى اسمها ال linear list .. فى حاجة تانية اسمها ال central list



ده شكلها اللى هو هنعمل first وبعدين نعمل next... next لحد اخر next مش هتدى fail وللا حاجة ... هتروح ال first تانى و تكمل عادى.

الدكتور بيقول زيها زى مستر اكس كده

Central list has no end :”D

سؤال صغير بقى ... هو الكلام اللي بنقعد نكتبه و ال design ده بيتحط فين؟ ... المفروض بيتحط في حاجة اسمها programmer reference manual اللي في ال user manual بنعمله في البرامج الكبيرة اوى ... بس في حالتنا دى ممكن نكتبه في comments في الكود  
نبدأ نكتب الكود بقى

### List.h

هنعرف فيها ال variables وال class وكده

```
Const int MaxListSize=1000;
```

هنكتب حاجة اسمها typedef عشان نعرف data type جديدة

```
Typedef <.....> listElemtype;
```

مكان النقط دى بنحط ... int مثلا او string على حسب الحاجة اللي هعرفها ب listElemtype ... عايزها فيها int وللا string وللا ايه.

نكمل

```
Class List {
```

```
Public:
```

```
List(); // this is the constructor(of the last lecture)
```

```
Bool insert ( const ListElemType &e);
```

```
Bool first (ListElemType &e);
```

```
Bool next (ListElemType &e);
```

فاحنا عرفنا هنا ال functions عادى اللي قعدنا علناها design من شوية .. بس زيادة عليهم ال constructor  
وده هنا مهم جدا جدا جدا (الدكتور هو اللي أكد كده)

```
Private:
```

```
ListElemType ListArray[MaxListSize];
```

```
Int NumberOfItems;
```

```
Int CurrentPosition;
```

```
};
```

ال file التانى بقى...



## List.cpp

```
List::List()
```

```
{NumberOfItems=0; //initially the list is empty
```

```
CurrentPosition=-1; //nowhere
```

دى ال initial value بتاعت ال current position عشان هو كده كده زى ما قلنا في ال design ال first  
هتت initialize ب zero ... وعشان حطينا فيها -1 عرفناها ب int كان ممكن نعرفها ب unsigned int لكن  
حطينا فيها -ve فلازم يبقى int

```
};
```

```
Bool list::insert(const ListElemType &e)
```

```
{ if (NumberOfItems>=MaxListSize)
```

```
Return false;
```

```
Else
```

```
{ListArray[NumberOfItems]=e;
```

```
NumberOfItems++;
```

```
}
```

```
Return true;
```

```
}
```

هنكمل الكود المرة الجاية .. بس سؤال

ايه حوار ال & وال const اللي بنبعث بيها ال e في ال functions؟؟

الدكتور قال انت لو خليت ال e تنبعث by value ... اللي هيحصل هو copy constructor او bit wise  
copy وده هياخد cycles كتيرة و شغل كتير في ال memory فينبعت by reference احسن و نط قبلها  
. const

في حاجات بقى الدكتور قال يصححها من المحاضرة اللي فاتت

وهى اننا لو كتبنا complex(10,5) .. كأنهم int كده .. ال compiler هيعمل explicit casting ل double  
ويعديها عادى.

تانى حاجة .. قال ال copy constructor لازم نبعت ال variable by reference

كده ... complex(const complex &c);