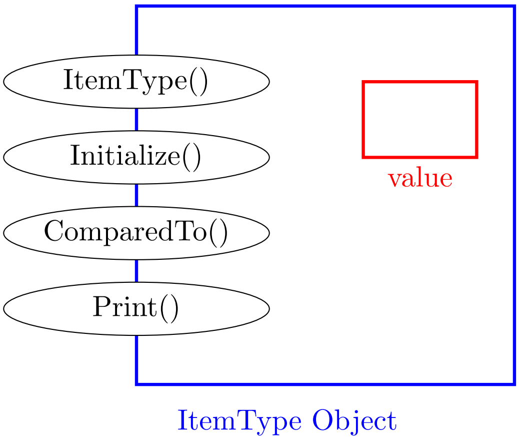
List Usage Example

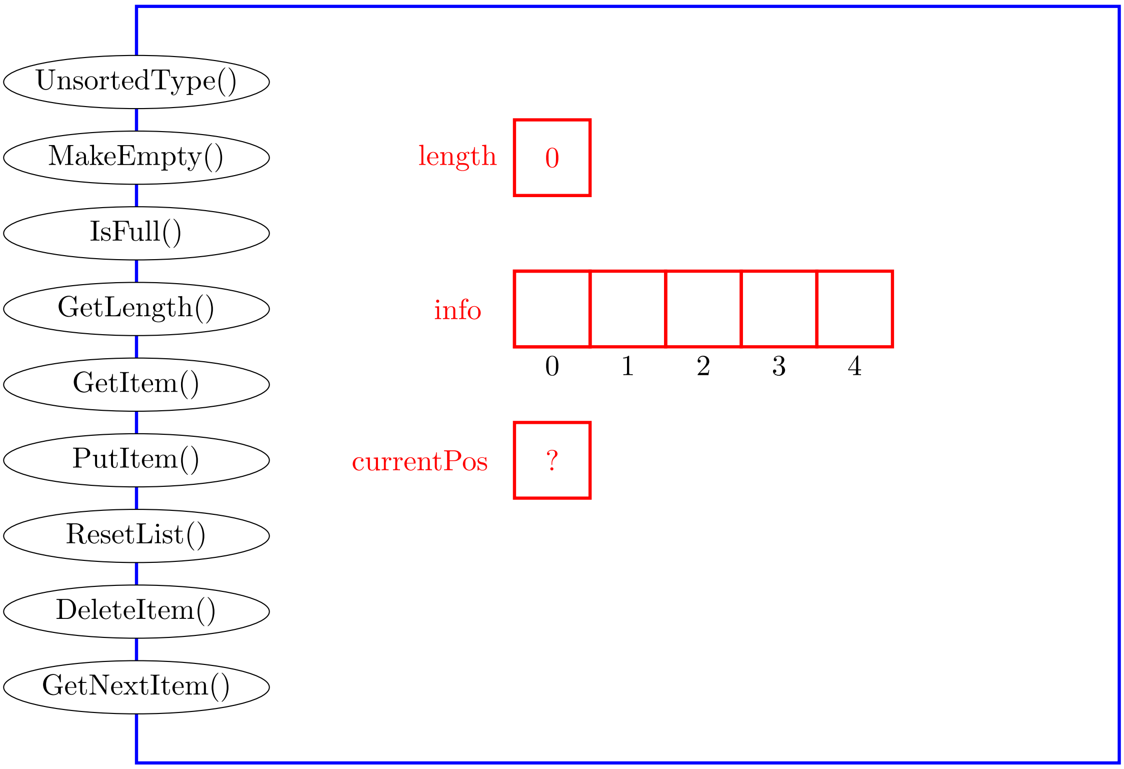
CSE 225 - Data Structures and Algorithms

Md. Mahfuzur Rahman
ECE Department
North South University

1 ItemType


ItemType()
Initialize()
ComparedTo()
Print()
value
ItemType Object

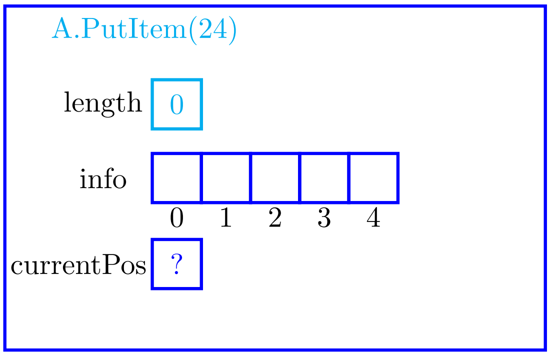
2 Array Based Unsorted List

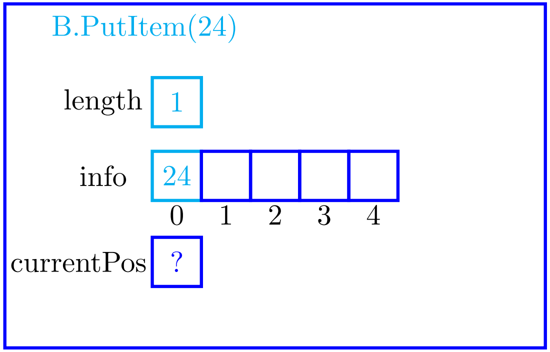


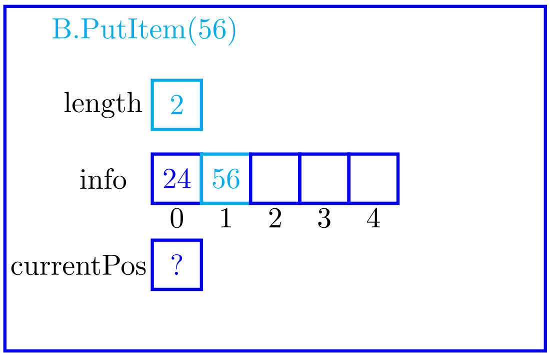
Array Based Unsorted List Object

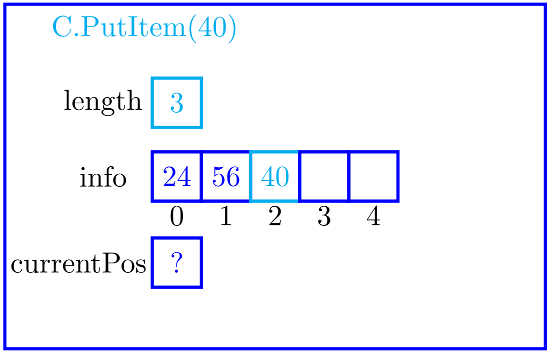
2.1 Use of PutItem()

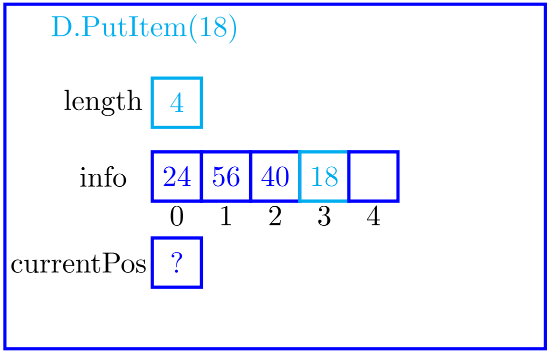
```
 //Use of PutItem()
#include "unsorted.h"
using namespace std;
int main()
{
  UnsortedType list;

  ItemType item1;
  item1.Initialize(24);
  list.PutItem(item1);

  ItemType item2;
  item2.Initialize(56);
  list.PutItem(item2);

  ItemType item3;
  item3.Initialize(40);
  list.PutItem(item3);

  ItemType item4;
  item3.Initialize(18);
  list.PutItem(item4);
  return 0;
}
```


A.PutItem(24)


B.PutItem(24)


B.PutItem(56)

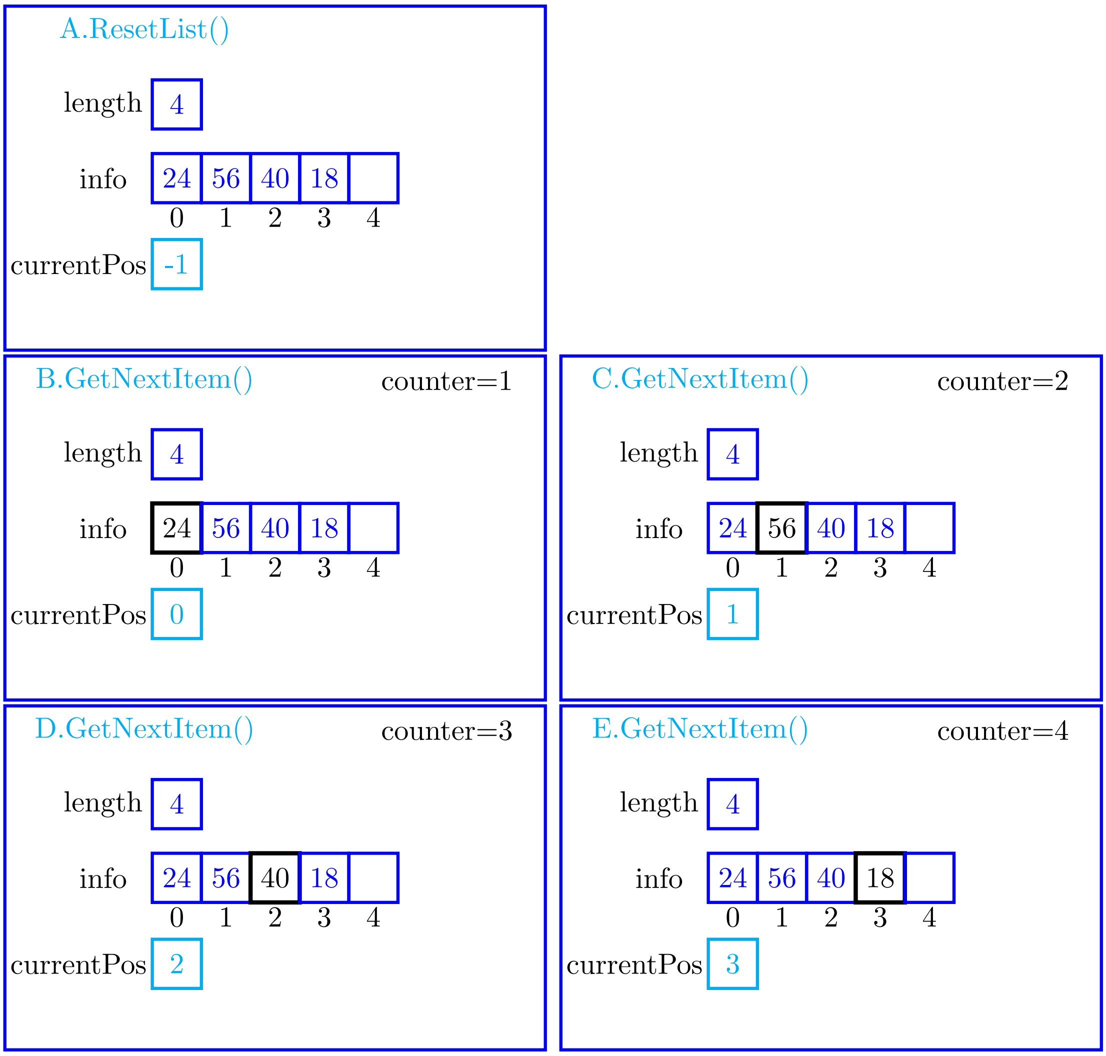

C.PutItem(40)


D.PutItem(18)

2.2 Use of ResetList() and GetNextItem()

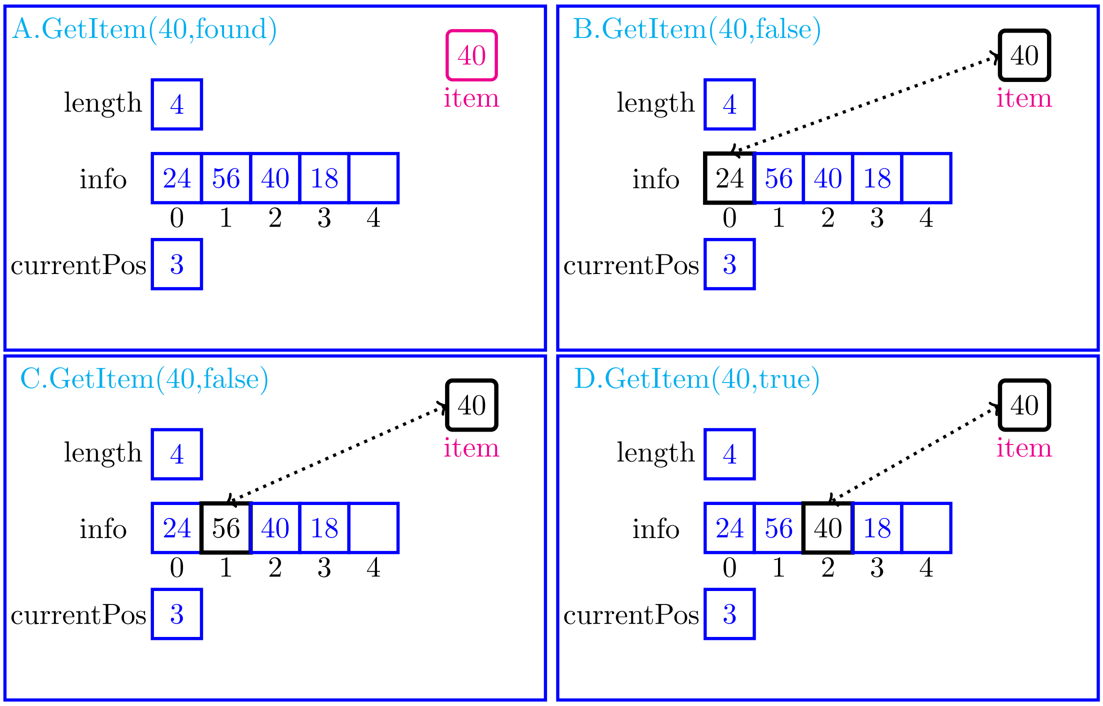
```
//Use of ResetList() and GetNextItem()
#include <iostream>
```

```
#include <fstream>
#include <string>
#include <cctype>
#include <cstring>

#include "unsorted.h"

using namespace std;
void PrintList(ofstream& outFile, UnsortedType& list);

int main()
{

  UnsortedType list;

 .............         // steps to insert items into list

  ofstream outFile;       // file containing output
  string outFileName;     // output file external name

  cout << "Enter name of output file; press return." << endl;
  cin  >> outFileName;
  outFile.open(outFileName.c_str());
  PrintList(outFile, list);

 }


void PrintList(ofstream& dataFile, UnsortedType& list)
{
  int length;
  ItemType item;
  dataFile << "PrintList" << endl;
  list.ResetList();
  length = list.GetLength();
  if (length == 0)
    dataFile << "List is empty.";
  else
    for (int counter = 1; counter <= length; counter++)
    {
      item = list.GetNextItem();
      item.Print(dataFile);
    }
  dataFile << endl;
}
```



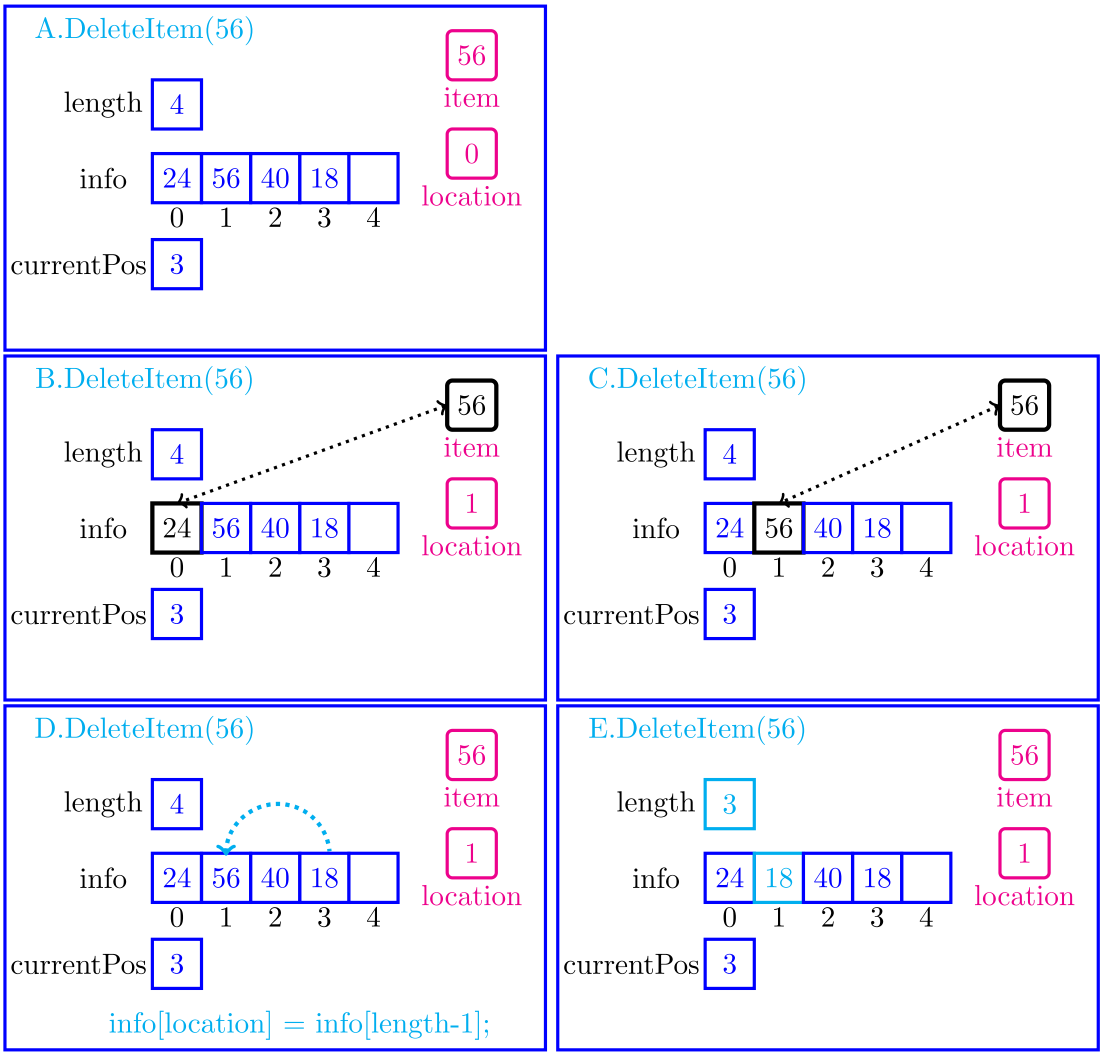
2.3 Use of GetItem()

```
//Use of GetItem()
#include <iostream>

#include "unsorted.h"

using namespace std;

int main()
{
  UnsortedType list;
    ..............        // steps to insert items into list

  ItemType item;
  bool found;
  int number = 40;
  item.Initialize(number);
  item = list.GetItem(item, found);
  if (found)
    cout << number << " found in list." << endl;
  else cout << number  << " not in list."  << endl;

 }
```



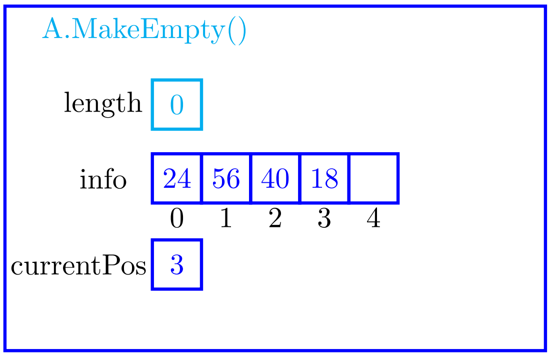
2.4 Use of DeleteItem()

```
//Use of DeleteItem()
#include <iostream>
#include "unsorted.h"

using namespace std;

int main()
{
  UnsortedType list;
    ..............          // steps to insert items into list

  ItemType item;
  int number = 56;
  item.Initialize(number);
  list.DeleteItem(item);

 }
```

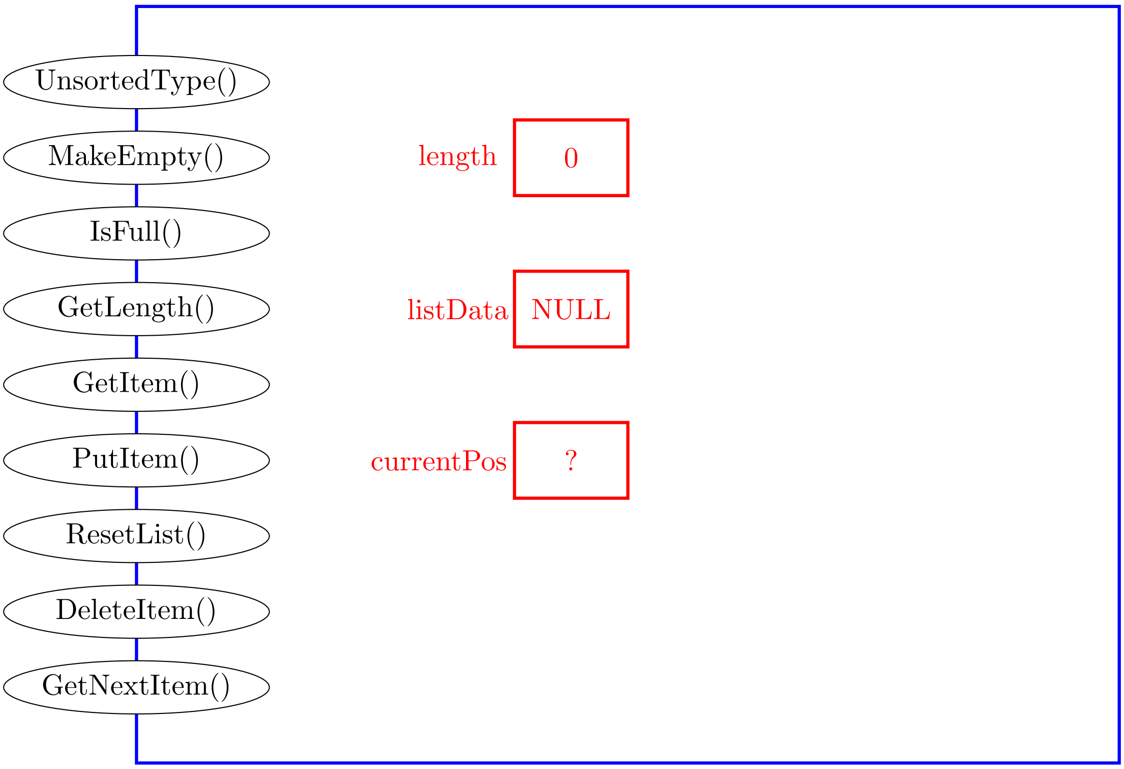


2.5 Use of MakeEmpty()

```
//Use of MakeEmpty()
#include <iostream>
#include ”unsorted.h”

using namespace std;

int main()
{
  UnsortedType list;
    ..............        // steps to insert items into list

  list.MakeEmpty();

 }
```


A.MakeEmpty()
length
0
info
24
56
40
18
0
1
2
3
4
currentPos
3

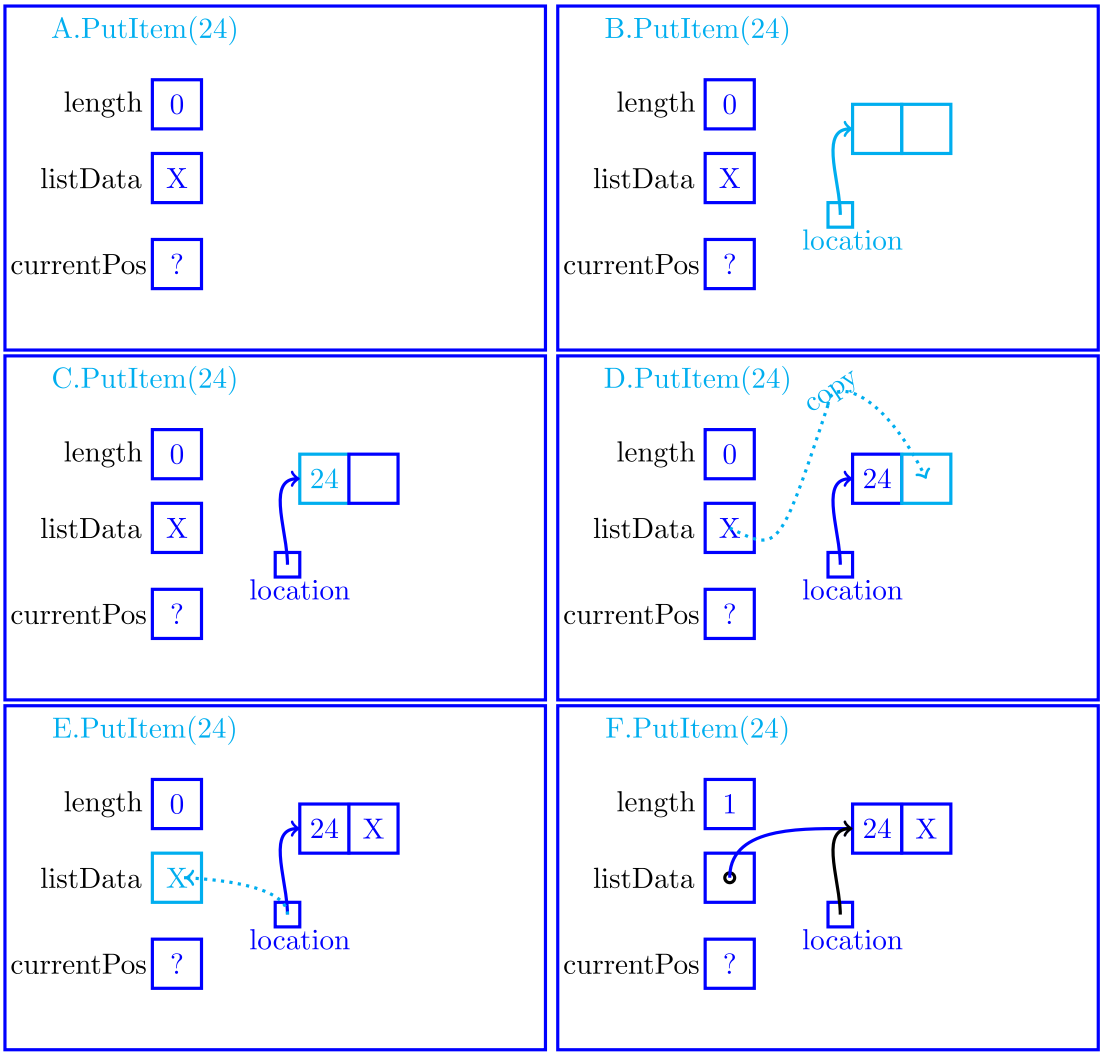
3 Linked-List Based Unsorted List

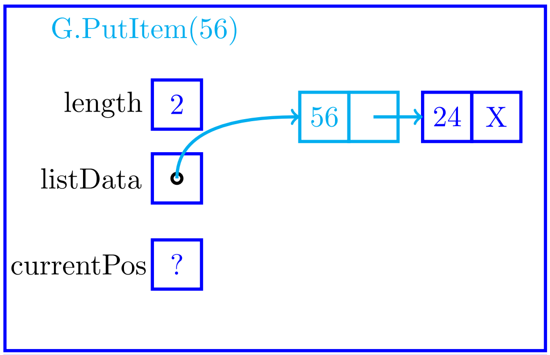


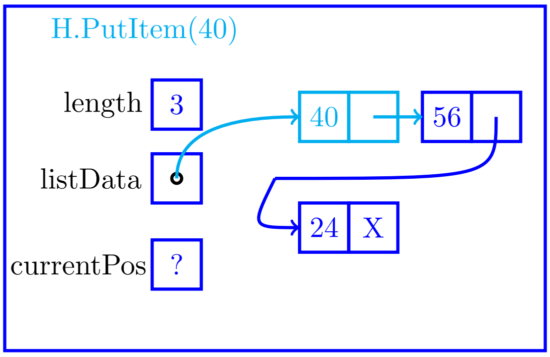
Linked-List Based Unsorted List Object

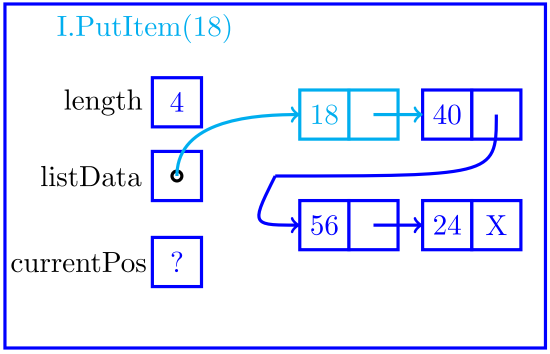
3.1 Use of PutItem()

```
 //Use of PutItem()
#include "unsorted.h"
using namespace std;
int main()
{
  UnsortedType list;

  ItemType item1;
  item1.Initialize(24);
  list.PutItem(item1);

  ItemType item2;
  item2.Initialize(56);
  list.PutItem(item2);

  ItemType item3;
  item3.Initialize(40);
  list.PutItem(item3);

  ItemType item4;
  item3.Initialize(18);
  list.PutItem(item4);
  return 0;
}
```



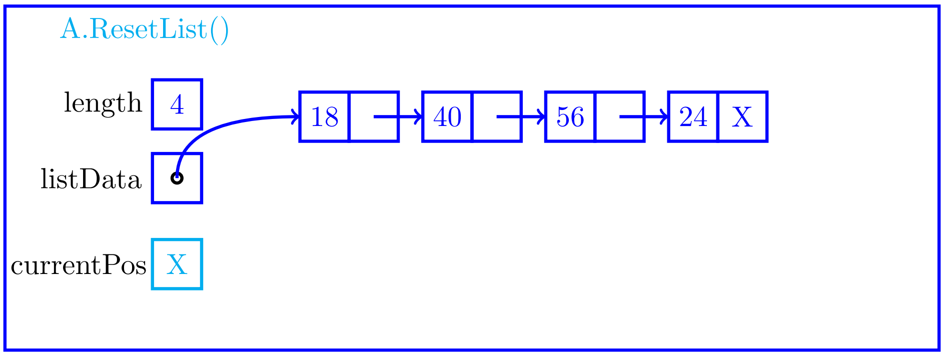


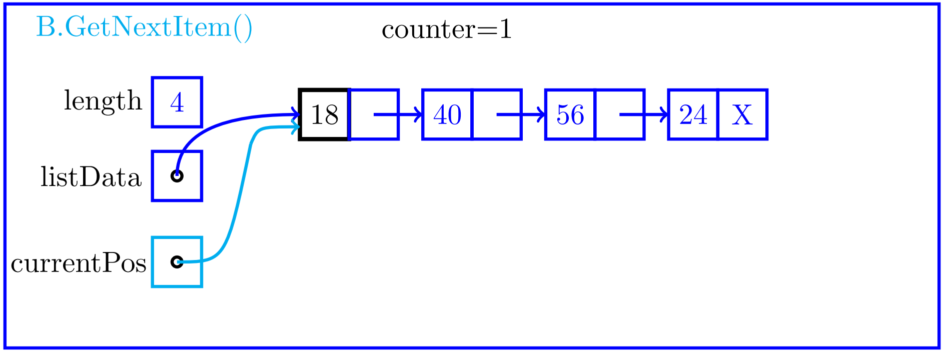


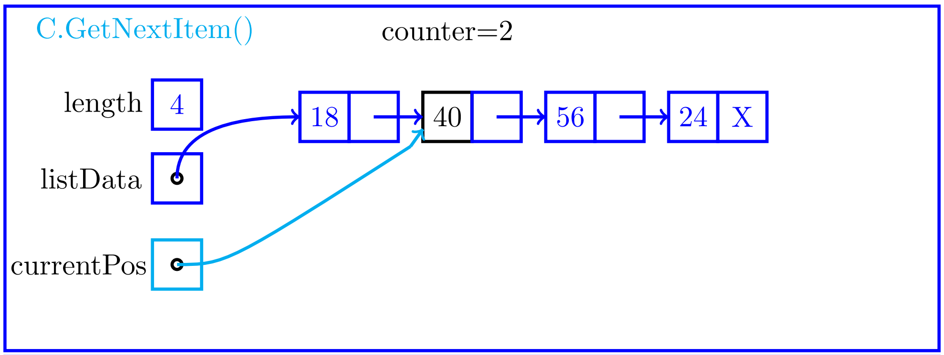


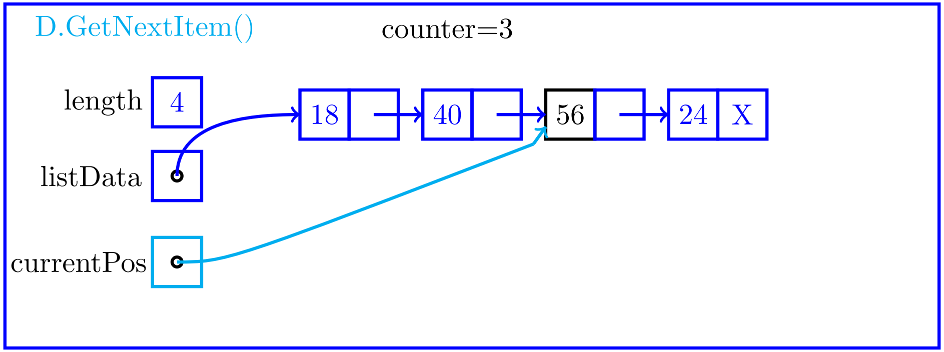
3.2 Use of ResetList() and GetNextItem()

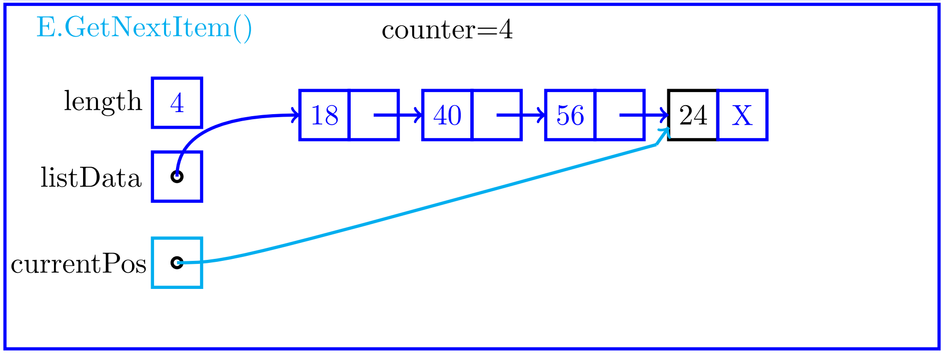
```
//Use of ResetList() and GetNextItem()
#include <iostream>
#include <fstream>
#include <string>
#include <cctype>
#include <cstring>

#include "unsorted.h"

using namespace std;
void PrintList(ofstream& outFile, UnsortedType& list);

int main()
{

  UnsortedType list;

 .............          // steps to insert items into list

  ofstream outFile;       // file containing output
  string outFileName;     // output file external name

  cout << "Enter name of output file; press return." << endl;
  cin  >> outFileName;
  outFile.open(outFileName.c_str());
  PrintList(outFile, list);

 }


void PrintList(ofstream& dataFile, UnsortedType& list)
{
  int length;
  ItemType item;
  dataFile << "PrintList" << endl;
  list.ResetList();
  length = list.GetLength();
  if (length == 0)
    dataFile << "List is empty.";
  else
    for (int counter = 1; counter <= length; counter++)
    {
      item = list.GetNextItem();
      item.Print(dataFile);
    }
  dataFile << endl;
}
```





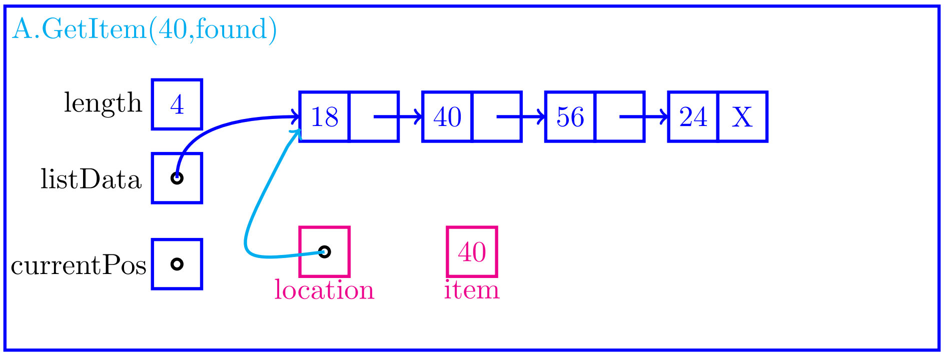


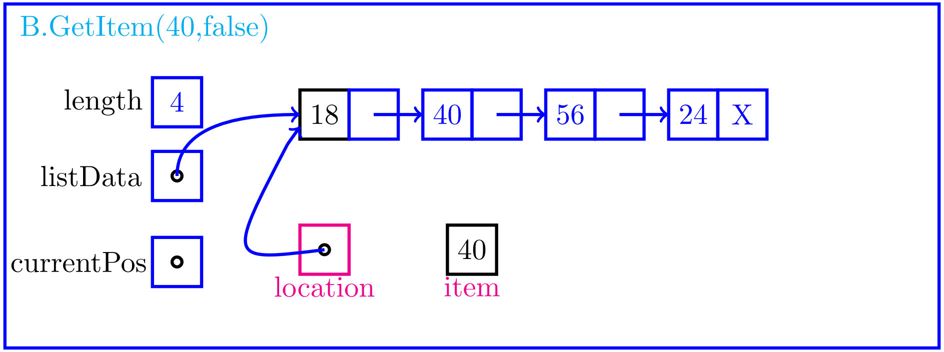


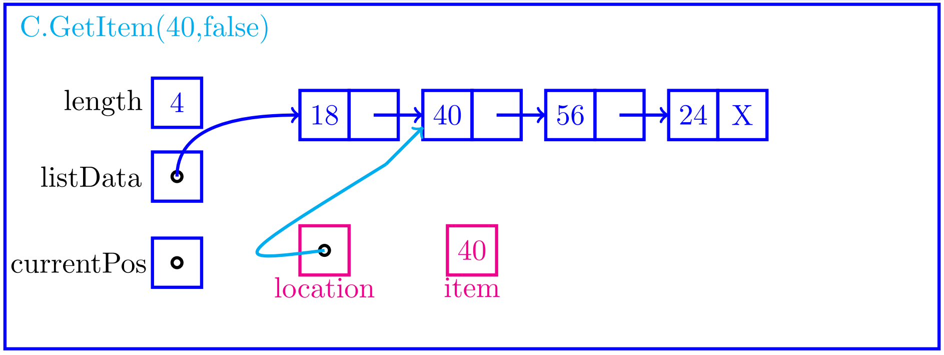


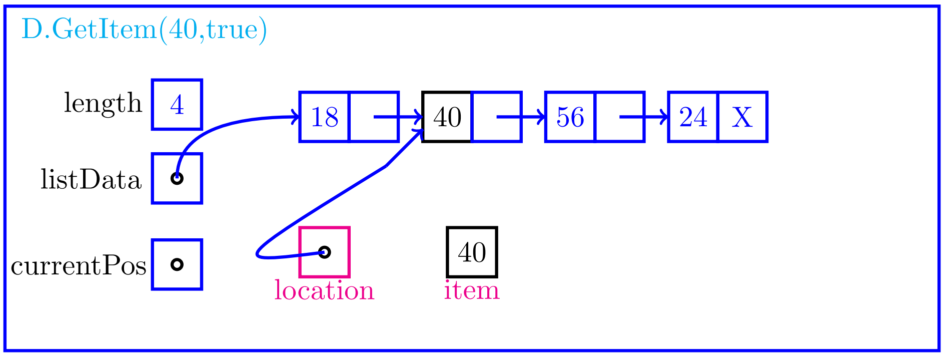
3.3 Use of GetItem()

```
//Use of GetItem()
#include <iostream>

#include "unsorted.h"

using namespace std;

int main()
{
  UnsortedType list;
    ..............        // steps to insert items into list

  ItemType item;
  bool found;
  int number = 40;
  item.Initialize(number);
  item = list.GetItem(item, found);
  if (found)
    cout << number << " found in list." << endl;
  else cout << number  << " not in list."  << endl;

}
```



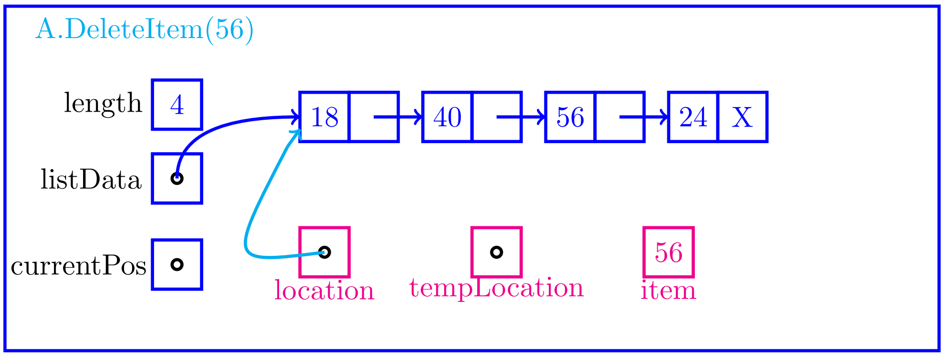


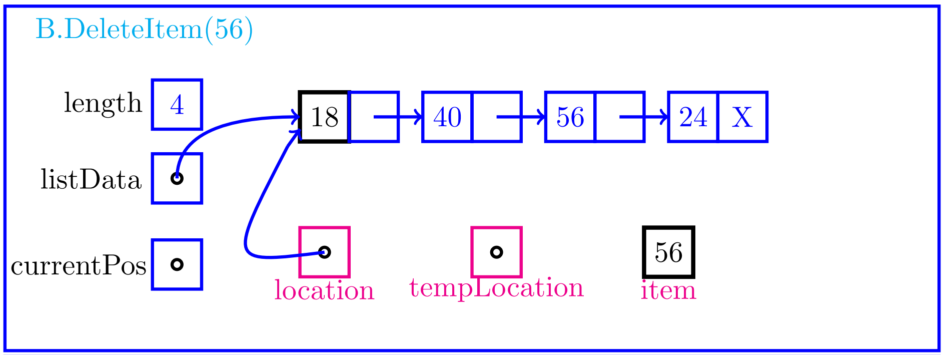


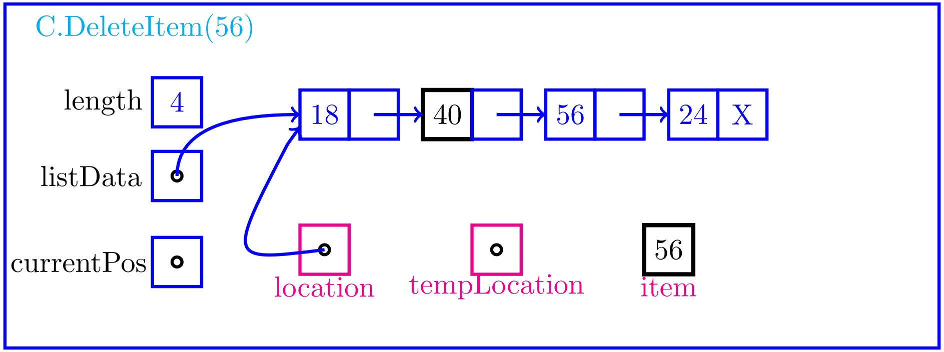


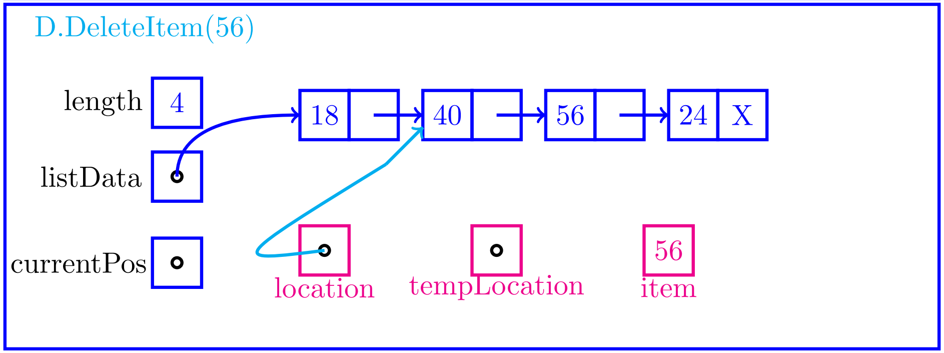
3.4 Use of DeleteItem()

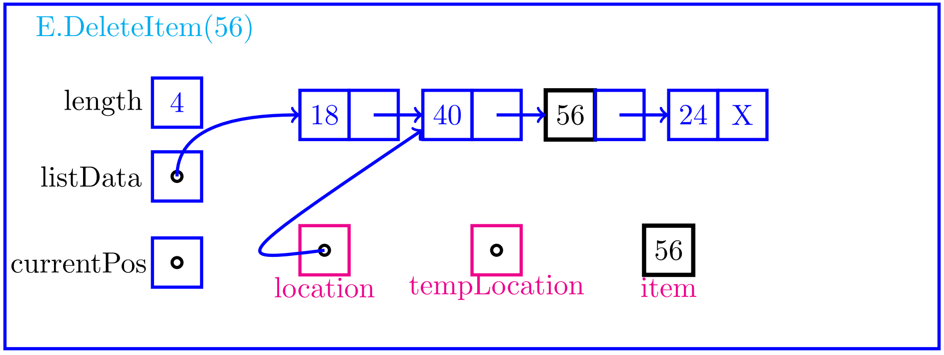
```
//Use of DeleteItem()
#include <iostream>
#include "unsorted.h"

using namespace std;

int main()
{
  UnsortedType list;
     ..............        // steps to insert items into list

  ItemType item;
  int number = 56;
  item.Initialize(number);
  list.DeleteItem(item);

 }
```

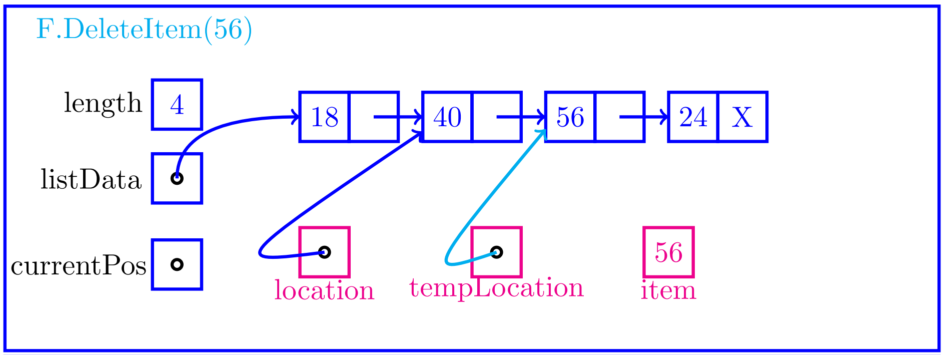


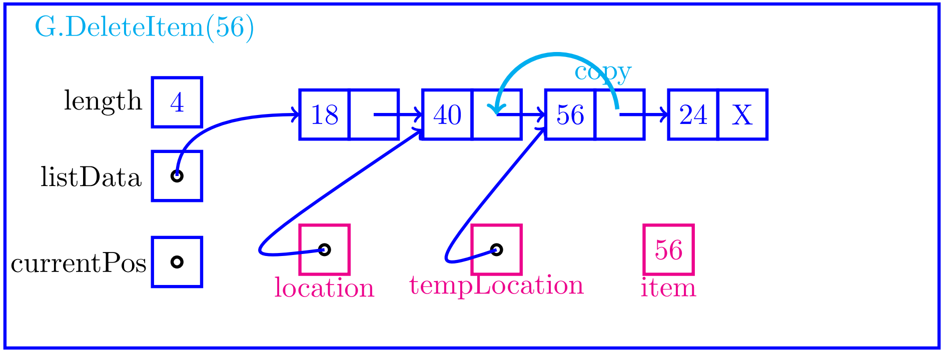

B.DeleteItem(56)
length
4
listData
currentPos
18
40
56
24
X
location
tempLocation
56
item

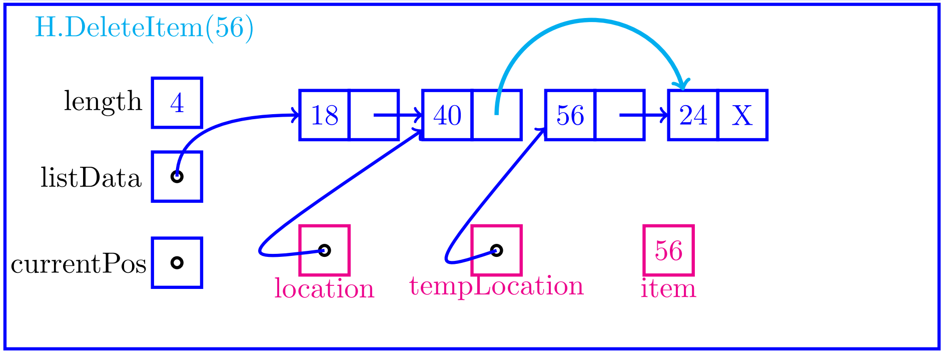

C.DeleteItem(56)
length
4
listData
currentPos
18
40
56
24
X
location
tempLocation
56
item

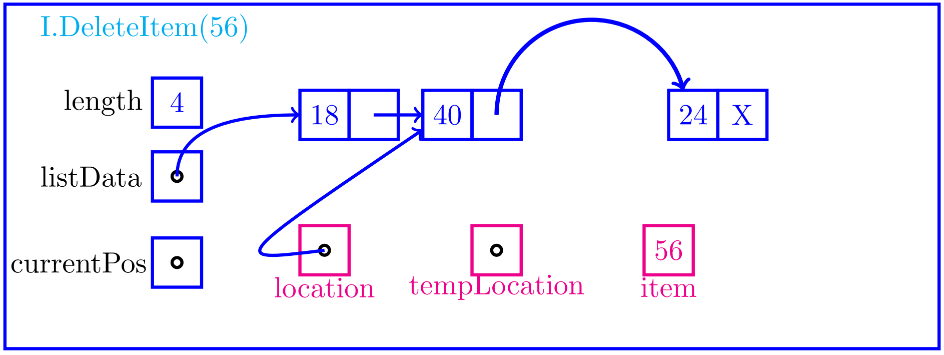

D.DeleteItem(56)
length
4
listData
currentPos
18
40
56
24
X
location
tempLocation
56
item


E.DeleteItem(56)
length
4
listData
currentPos
18
40
56
24
X
location
tempLocation
56
item



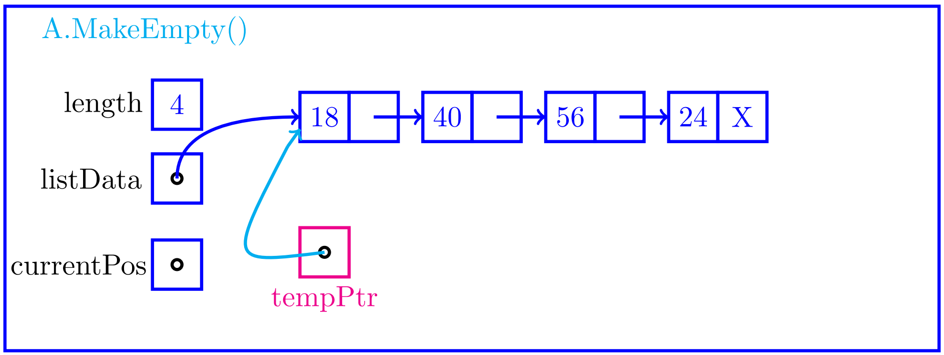


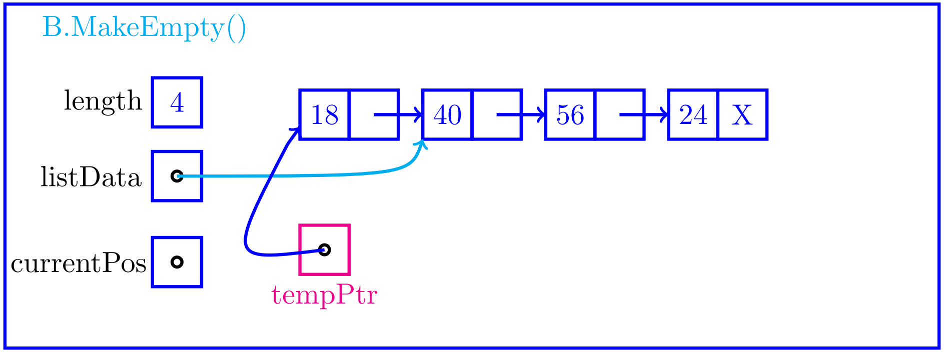


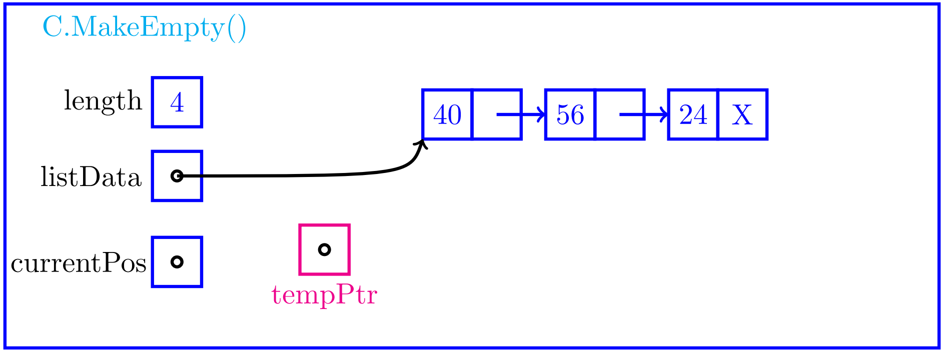


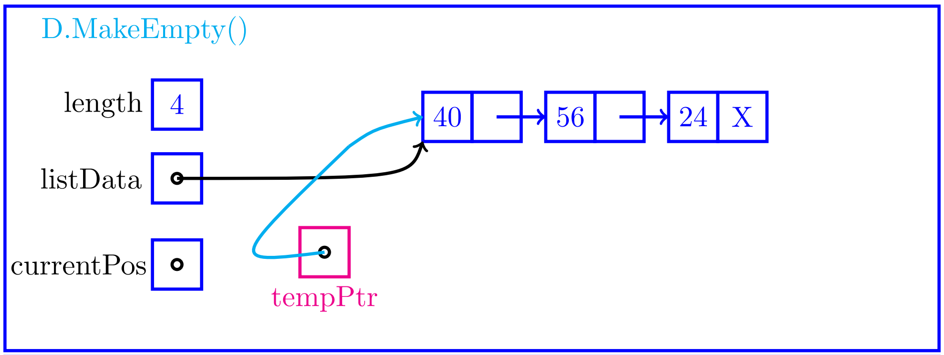
3.5 Use of MakeEmpty()

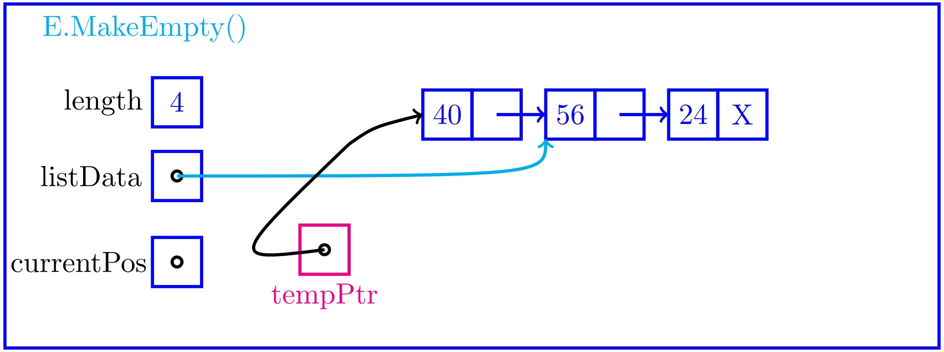
```
//Use of MakeEmpty()
#include <iostream>
#include "unsorted.h"

using namespace std;

int main()
{
  UnsortedType list;
    ..............        // steps to insert items into list

  list.MakeEmpty();

 }
```

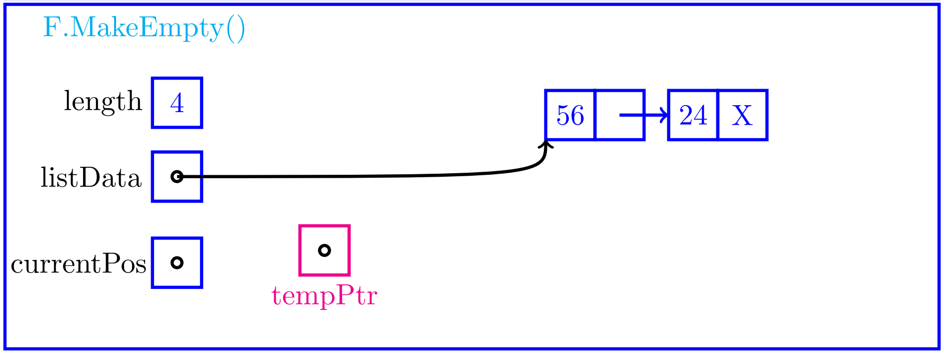


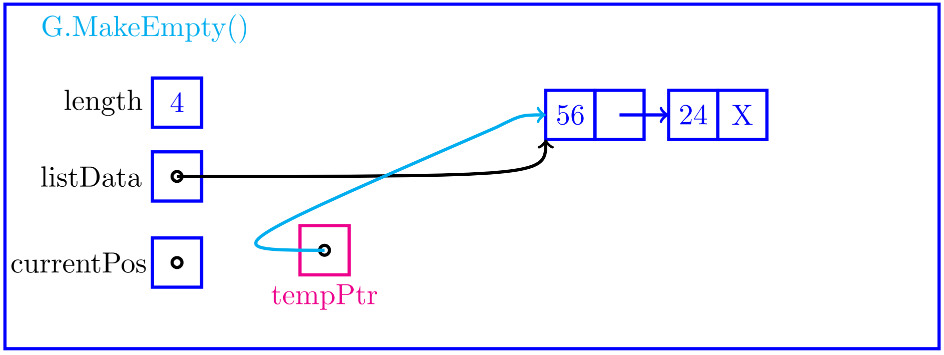


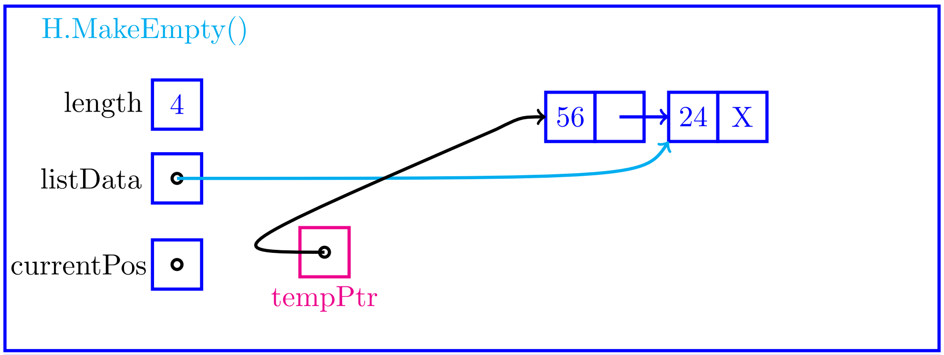



D.MakeEmpty()
length
4
listData
currentPos
40
56
24
X
tempPtr

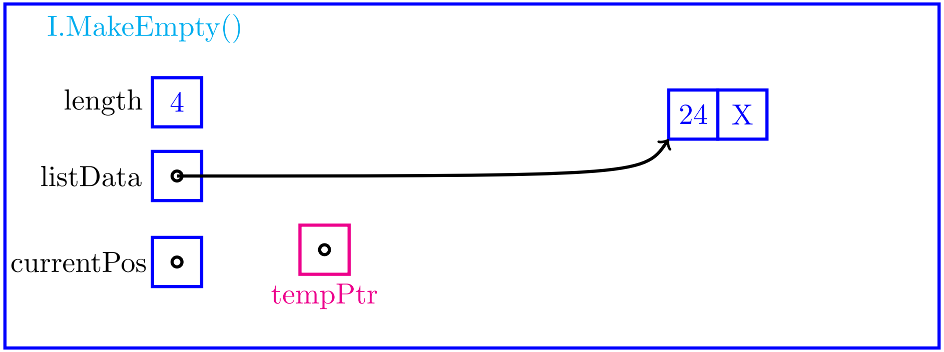

E.MakeEmpty()
length
4
listData
currentPos
40
56
24
X
tempPtr


F.MakeEmpty()
length
4
listData
currentPos
56
24
X
tempPtr

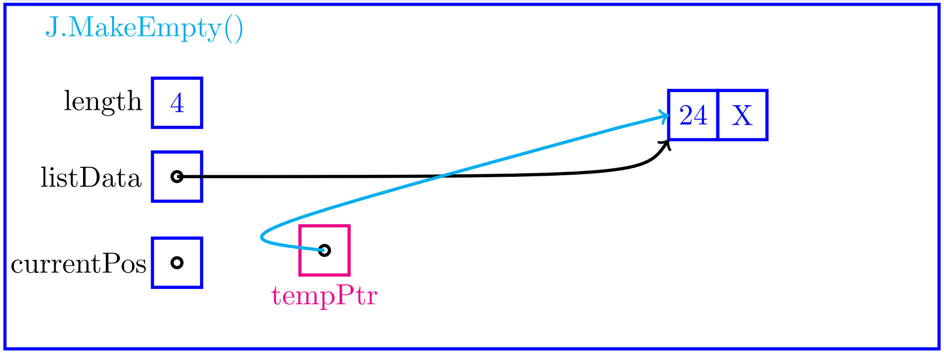

G.MakeEmpty()
length
4
listData
currentPos
56
24
X
tempPtr



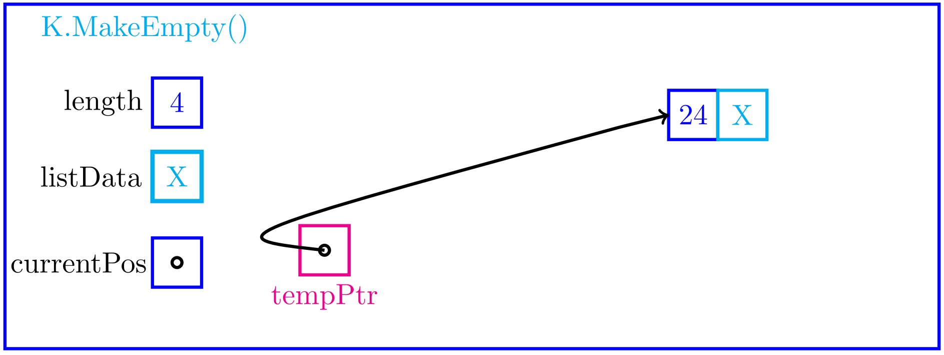
H.MakeEmpty()



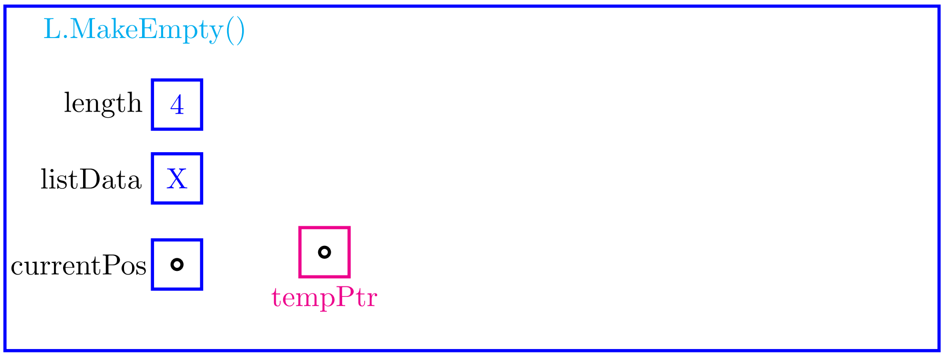
I.MakeEmpty()



J.MakeEmpty()



K.MakeEmpty()


L.MakeEmpty()
length
4
listData
X
currentPos
tempPtr