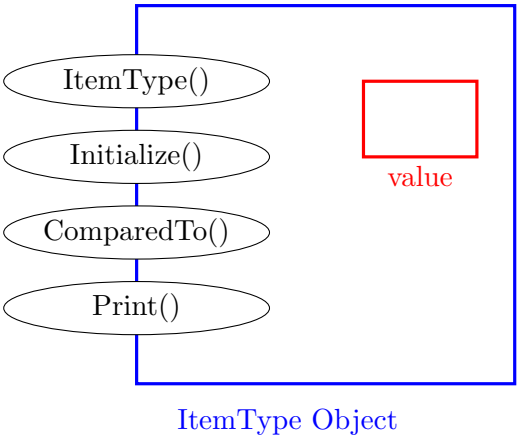
List Usage Example

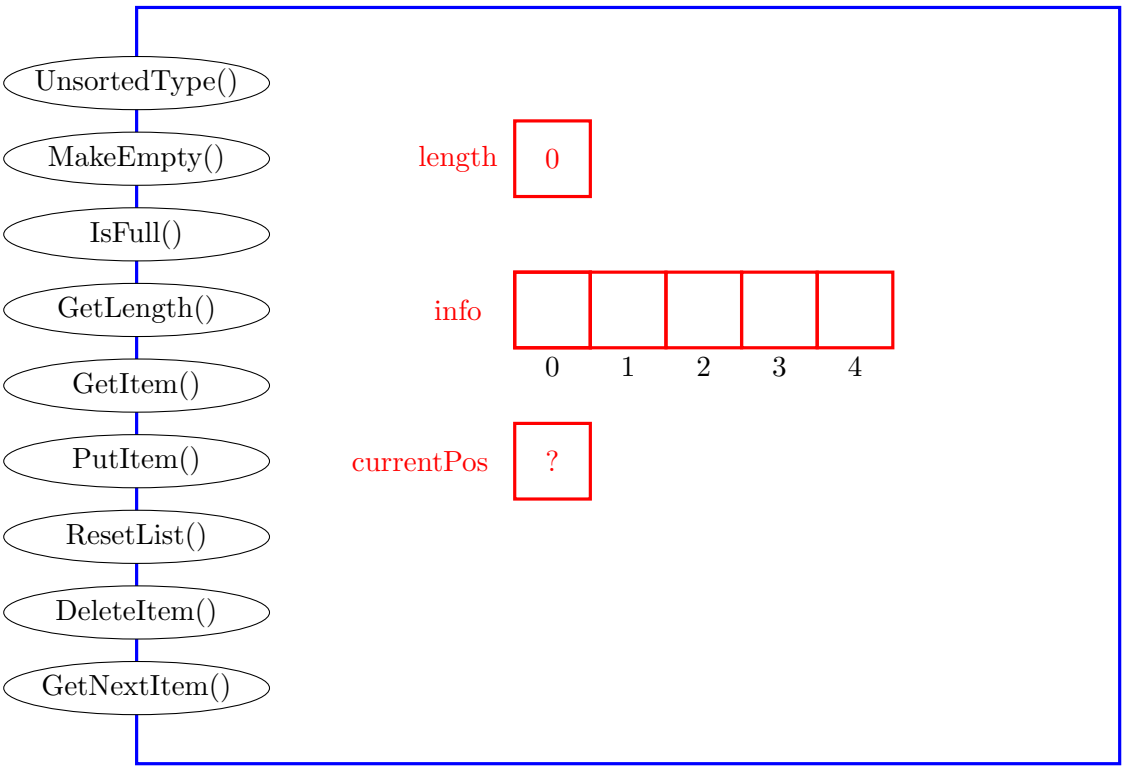
CSE 225 - Data Structures and Algorithms

Md. Mahfuzur Rahman
ECE Department
North South University

1 ItemType


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

2 Array Based Unsorted List

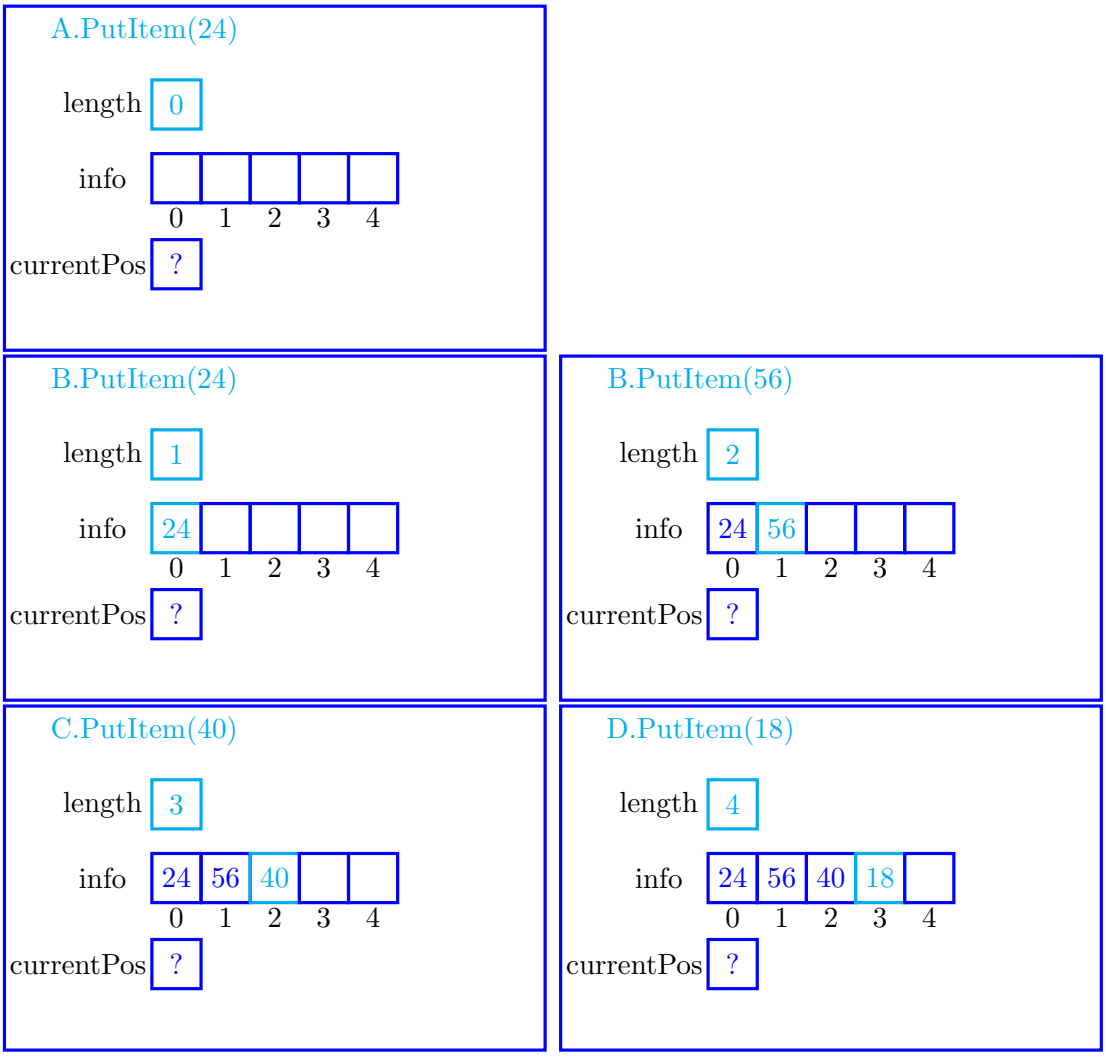


Array Based Unsorted List Object

2.1 Use of PutItem()

```
1  //Use of PutItem()
2 #include "unsorted.h"
3 using namespace std;
4 int main()
5 {
6   UnsortedType list;
7
8   ItemType item1;
9   item1.Initialize(24);
10  list.PutItem(item1);
11
12  ItemType item2;
13  item2.Initialize(56);
```

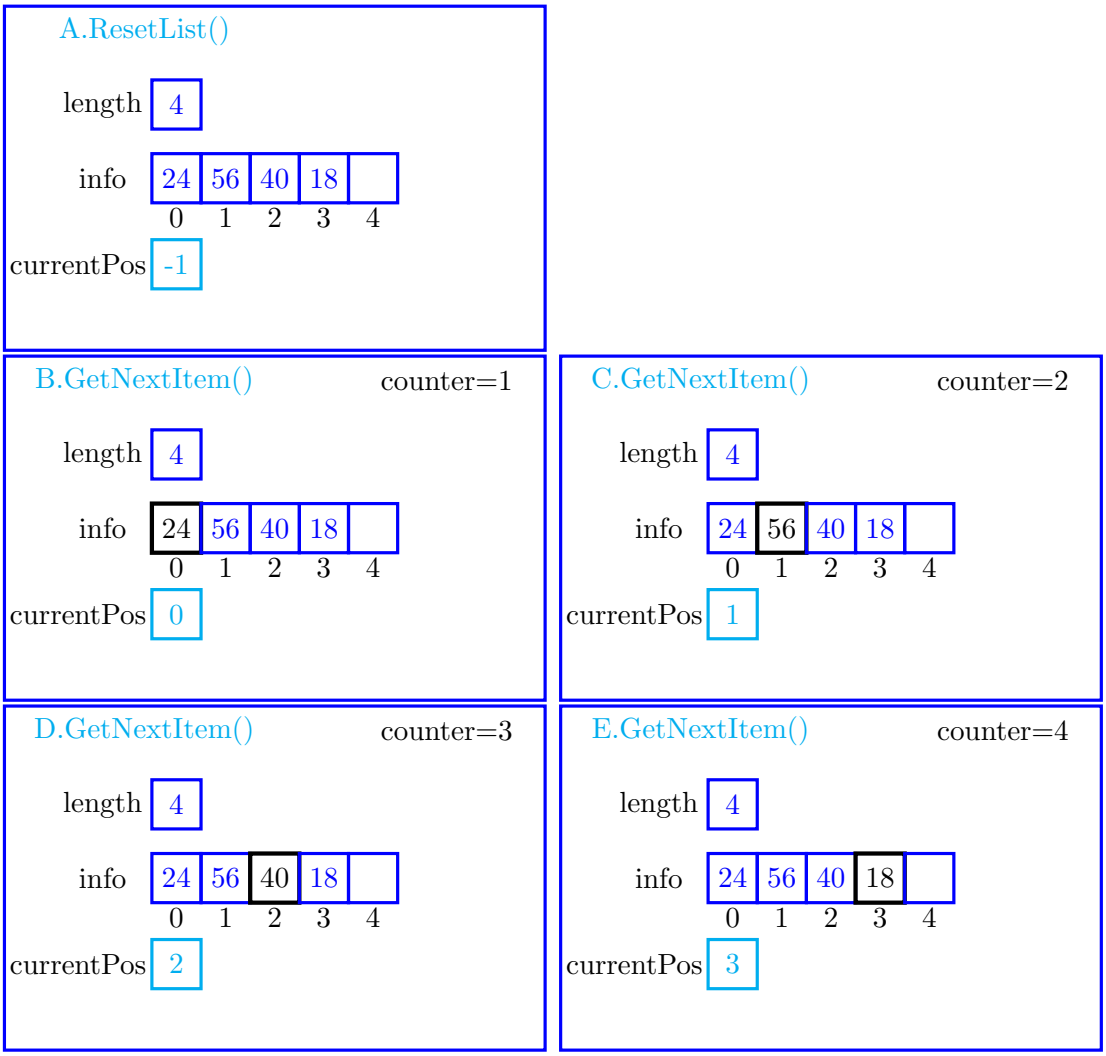
```
14    list.PutItem(item2);
15
16    ItemType item3;
17    item3.Initialize(40);
18    list.PutItem(item3);
19
20    ItemType item4;
21    item3.Initialize(18);
22    list.PutItem(item4);
23    return 0;
24 }
```



2.2 Use of ResetList() and GetNextItem()

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

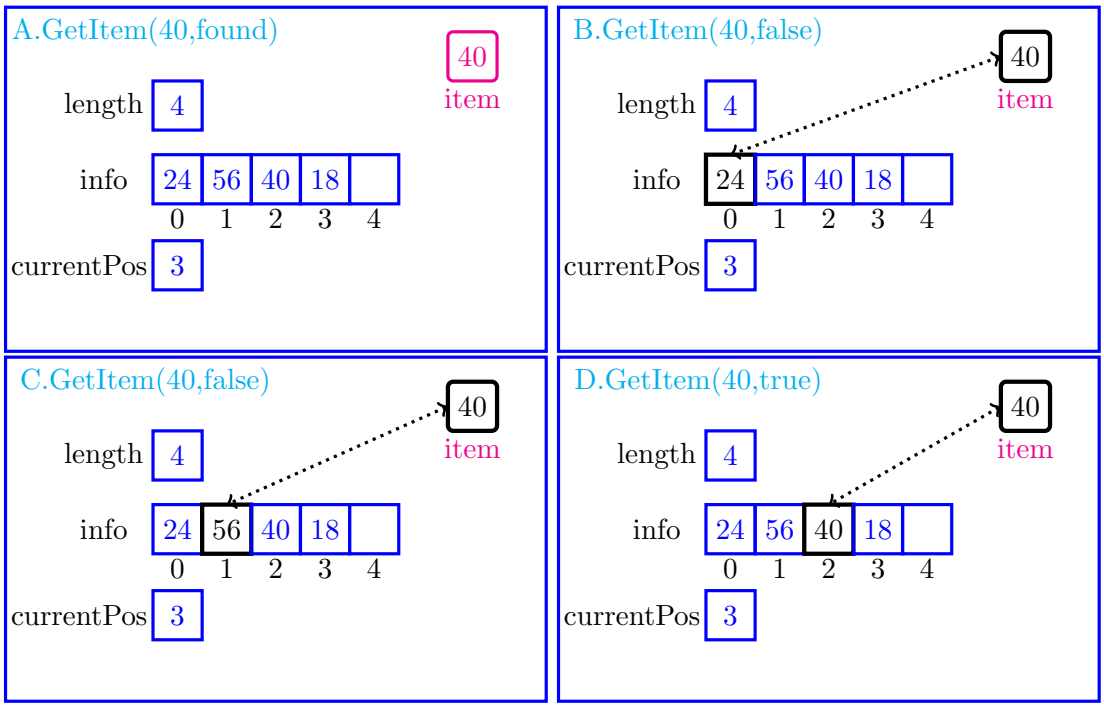
```cpp
#include <fstream>
#include <string>
#include <cctype>
#include <cstring>

#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()

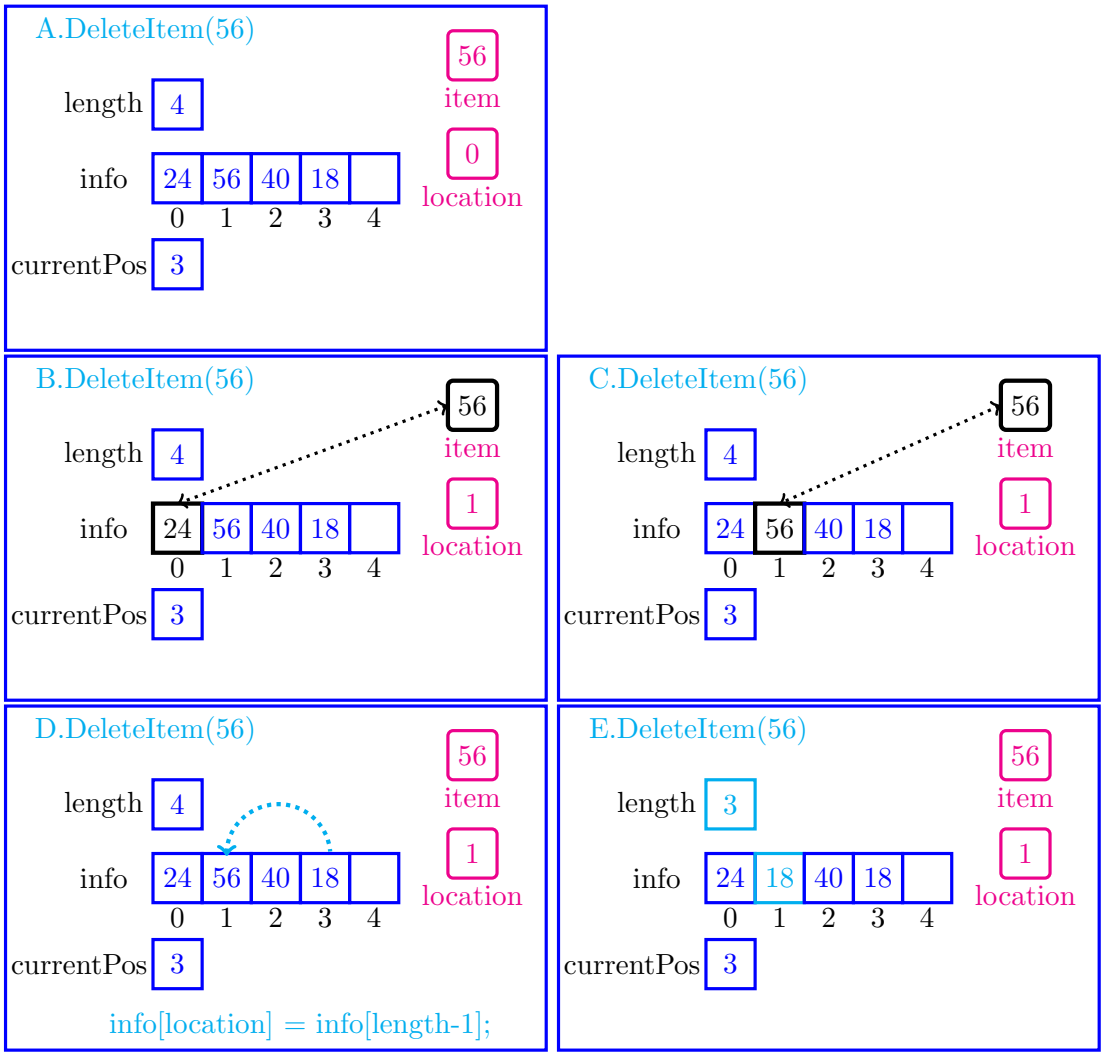
```
1  //Use of GetItem()
2  #include <iostream>
3
4  #include "unsorted.h"
5
6  using namespace std;
7
8  int main()
9  {
10   UnsortedType list;
11     .............       // steps to insert items into list
12
13   ItemType item;
14   bool found;
15   int number = 40;
```

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

}
```



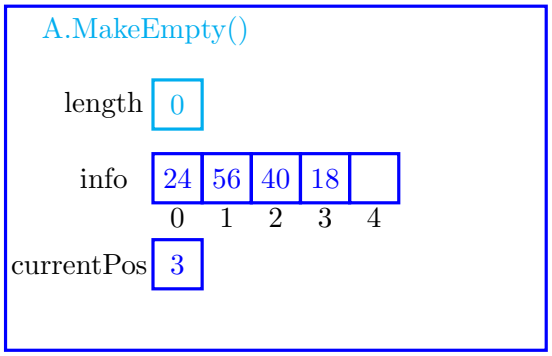
2.4 Use of DeleteItem()

```
//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);

}
```

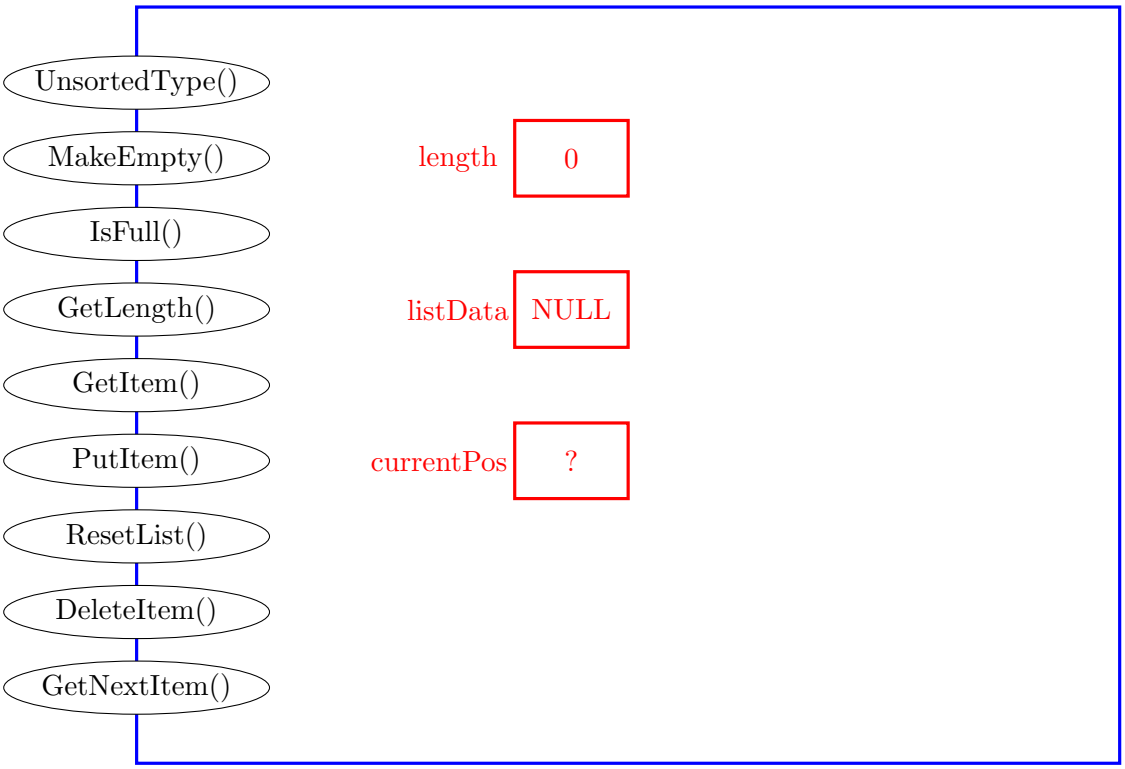


2.5 Use of MakeEmpty()

```
1  //Use of MakeEmpty()
2  #include <iostream>
3  #include "unsorted.h"
4
5  using namespace std;
6
7  int main()
8  {
9    UnsortedType list;
10     .............        // steps to insert items into list
11
12   list.MakeEmpty();
13
14  }
```


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

3 Linked-List Based Unsorted List

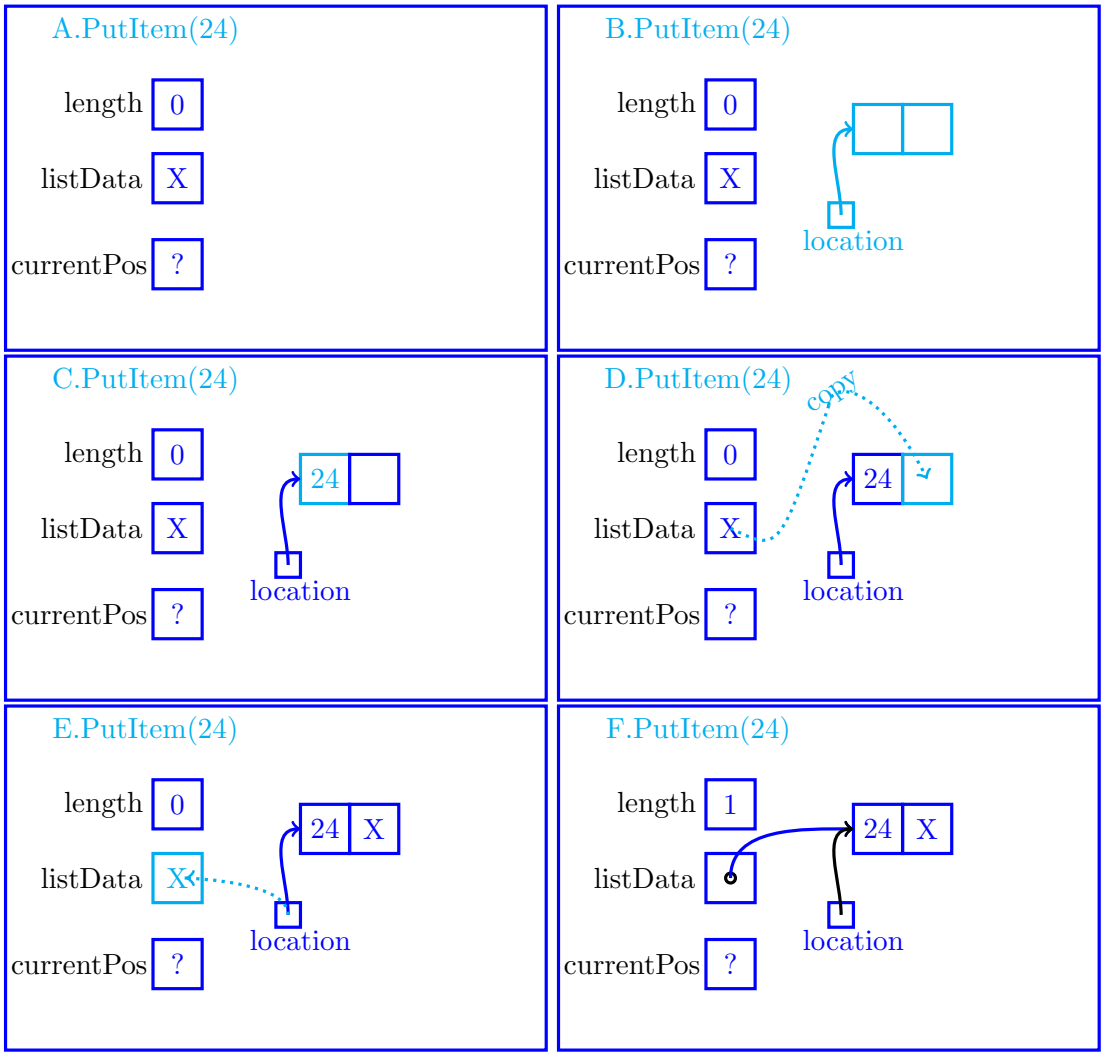


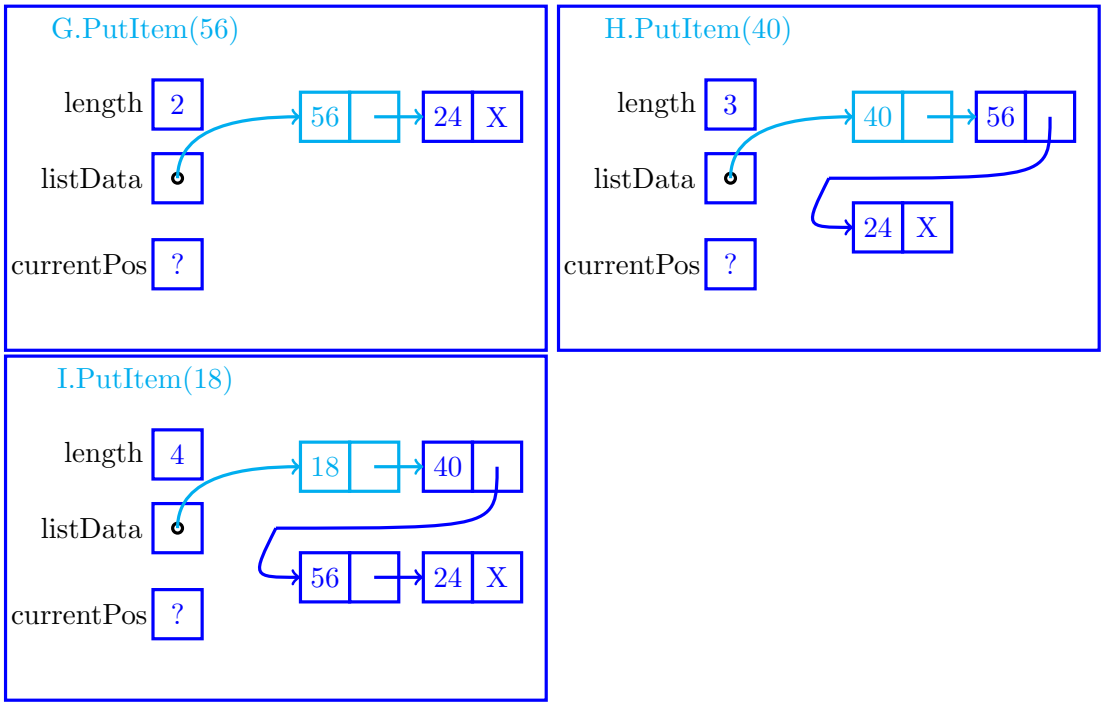
Linked-List Based Unsorted List Object

3.1 Use of PutItem()

```
1  //Use of PutItem()
2 #include "unsorted.h"
3 using namespace std;
4 int main()
5 {
6   UnsortedType list;
7
8   ItemType item1;
9   item1.Initialize(24);
10  list.PutItem(item1);
11
12  ItemType item2;
13  item2.Initialize(56);
```

```
14    list.PutItem(item2);
15
16    ItemType item3;
17    item3.Initialize(40);
18    list.PutItem(item3);
19
20    ItemType item4;
21    item3.Initialize(18);
22    list.PutItem(item4);
23    return 0;
24  }
```

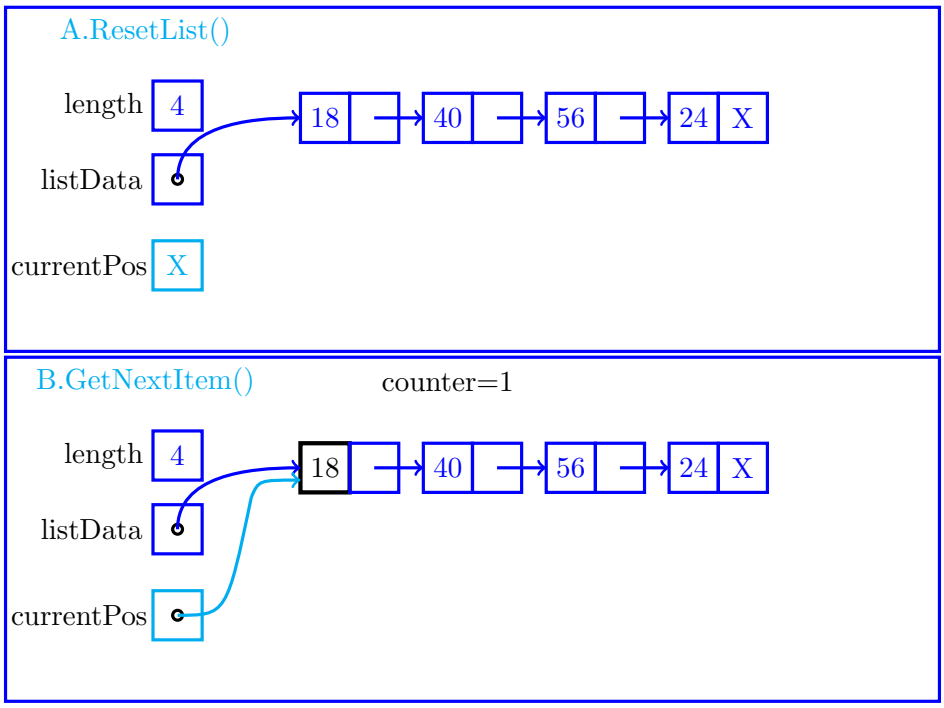


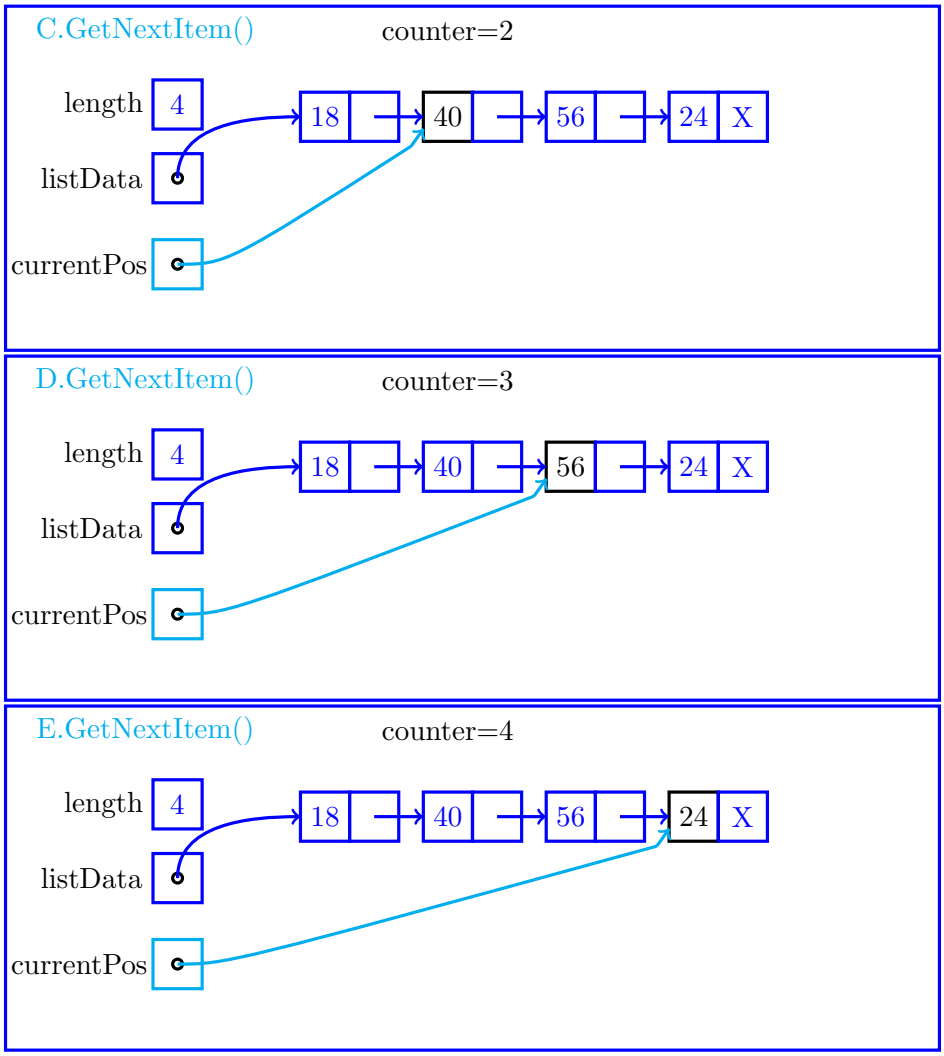


3.2 Use of ResetList() and GetNextItem()

```
1  //Use of ResetList() and GetNextItem()
2  #include <iostream>
3  #include <fstream>
4  #include <string>
5  #include <cctype>
6  #include <cstring>
7
8  #include "unsorted.h"
9
10 using namespace std;
11 void PrintList(ofstream& outFile, UnsortedType& list);
12
13 int main()
14 {
15
16   UnsortedType list;
17
18   .............       // steps to insert items into list
19
20   ofstream outFile;      // file containing output
21   string outFileName;    // output file external name
22
23   cout << "Enter name of output file; press return." << endl;
24   cin  >> outFileName;
25   outFile.open(outFileName.c_str());
26   PrintList(outFile, list);
27
```

```
28  }
29
30
31  void PrintList(ofstream& dataFile, UnsortedType& list)
32  {
33    int length;
34    ItemType item;
35    dataFile << "PrintList" << endl;
36    list.ResetList();
37    length = list.GetLength();
38    if (length == 0)
39      dataFile << "List is empty.";
40    else
41      for (int counter = 1; counter <= length; counter++)
42      {
43        item = list.GetNextItem();
44        item.Print(dataFile);
45      }
46    dataFile << endl;
47  }
```

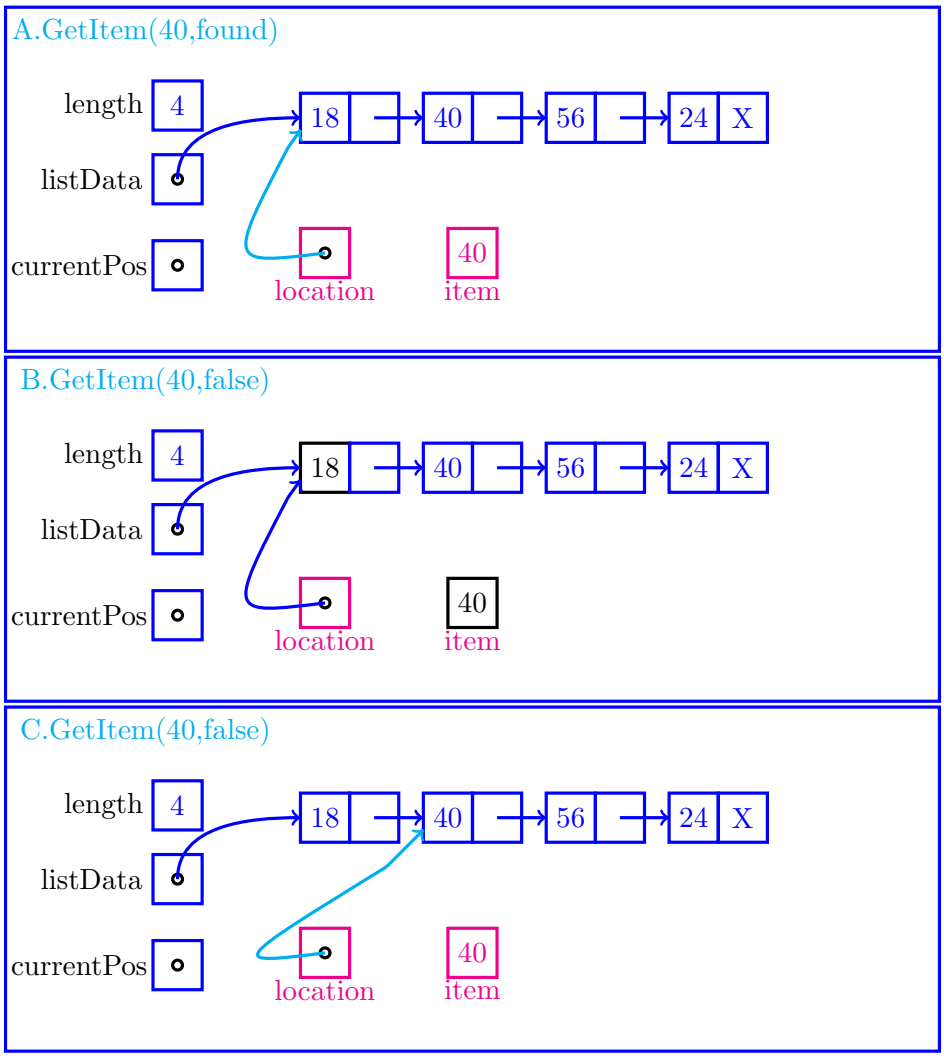


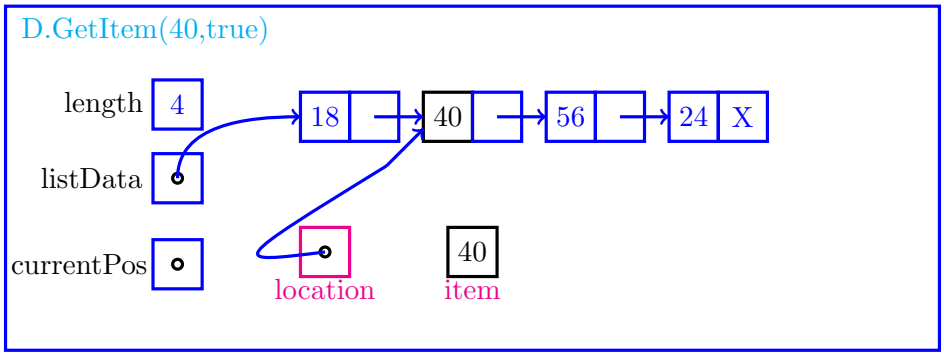


3.3 Use of GetItem()

```
1  //Use of GetItem()
2  #include <iostream>
3
4  #include "unsorted.h"
5
6  using namespace std;
7
8  int main()
9  {
10   UnsortedType list;
11     .............       // steps to insert items into list
12
13   ItemType item;
14   bool found;
15   int number = 40;
```

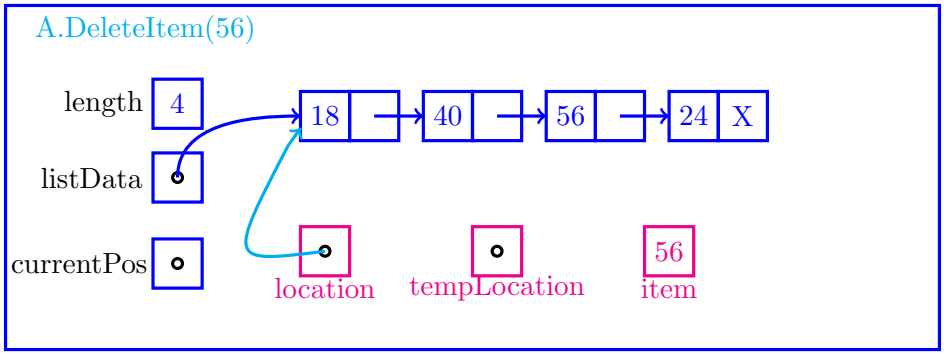
```
16   item.Initialize(number);
17   item = list.GetItem(item, found);
18   if (found)
19     cout << number << " found in list." << endl;
20   else cout << number  << " not in list."  << endl;
21
22 }
```

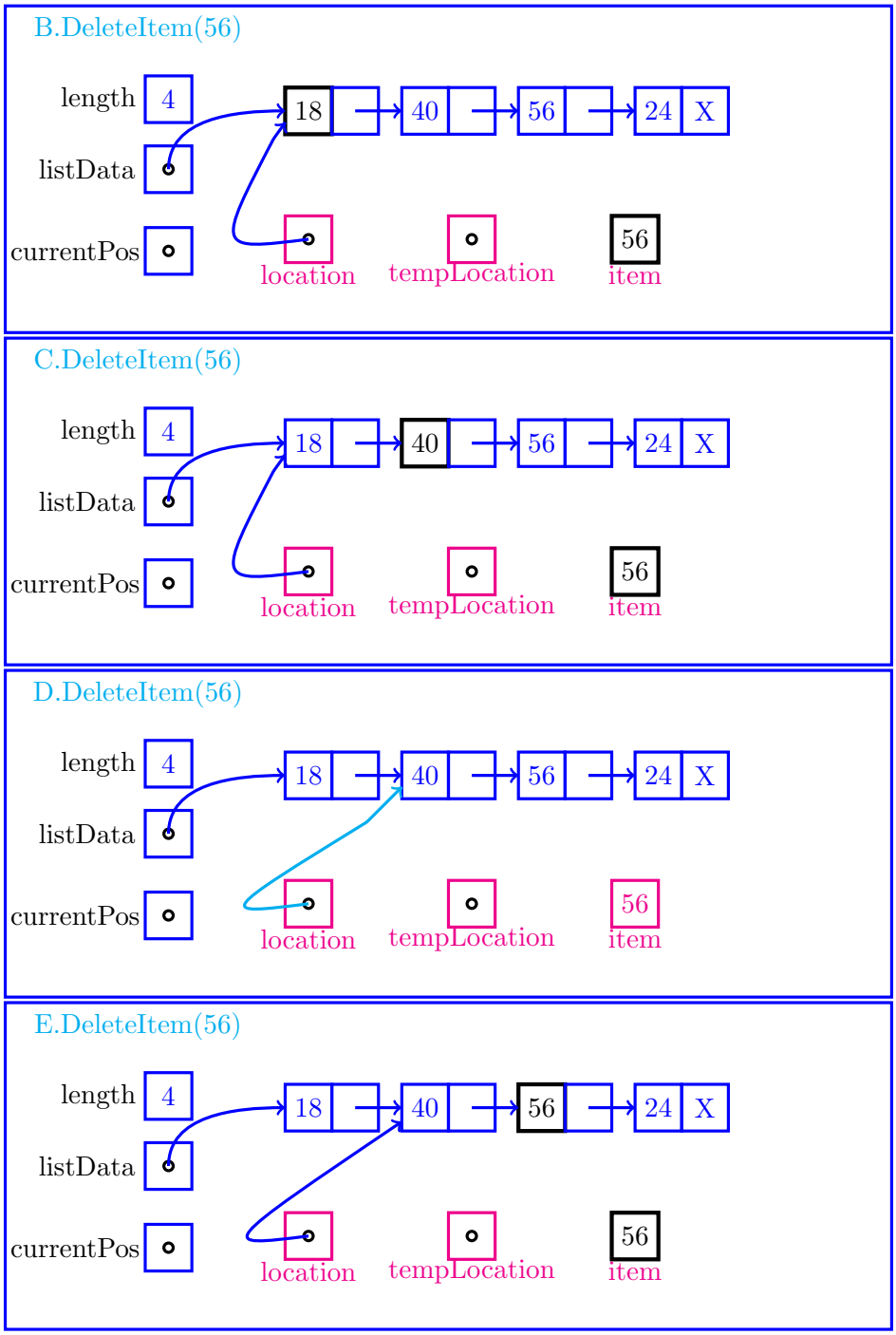


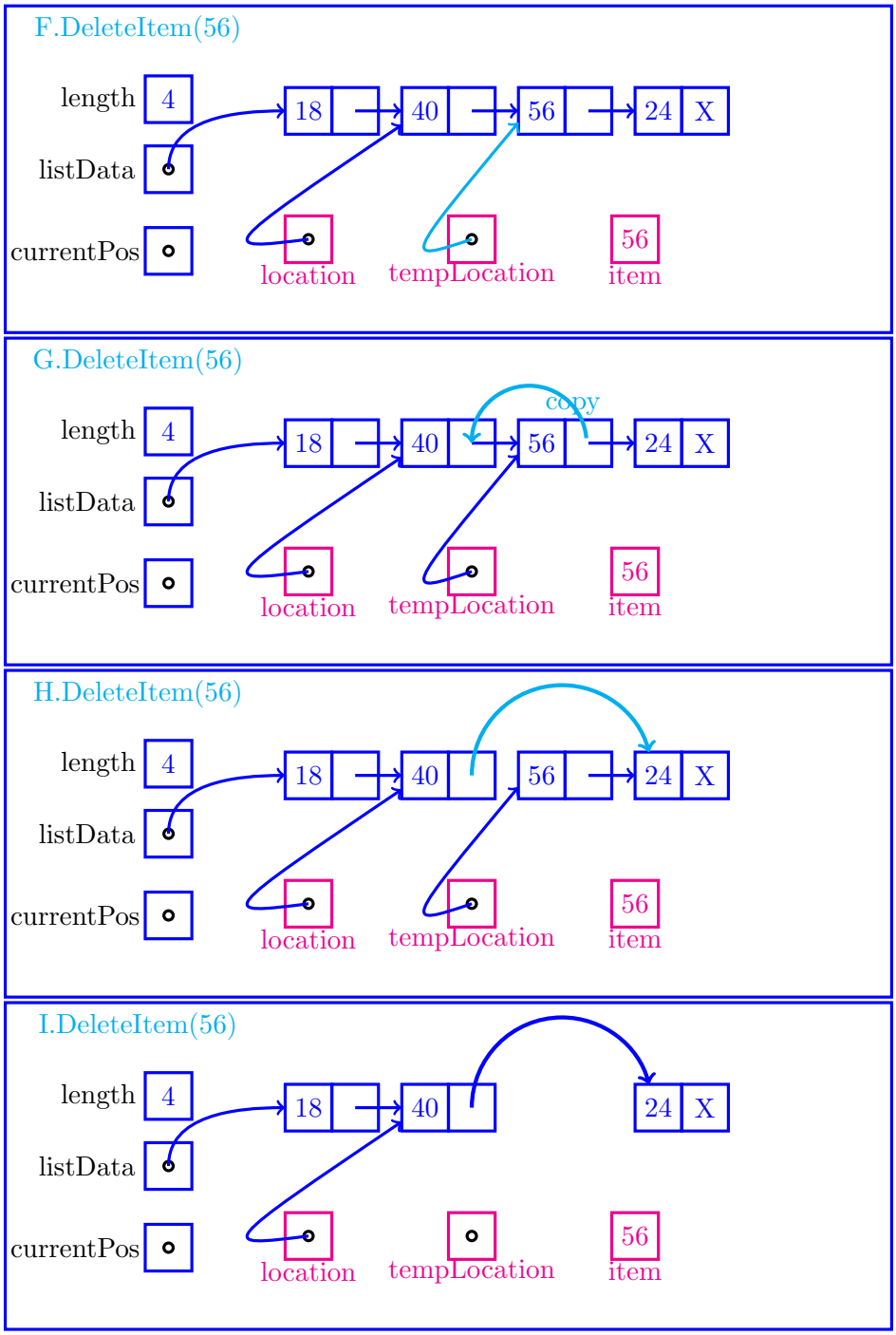


3.4 Use of DeleteItem()

```
1  //Use of DeleteItem()
2  #include <iostream>
3  #include "unsorted.h"
4
5  using namespace std;
6
7  int main()
8  {
9    UnsortedType list;
10     .............        // steps to insert items into list
11
12   ItemType item;
13   int number = 56;
14   item.Initialize(number);
15   list.DeleteItem(item);
16
17 }
```

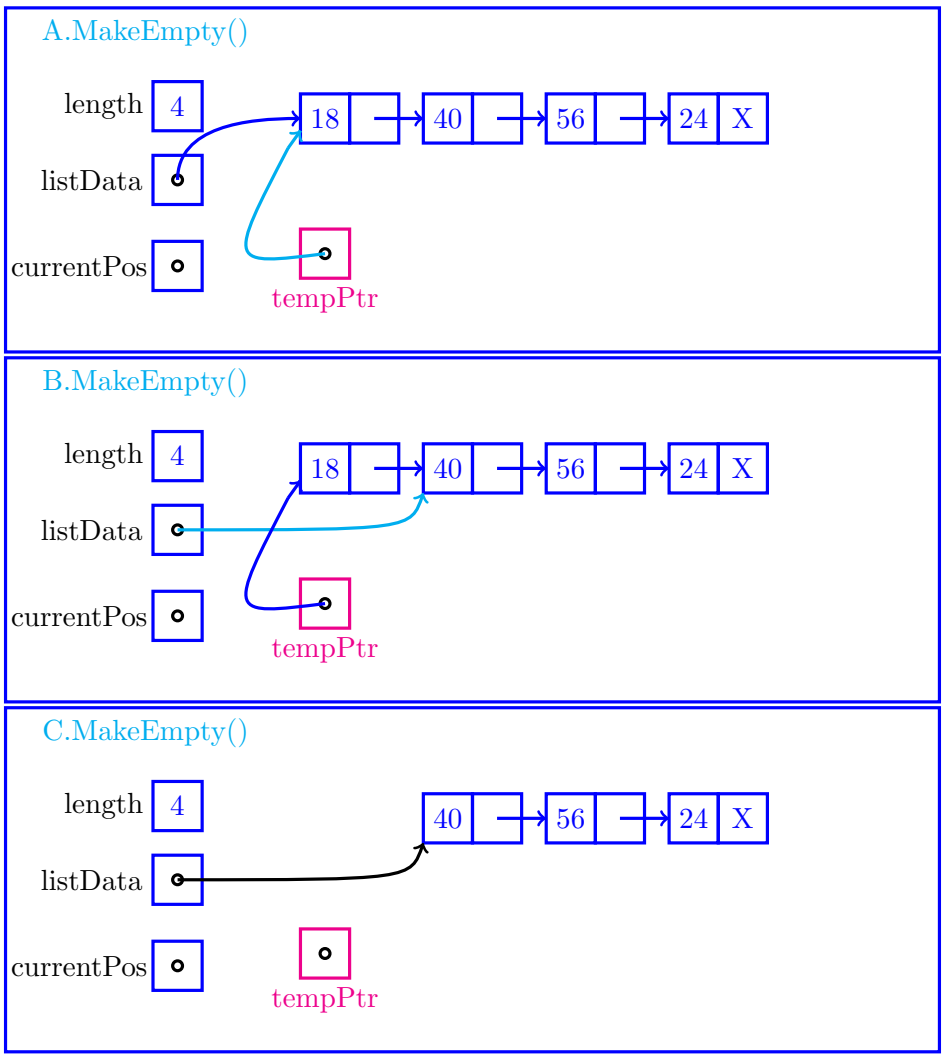


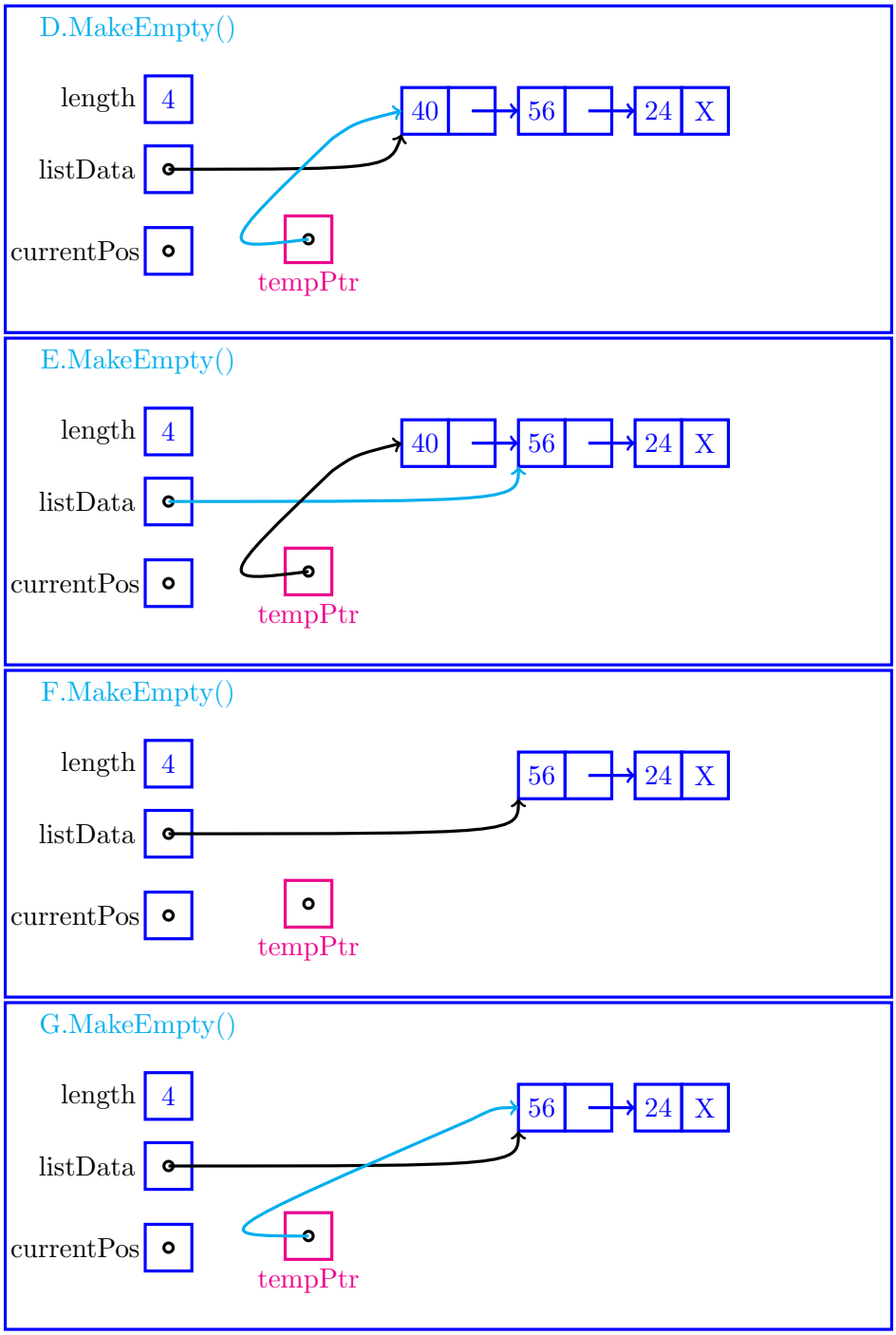

B.DeleteItem(56)
length
4
listData
currentPos
18
40
56
24
X
location
tempLocation
56
item
C.DeleteItem(56)
length
4
listData
currentPos
18
40
56
24
X
location
tempLocation
56
item
D.DeleteItem(56)
length
4
listData
currentPos
18
40
56
24
X
location
tempLocation
56
item
E.DeleteItem(56)
length
4
listData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

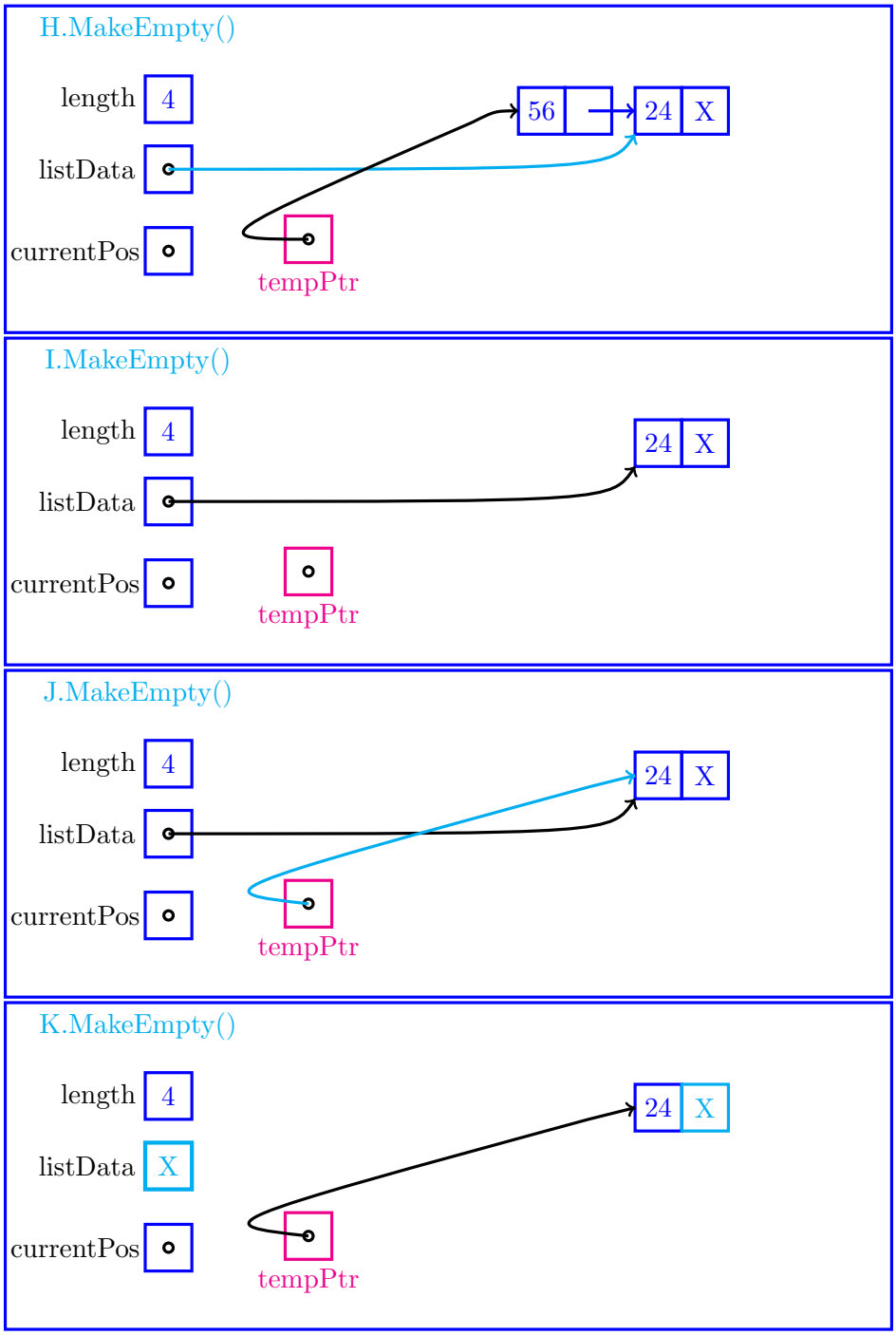


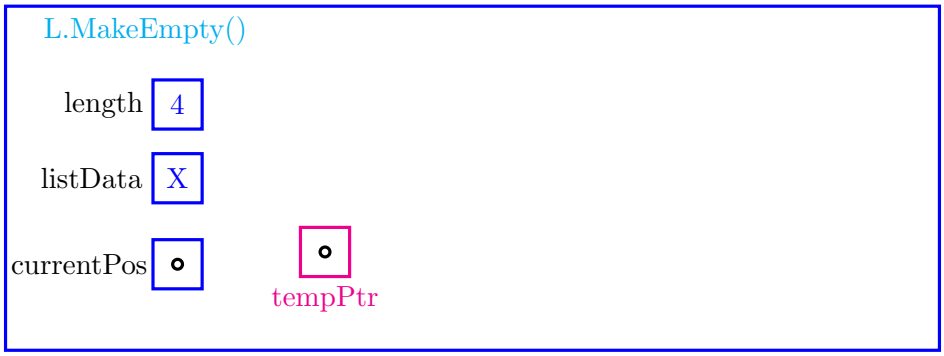
3.5 Use of MakeEmpty()

```cpp
//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()
length
4
listData
currentPos
56
24
X
tempPtr
I.MakeEmpty()
length
4
listData
currentPos
24
X
tempPtr
J.MakeEmpty()
length
4
listData
currentPos
24
X
tempPtr
K.MakeEmpty()
length
4
listData
X
currentPos
24
X
tempPtr


L.MakeEmpty()
length
4
listData
X
currentPos
tempPtr