자료구조 실습02

Data Structures Lab02

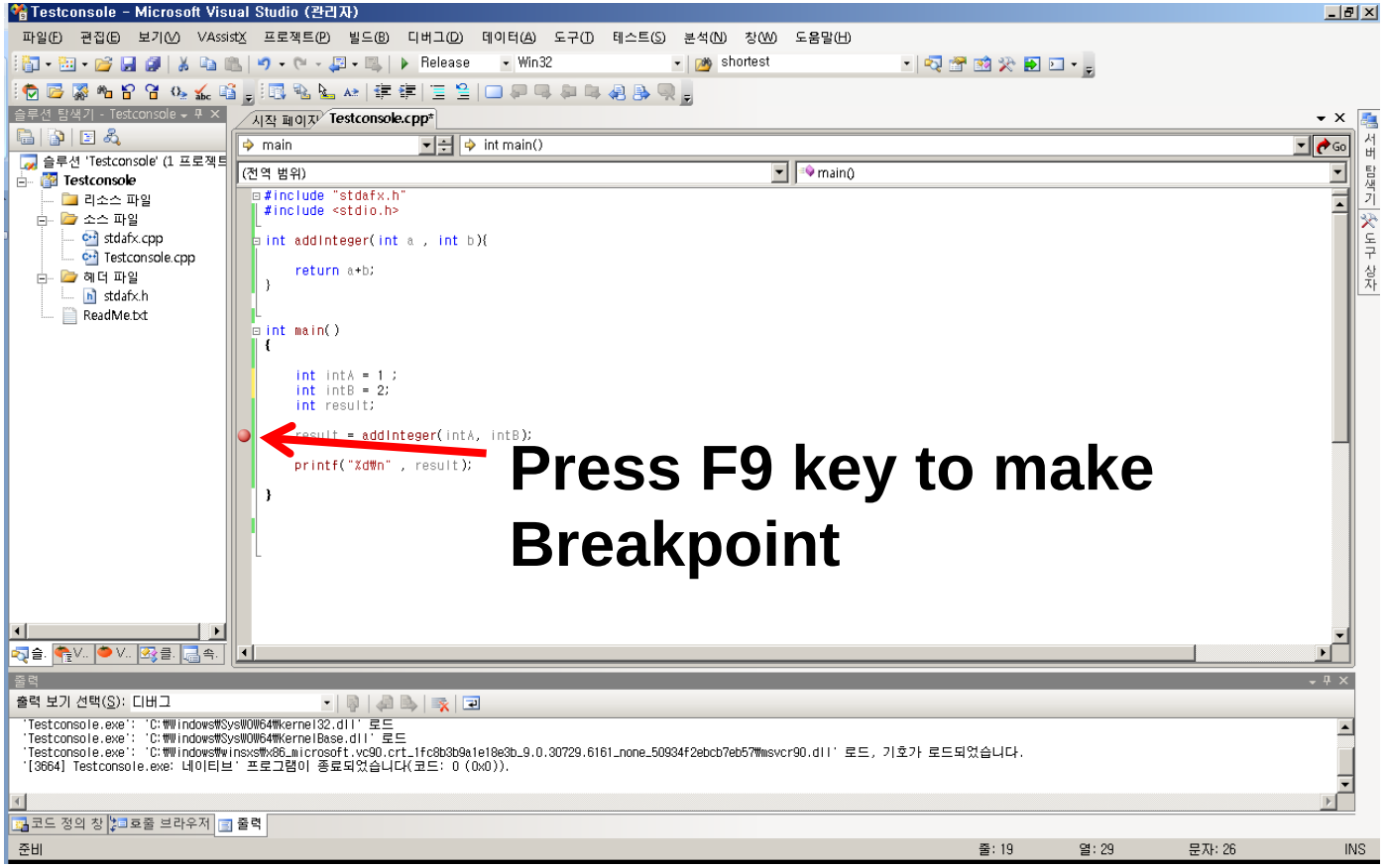
Basic Knowledge

- Debugger
  - A debugging tool that is used to test and debug other programs.
- Debuggee
  - A process or application upon which a debugger acts. The process that is being debugged.

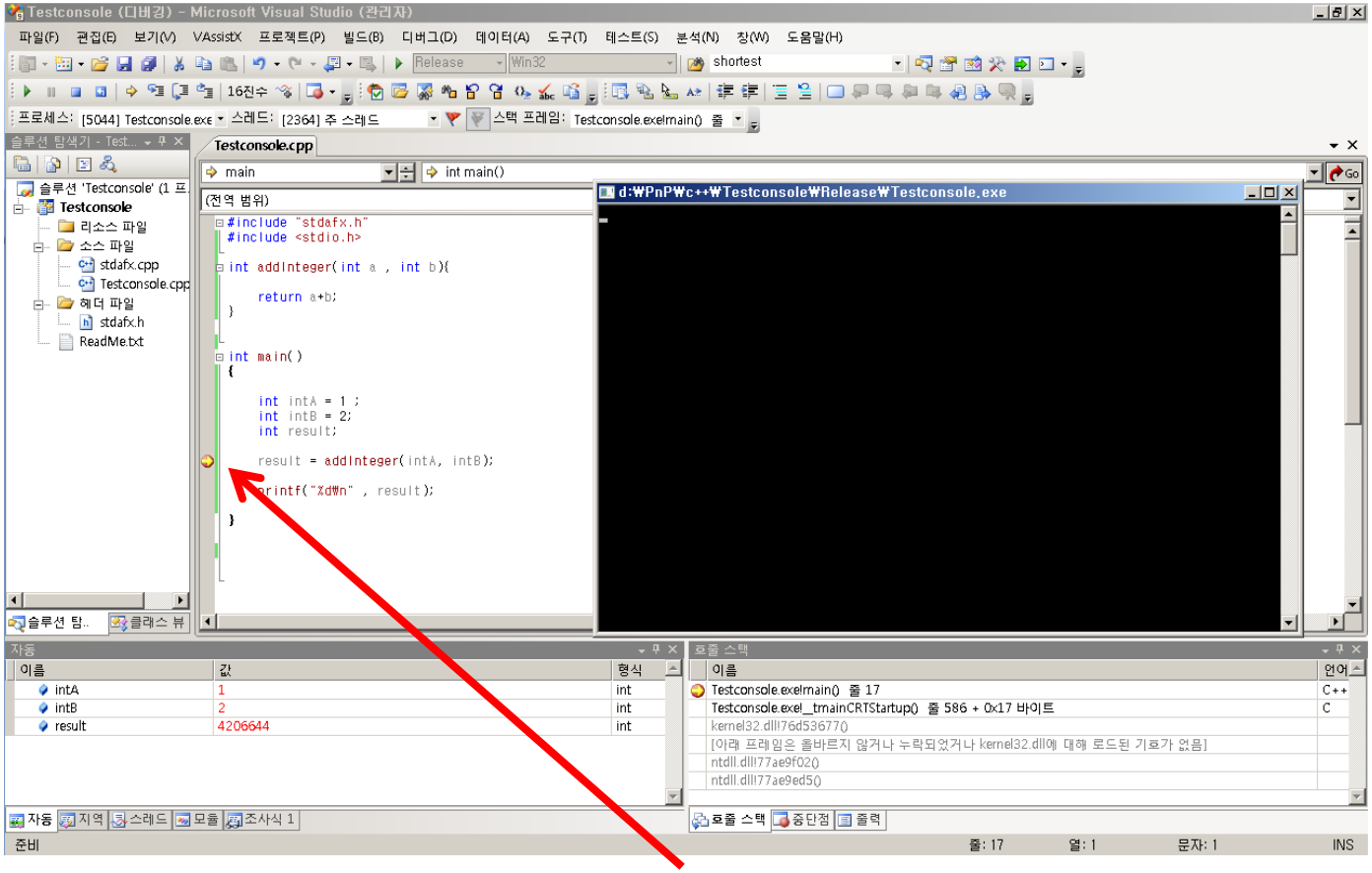
Sorts of Debug Shortcut (Visual Studio)

- Start Debugging(F5)
  - Start Debugging. Application will stop at the first Breakpoint.
- Stop Debugging(Shift + F5)
  - Stop Debugging. Debugger will terminate the application.
- Make Break Point(F9)
  - When you press “F9” key, red point will appear the headline of source code. Debugger will break the application when meet the break point. Breakpoint can be made while coding time and debugging time.
- Step Into(F10)
  - Run into the function(method). Can’t enter library function(method) as like API, Standard library.
- Step Over(F11)
  - Run over the function(method).

Set ‘Break point’

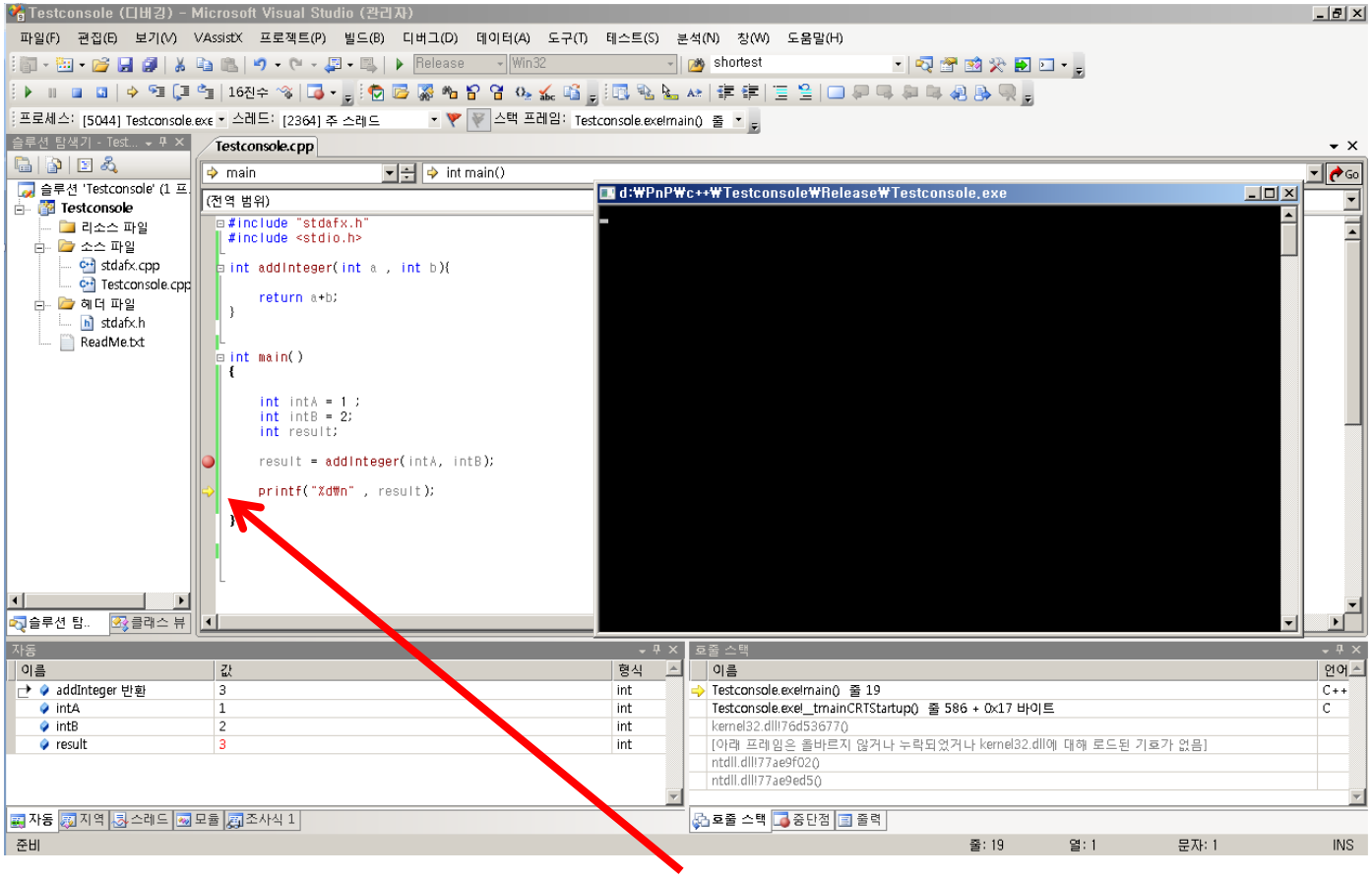


Run debug mode



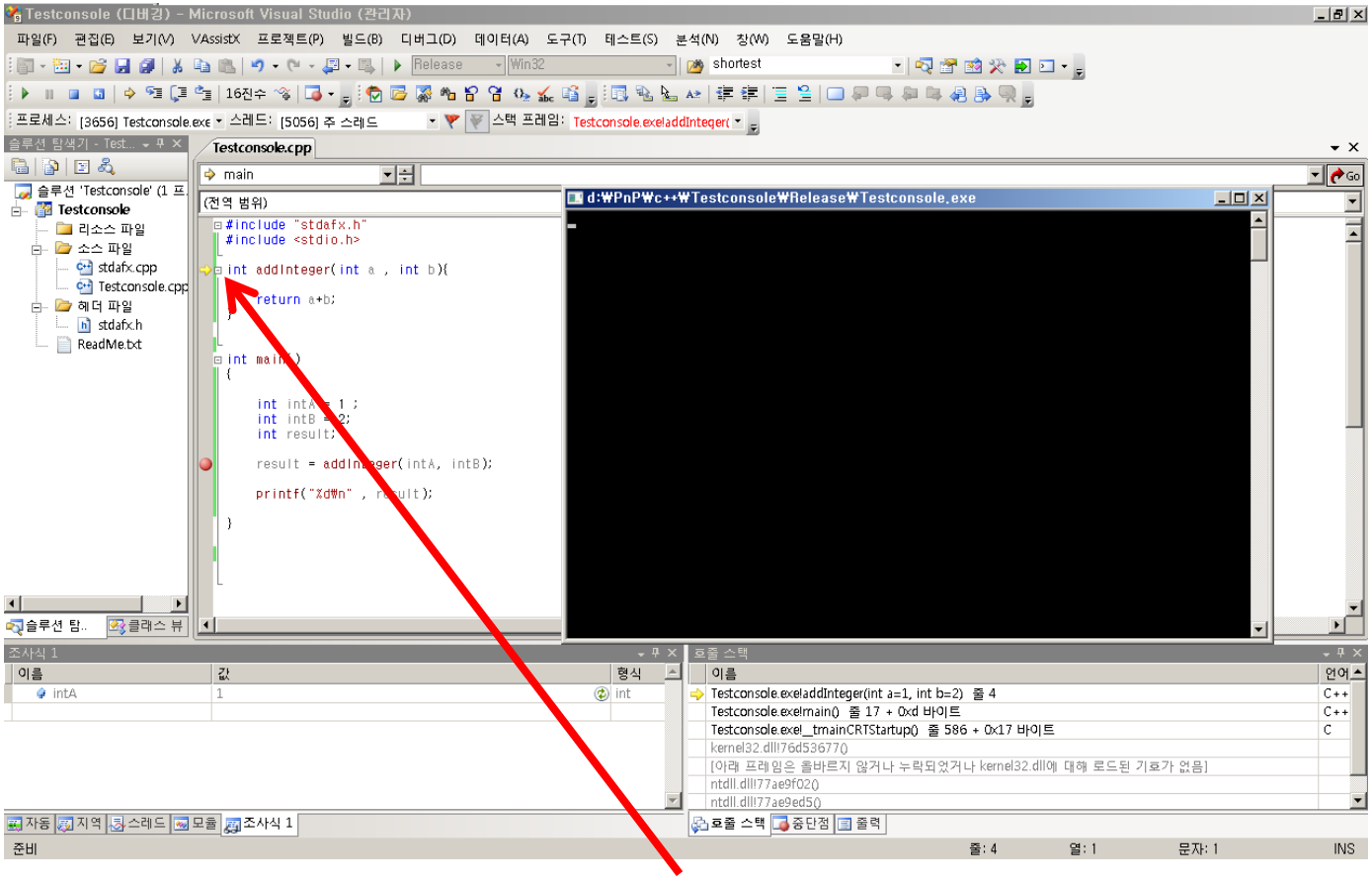
- Press F5 key to enter debug mode. Application will stop at first breakpoint.
- Yellow arrow indicates the source code line that will be implemented very next time.

Step over (F10)



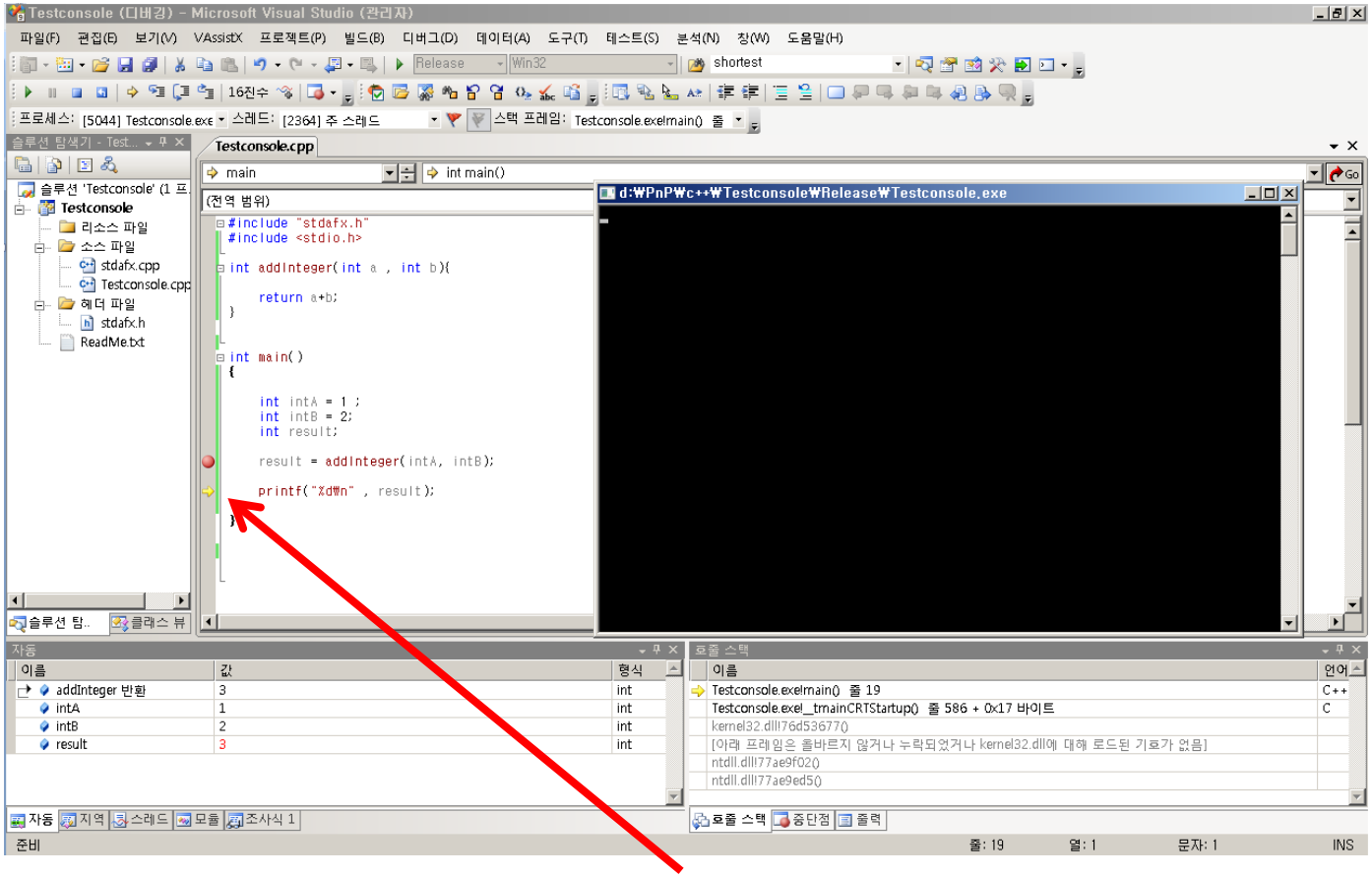
- Press F10 key to step over the function “addinteger()”.
- Yellow arrow will indicate very next line.

Step into (F11)



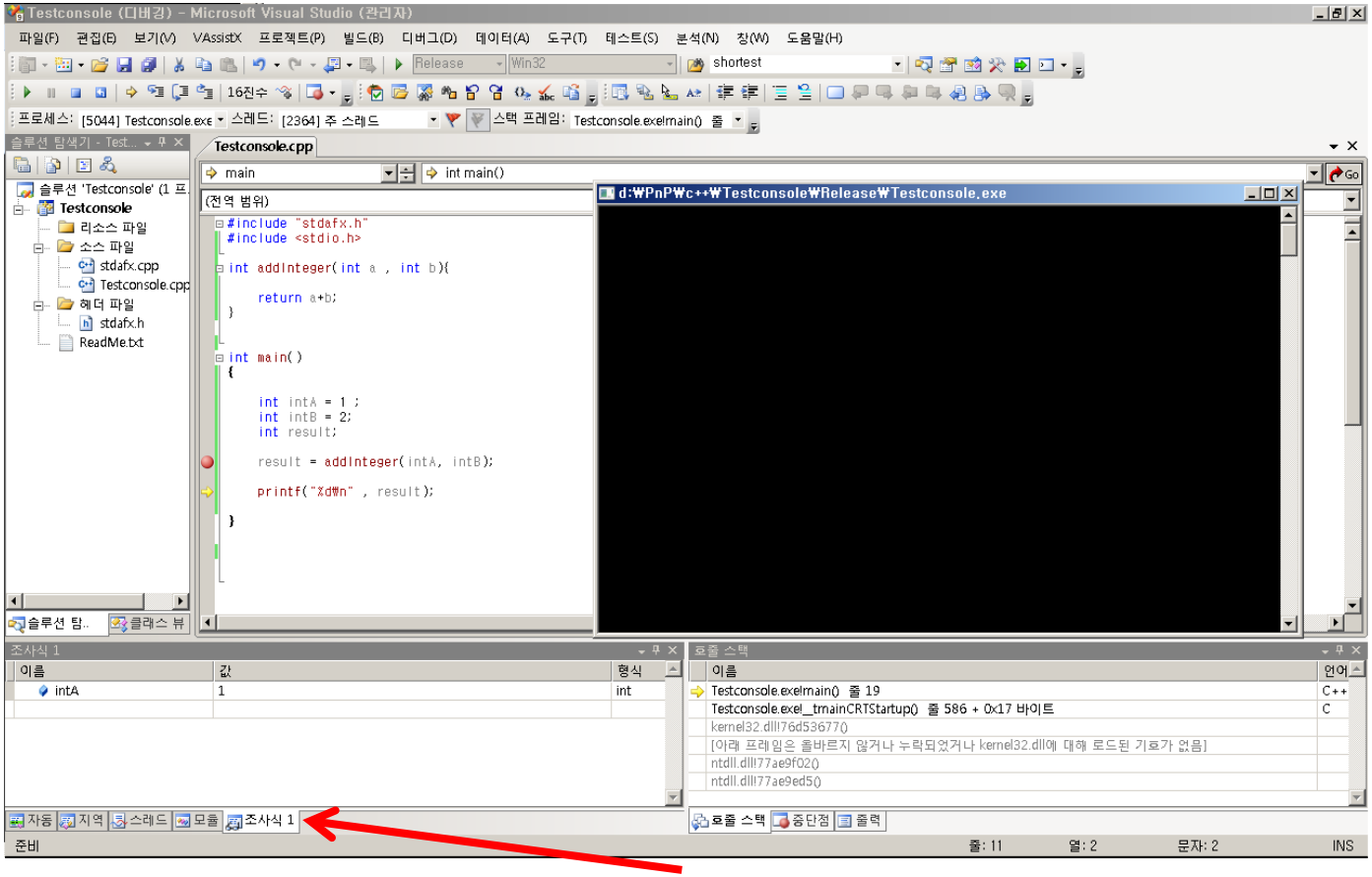
- Press F11 key to step into the function “addinteger()”.
- Yellow arrow will indicate the insight of function.

Check value of variables



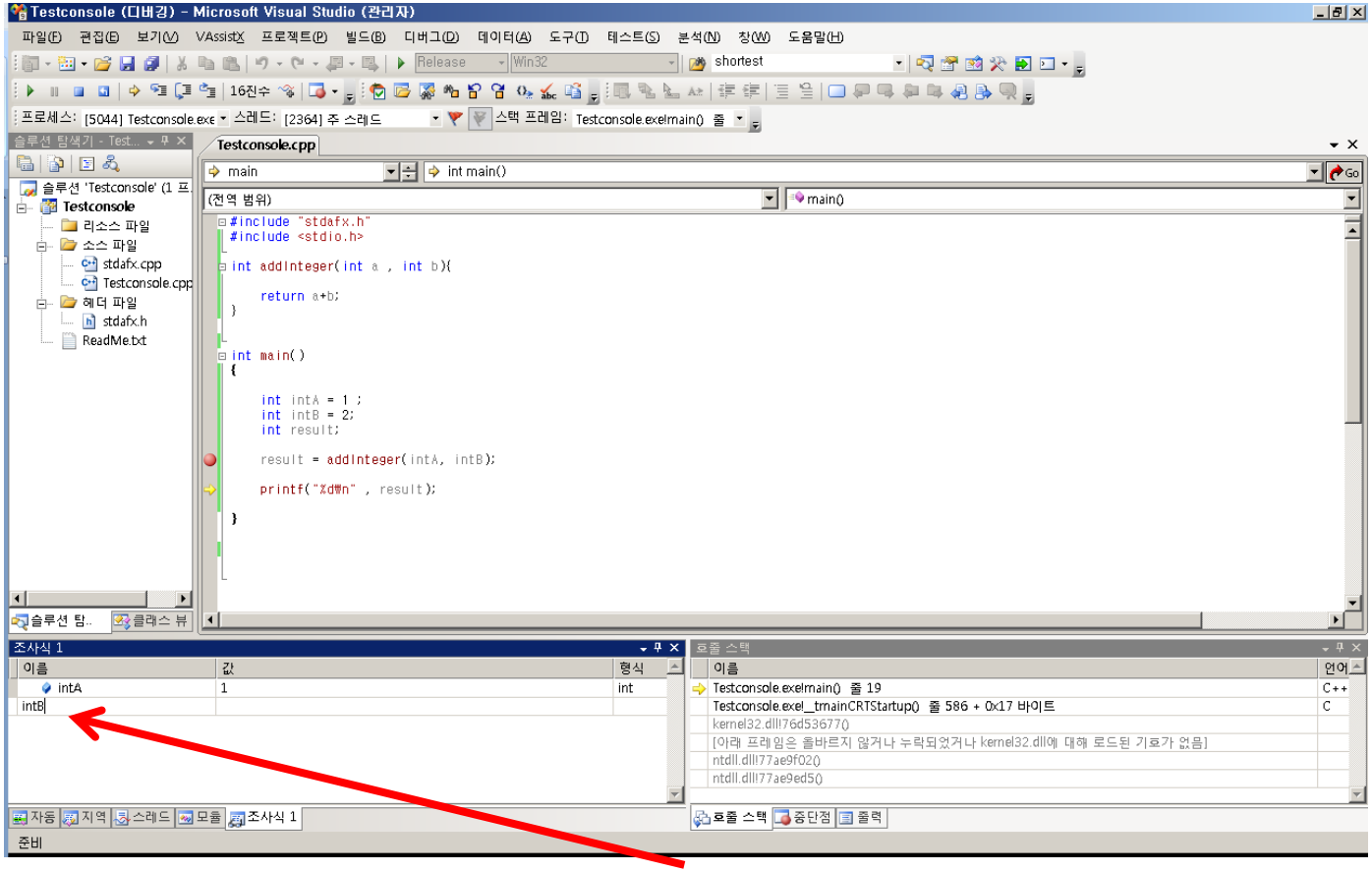
- You can confirm the variables' value.
- Red character means that the variable's value is changed.

Check value of variables



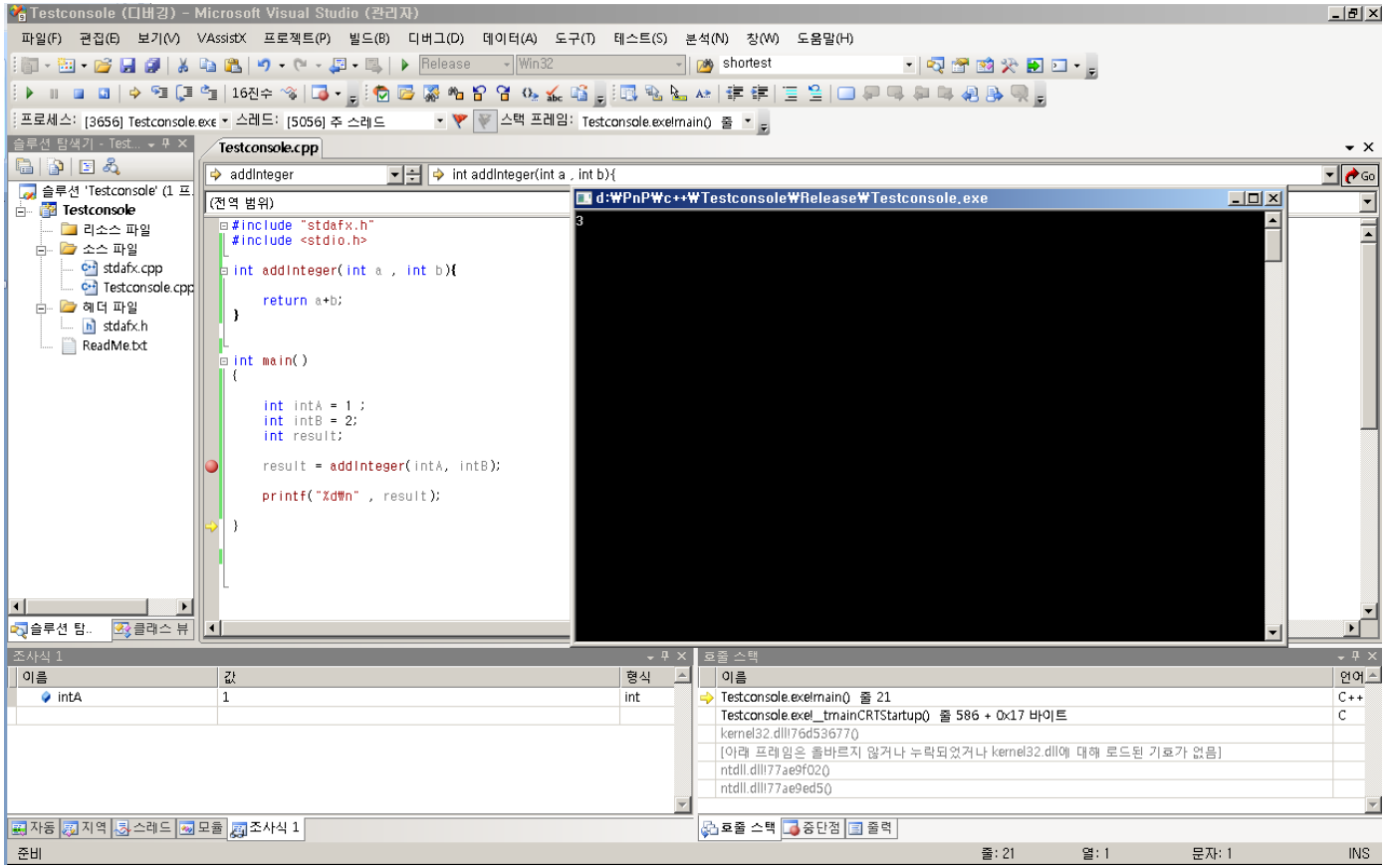
◎ You can confirm the variables' value by expression(조사식) window too.

Check value of variables



◎ You can also add variables that you want to confirm the value.

End of debug


Testconsole (디버깅) - Microsoft Visual Studio (관리자)
Testconsole.cpp
int addInteger(int a , int b){
d:₩PnP₩c++₩Testconsole₩Release₩Testconsole.exe
3
#include "stdafx.h"
#include <stdio.h>
int addInteger(int a , int b){
return a+b;
}
int main( )
{
int intA = 1 ;
int intB = 2;
int result;
result = addInteger(intA, intB);
printf("%d₩n" , result);
}

Lab02 예제(1/2)

◎ 목표: Sorted List의 ADT 설계 및 구현

◎ 내용:

☞ Array를 사용한 Unsorted List를 수정하여 Sorted List를 작성

☞ 과제: Lab01 기반의 iteration을 이용한 Sorted List 구현
학생 정보 검색 기능을 갖춘 Application 구현
Primary Key 검색을 위한 Binary Search 구현

◎ 방법

☞ Lab01과 제공된 ADT를 바탕으로 iteration을 이용한 Sorted List를 작성

Lab02 예제(2/2)

◎ 내용:

☞ UnSorted List를 Sorted List로 수정하기 위해 다음을 수정

Add(ItemType data);

- ➢ Primary key를 이용한 중복검사
- ➢ Primary key를 이용한 정렬

☞ Application class에 다음을 추가

SearchByID(); **// primary key인 id로 검색**

SearchByName(); **// name으로 검색 (해당 키워드가 포함된 모든 회원 출력)**

// ex) '김'으로 검색 → '김파래' '김연아' '김김김' 모두 출력

// Hint) string clss의 find() 함수 사용

예제: SortedList ADT

```
class SortedList
{
public:
        SortedListType();                       // default constructor
        ~SortedListType();                      // default destructor

        void MakeEmpty();                       // Make list empty(Initialize list)
        int GetLength();                        // Return the number of records in the list
        bool IsFull();                          // Check the list upper limit
        bool IsEmpty();                         // Check the list is empty
        void ResetList();                       // Initialize the iterator pointer
        int GetNextItem(ItemType& data);        // Update pointer to point to next record and return this new record.
        int Add(ItemType data);                 // Add a new data to list
        int Delete(ItemType data);              // Delete data record from list
        int Replace(ItemType data);             // Find to same record using primary key and replace it
        int Get(ItemType& target);              // Find the primary key in target from the list and copy the record to target
        int GetByBinarySearch(ItemType& data);          // find the primary key data from the list using binary
                                                        search and copy the record to data
private:
        ItemType m_Array[MAXSIZE];              // array for record
        int m_Length;                           // number of record
        int m_CurPointer;                       // current pointer
};
```

예제: console

◎ **Sorted List를 테스트할 driver를 다음과 같이 작성함**

```
--- ID – Command -----
        1 : Add item
        2 : Print all on screen
        3 : Make empty list
        4 : Search item by ID
        5 : Search item by Name
        5 : Delete item
        6 : Update item
        7 : Get from file
        8 : Put to file
        0 : Quit

Choose a Command -->
```

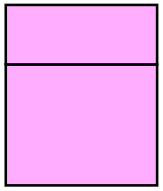
Lab02: Add (1/3)

◎ Add new item: Empty array

MAXSIZE: 10 Length: 0

index	0	1	2	3	4	5	6	7	8	9
item										

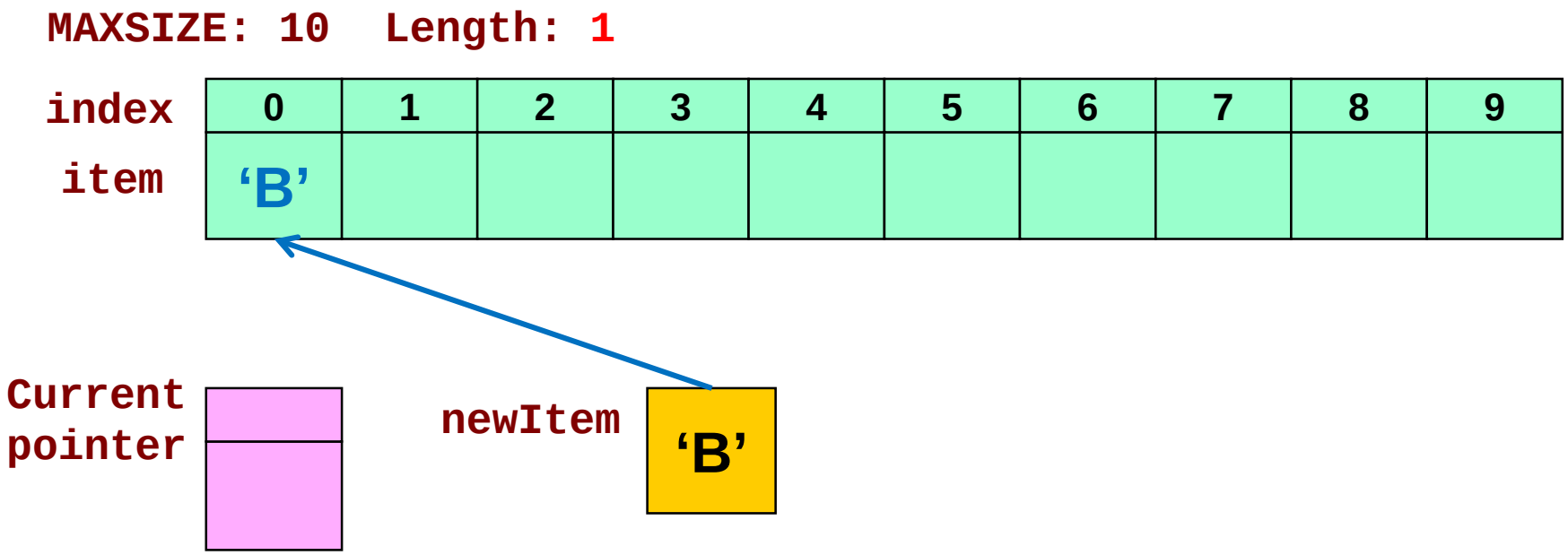
Current pointer



newItem **'B'**

Lab02: Add (1/3)

◎ Add new item: Empty array



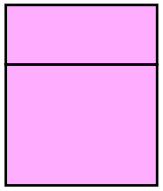
Lab02: Add (2/3)

◎ Add new item: Add to last

MAXSIZE: 10 Length: 1

index	0	1	2	3	4	5	6	7	8	9
item	'B'									

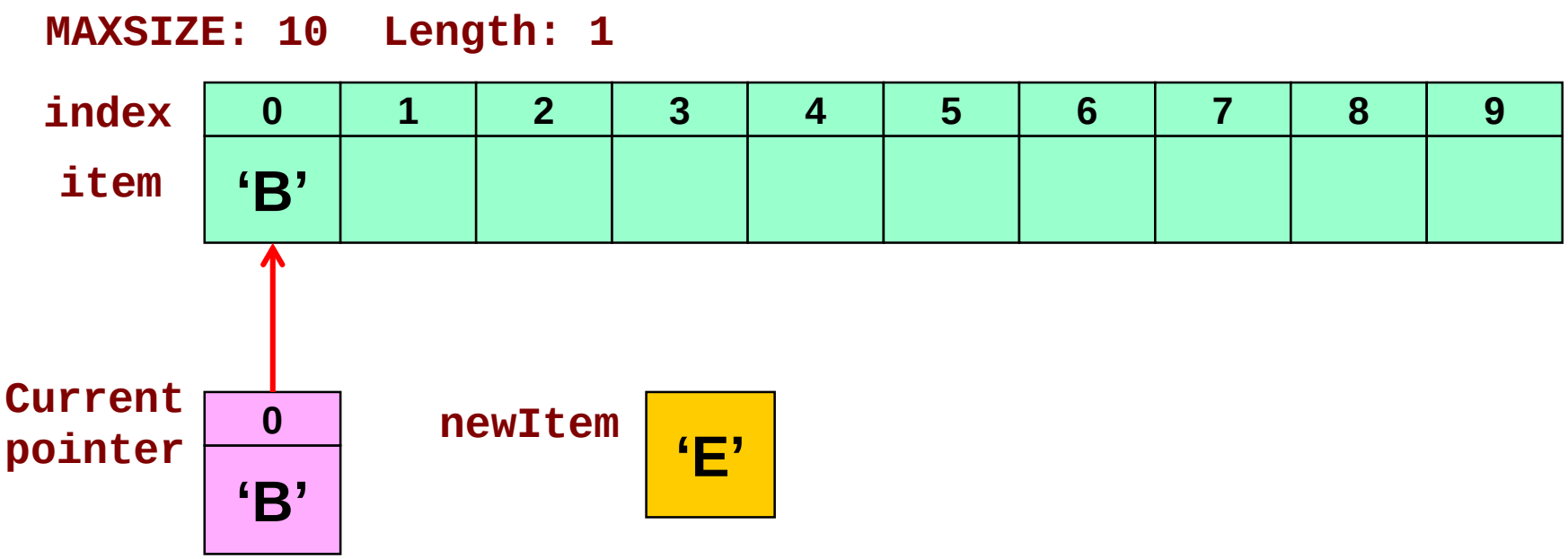
Current pointer



newItem

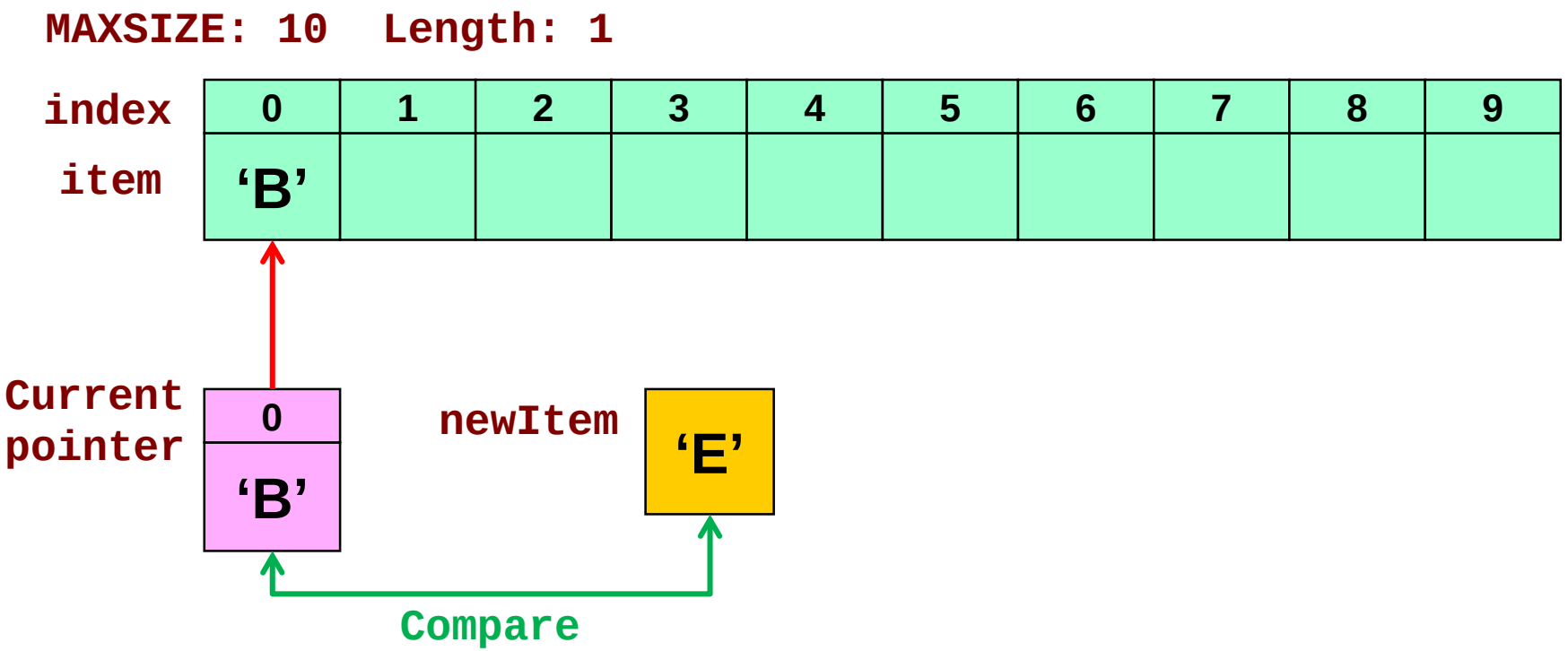
Lab02: Add (2/3)

◎ Add new item: Add to last



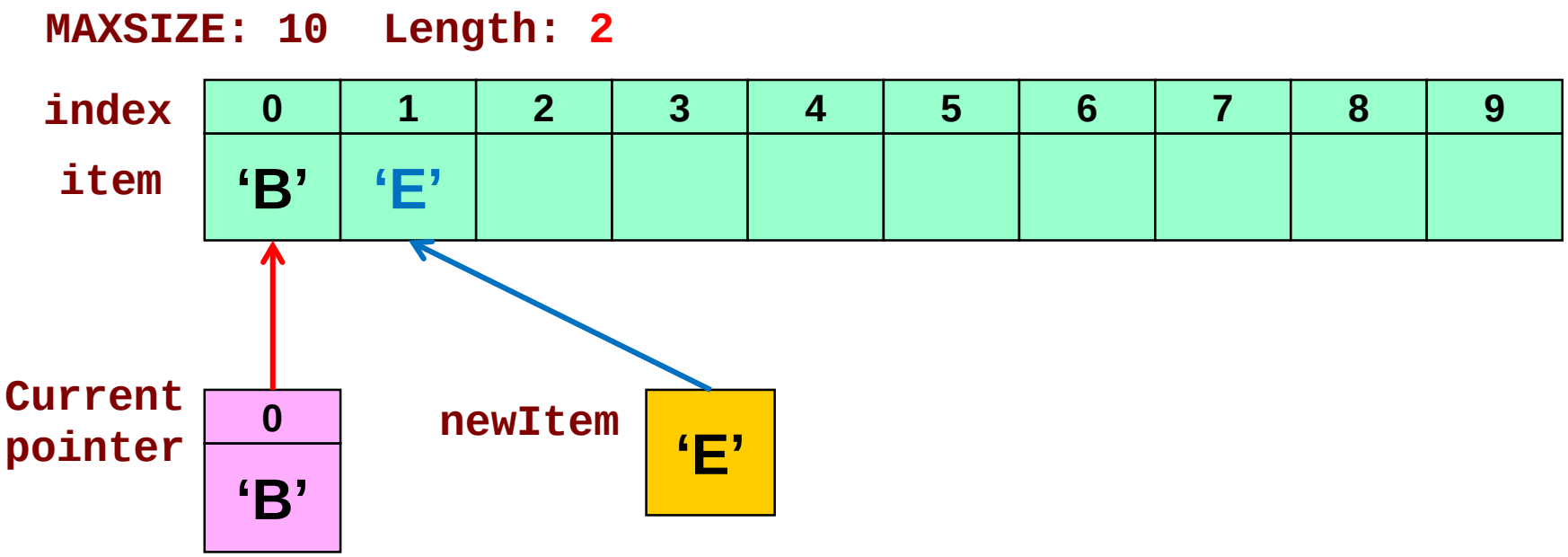
Lab02: Add (2/3)

◎ Add new item: Add to last



Lab02: Add (2/3)

◎ Add new item: Add to last



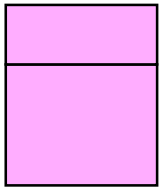
Lab02: Add (2/3)

◎ Add new item: Add to last

MAXSIZE: 10 Length: 2

index	0	1	2	3	4	5	6	7	8	9
item	'B'	'E'								

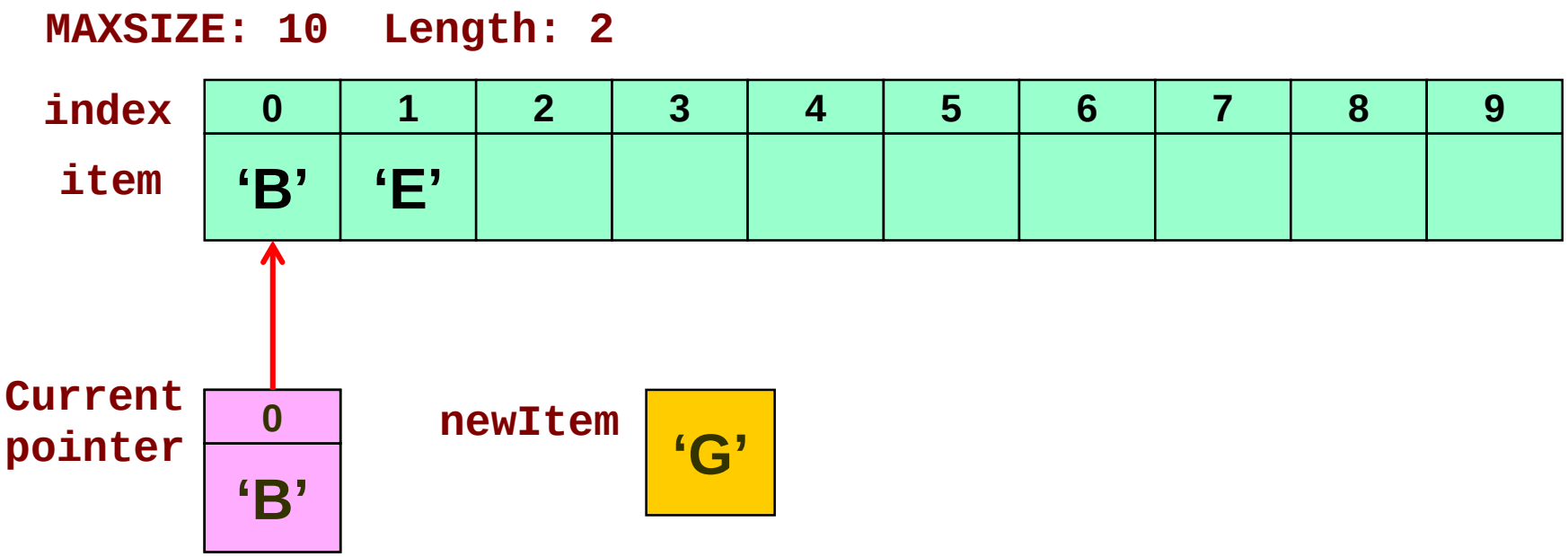
Current pointer



newItem

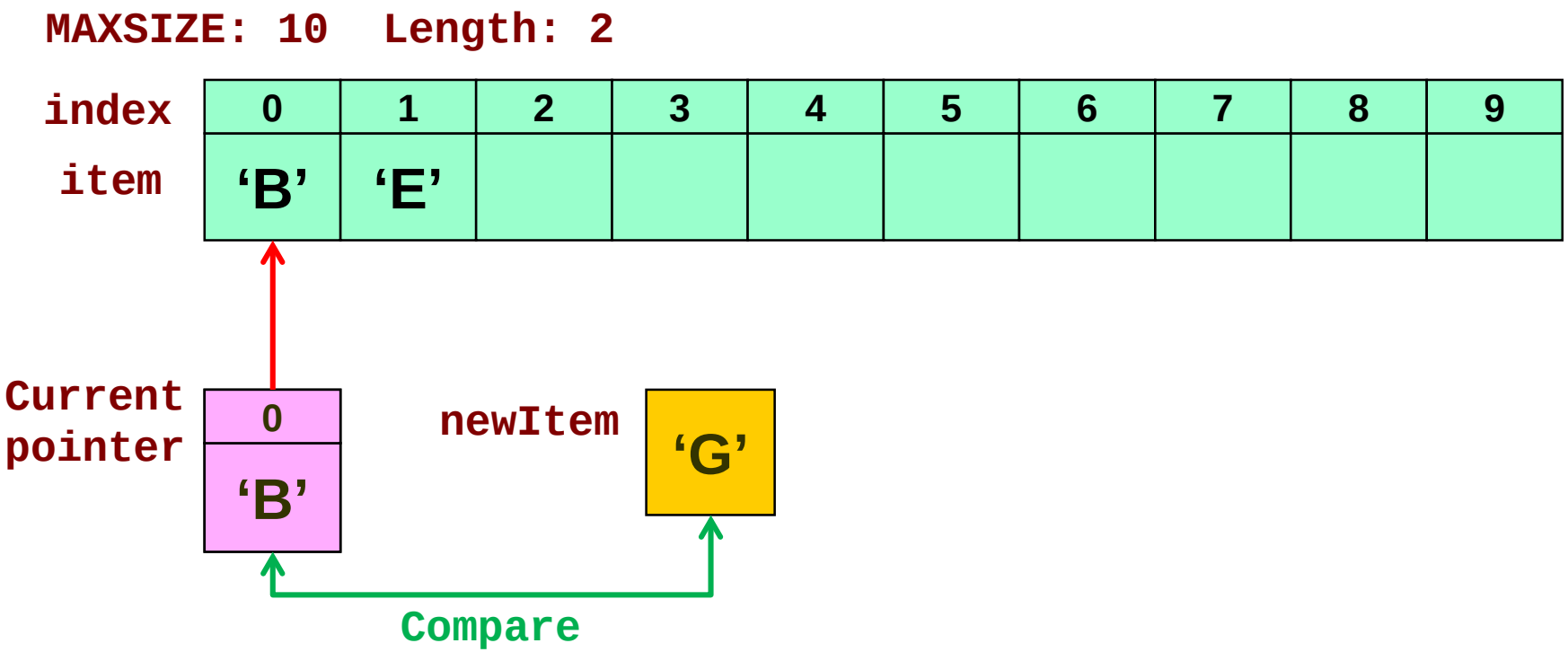
Lab02: Add (2/3)

◎ Add new item: Add to last



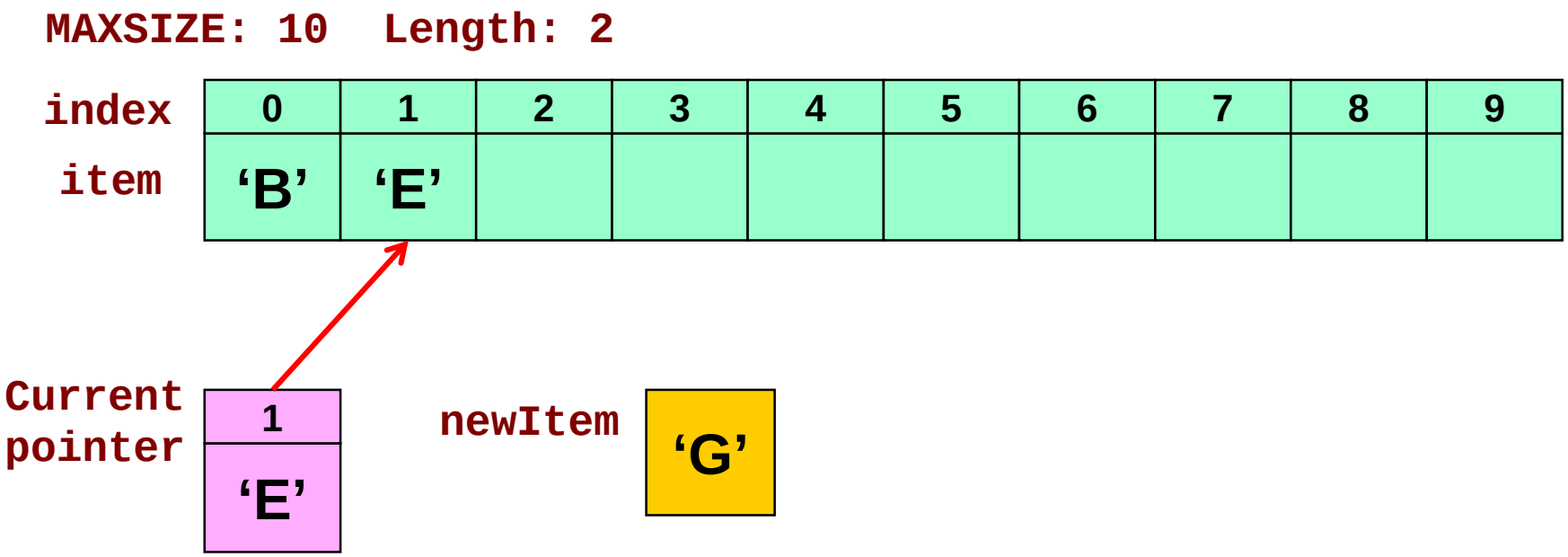
Lab02: Add (2/3)

◎ Add new item: Add to last



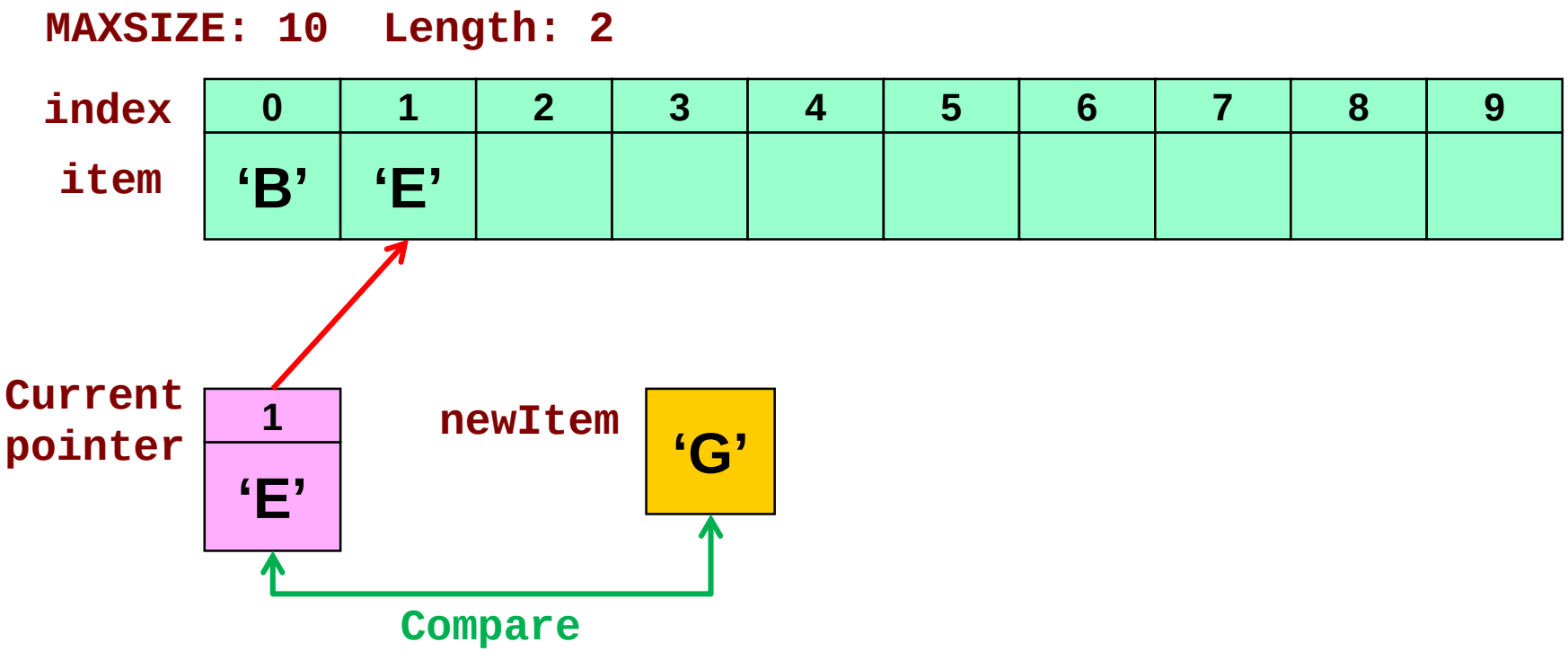
Lab02: Add (2/3)

◎ Add new item: Add to last



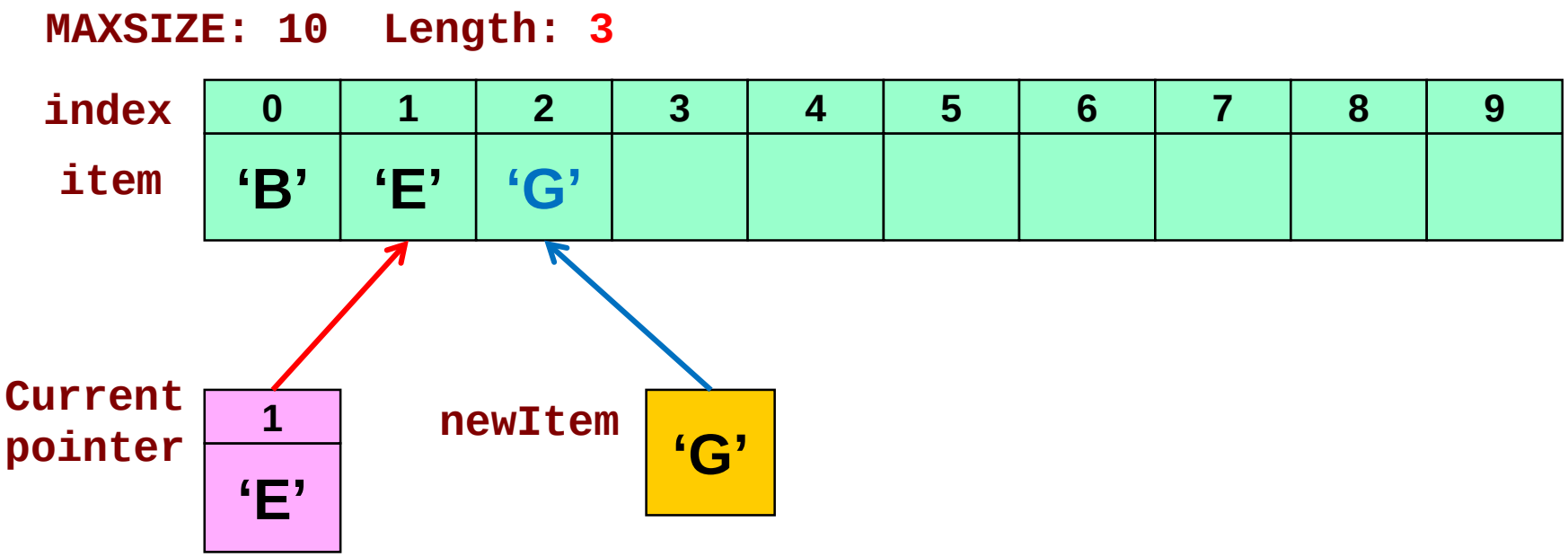
Lab02: Add (2/3)

◎ Add new item: Add to last



Lab02: Add (2/3)

◎ Add new item: Add to last



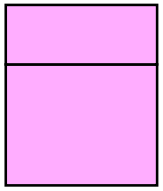
Lab02: Add (3/3)

◎ Add new item: Add to middle

MAXSIZE: 10 Length: 3

index	0	1	2	3	4	5	6	7	8	9
item	'B'	'E'	'G'							

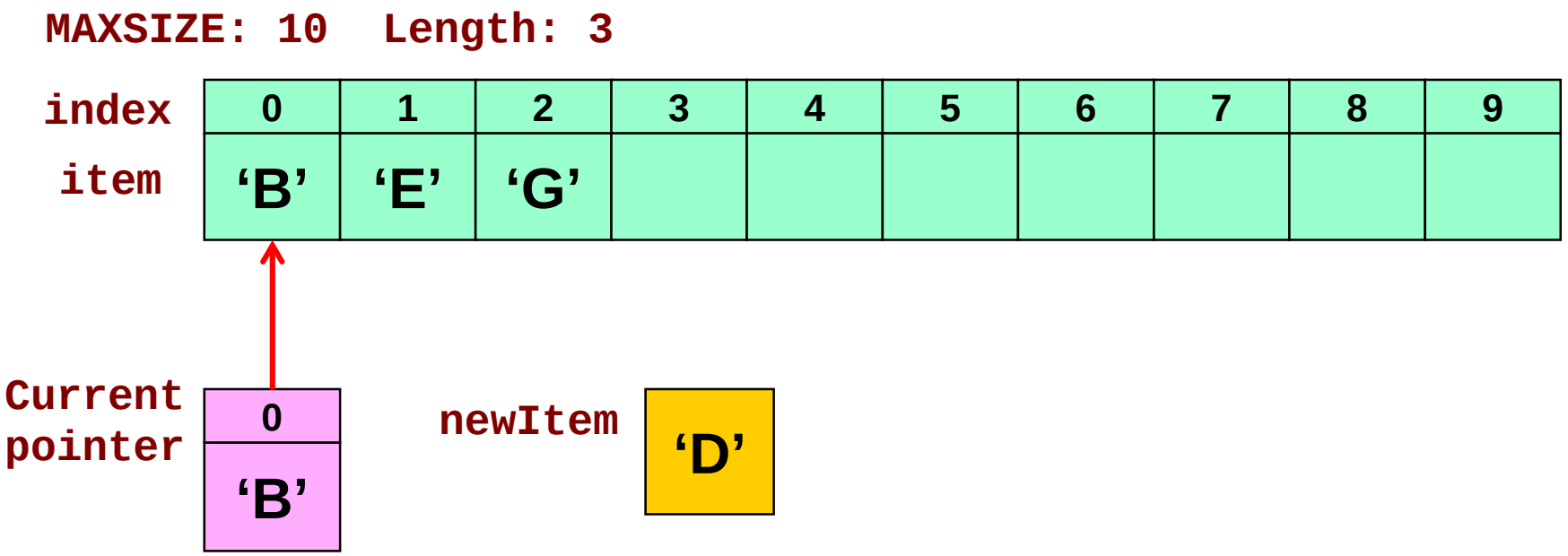
Current pointer



newItem

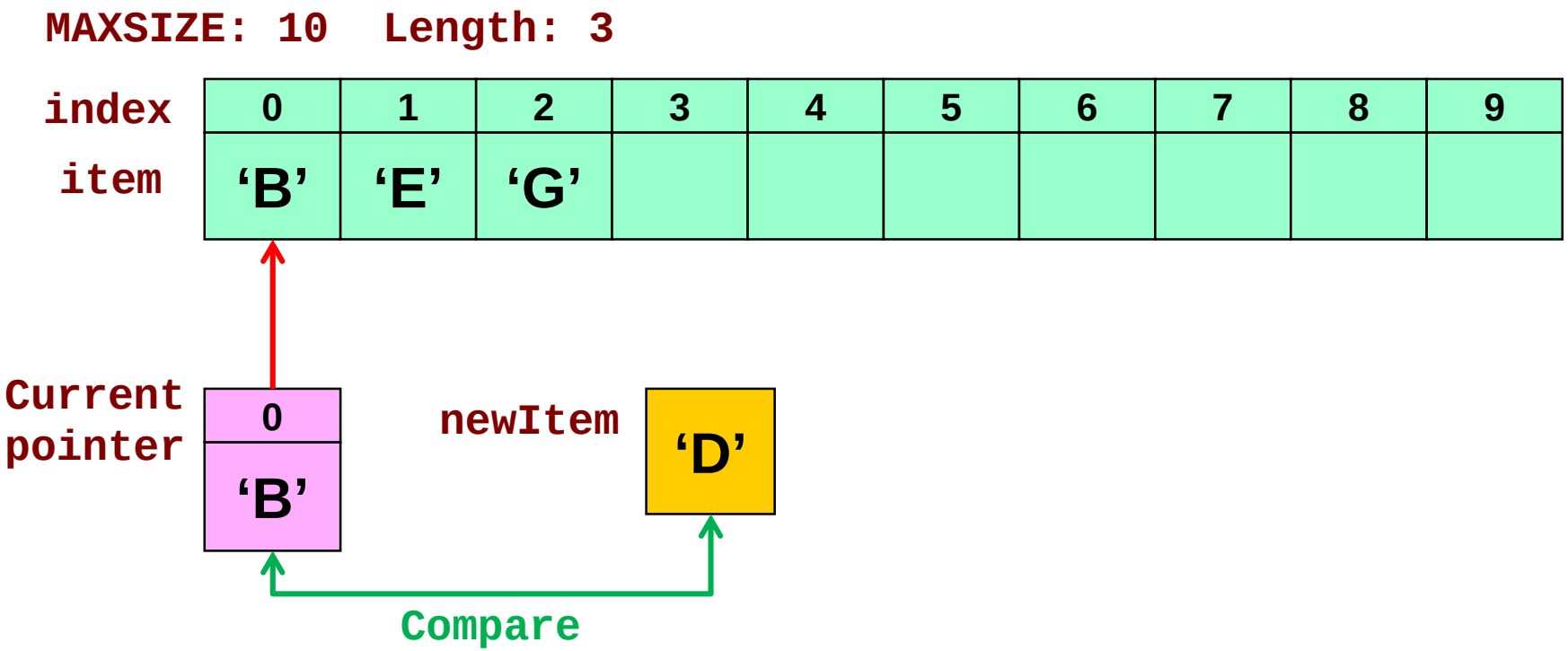
Lab02: Add (3/3)

◎ Add new item: Add to middle



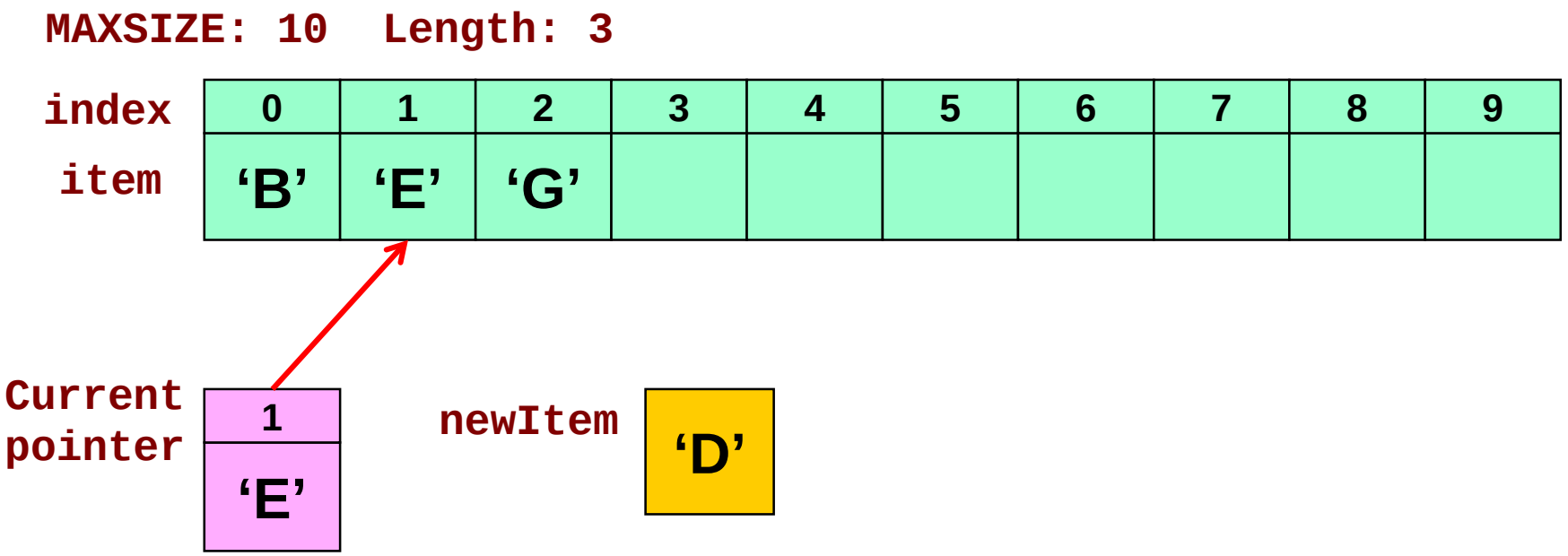
Lab02: Add (3/3)

◎ Add new item: Add to middle



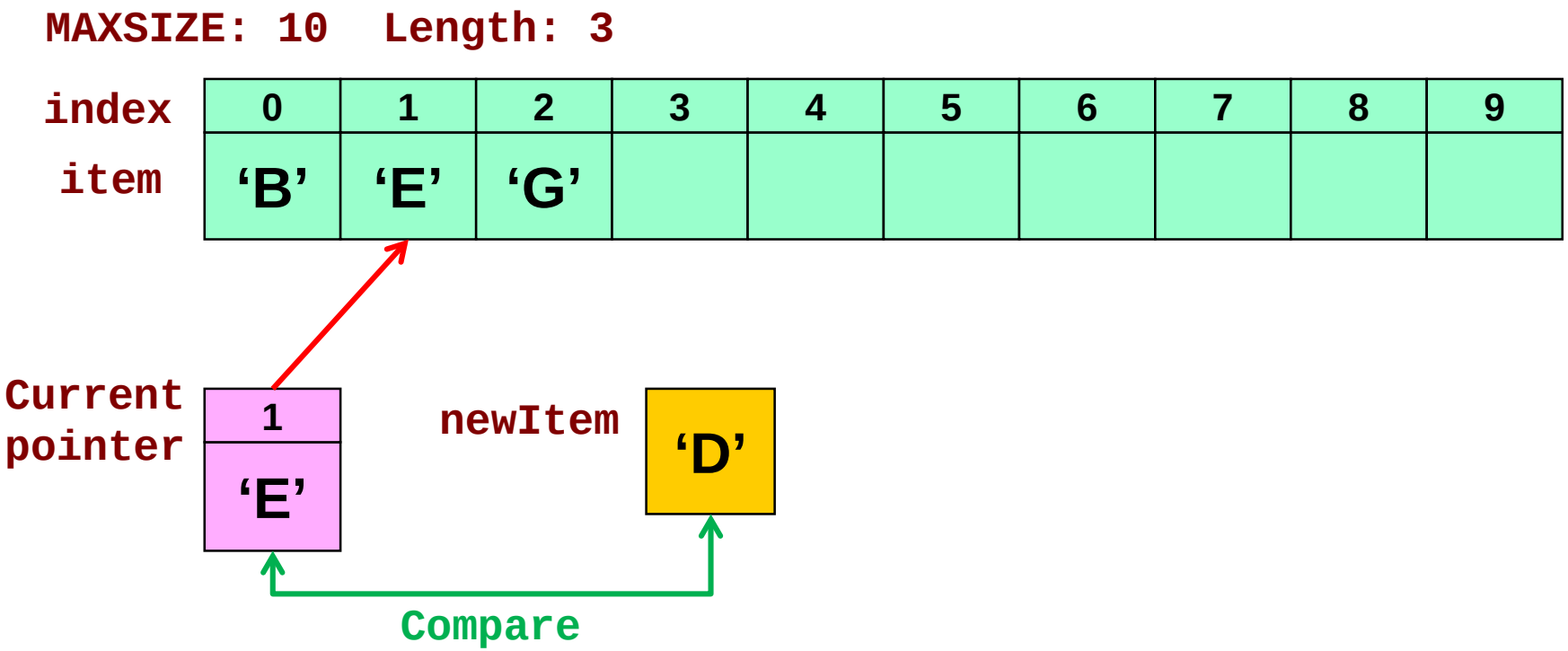
Lab02: Add (3/3)

◎ Add new item: Add to middle



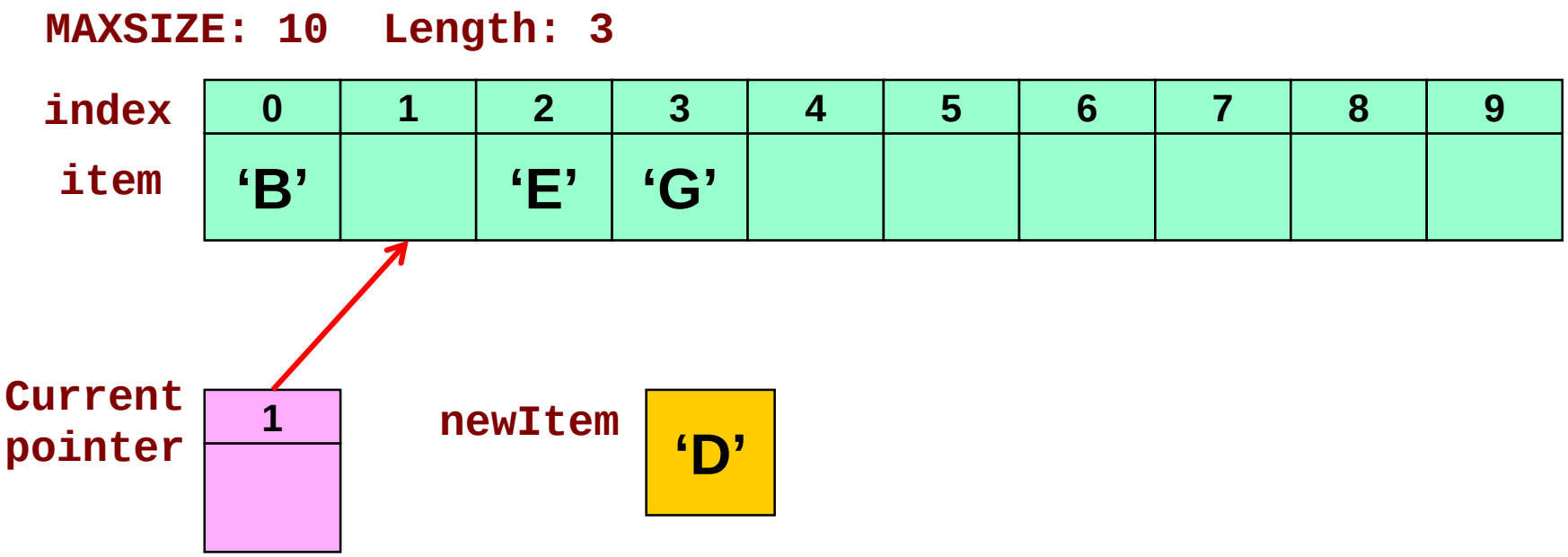
Lab02: Add (3/3)

◎ Add new item: Add to middle



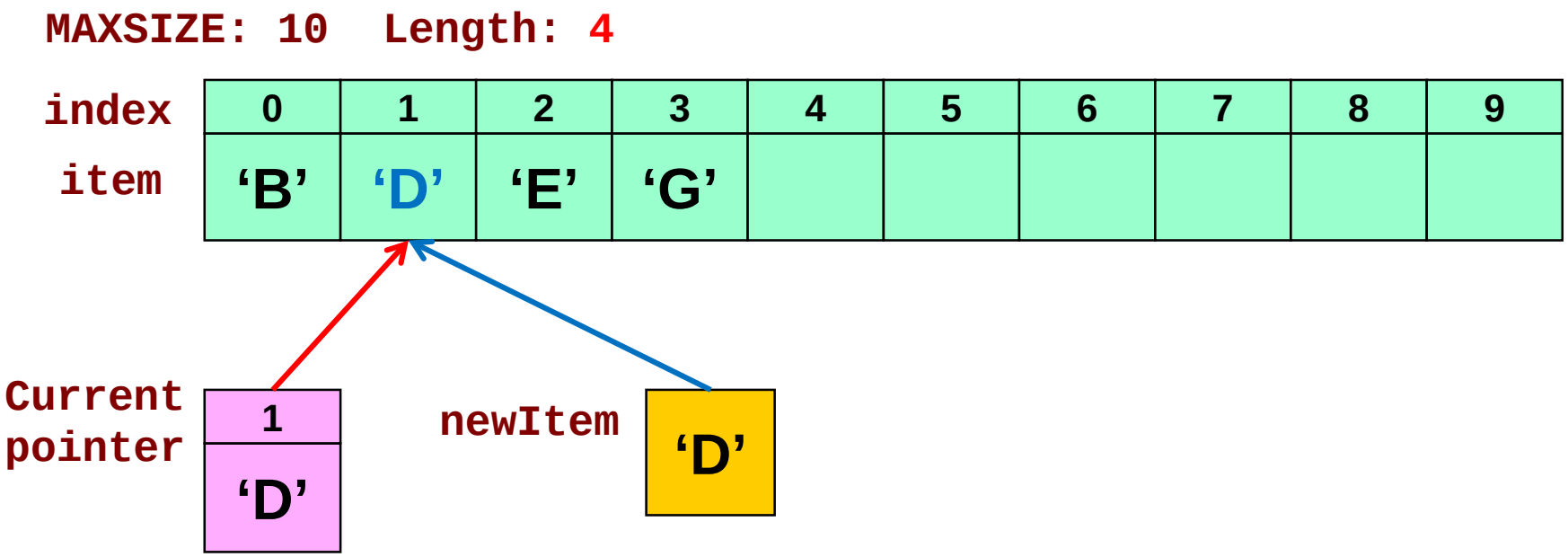
Lab02: Add (3/3)

◎ Add new item: Add to middle



Lab02: Add (3/3)

◎ Add new item: Add to middle



Lab02: Binary Search

If searching for 23 in the 10-element array: