5장. 정렬 프로그램

- 버블 정렬
- 선택 정렬
- 버블 정렬 프로그램

버블정렬

- 정렬이란
  - 임의의 순서대로 나열되어 있는 자료를 일정한 순서대로 재배열 하는 것
- 거품정렬
  - 인접한 요소간의 비교, 교환을 통해 최댓값을 맨 뒤로 보내는 것을 반복

다음과 같이 문자 'B', 'A', 'T', 'C', 'D'가 나열되어 있을 때, 알고리즘은 다음과 같다.

정렬 전	B	A	T	C	D	
정렬 과정	B	A	T	C	D	B와 A를 비교하여 교환한다.
	A	B	T	C	D	B와 T를 비교해서 교환한다.
	A	B	C	T	D	C와 T를 비교해서 교환한다.
	A	B	C	D	T	D와 T를 비교해서 교환한다.
정렬 후	A	B	C	D	T	

버블정렬 알고리즘

nSource[] : 배열
k: 데이터 수

```
    for (i = 0; i <= k - 1; i++)
    {
        for (j = 0; j < k - 1 - i; j++)
        {
            if (nSource[j] > nSource[j + 1])
            {
                temp = nSource[j];
                nSource[j] = nSource[j + 1];
                nSource[j + 1] = temp;
            }

        }
    }
```

선택정렬

- **선택정렬**
  - **가장 작은 수를 선택하여 맨 앞 자리에 위치시키기를 반복**

다음과 같이 숫자 35, 20, 90, 11, 10, 2가 나열되어 있을 때, 알고리즘은 다음과 같다.

정렬 전	35	20	90	11	10	2	
정렬과정	2	20	90	11	10	35	가장 작은 값 2를 1번째에 기억시킨다.
	2	10	90	11	20	35	2번째로 작은 값인 10을 2번째에 기억시킨다.
	2	10	11	90	20	35	3번째로 작은 값인 11을 3번째에 기억시킨다.
	2	10	11	20	90	35	4번째로 작은 값인 20을 4번째에 기억시킨다.
	2	10	11	20	35	90	5번째로 작은 값인 35를 5번째에 기억시킨다.
정렬 후	2	10	11	20	35	90	

선택정렬 알고리즘

nSource[] : 배열
k: 데이터 수

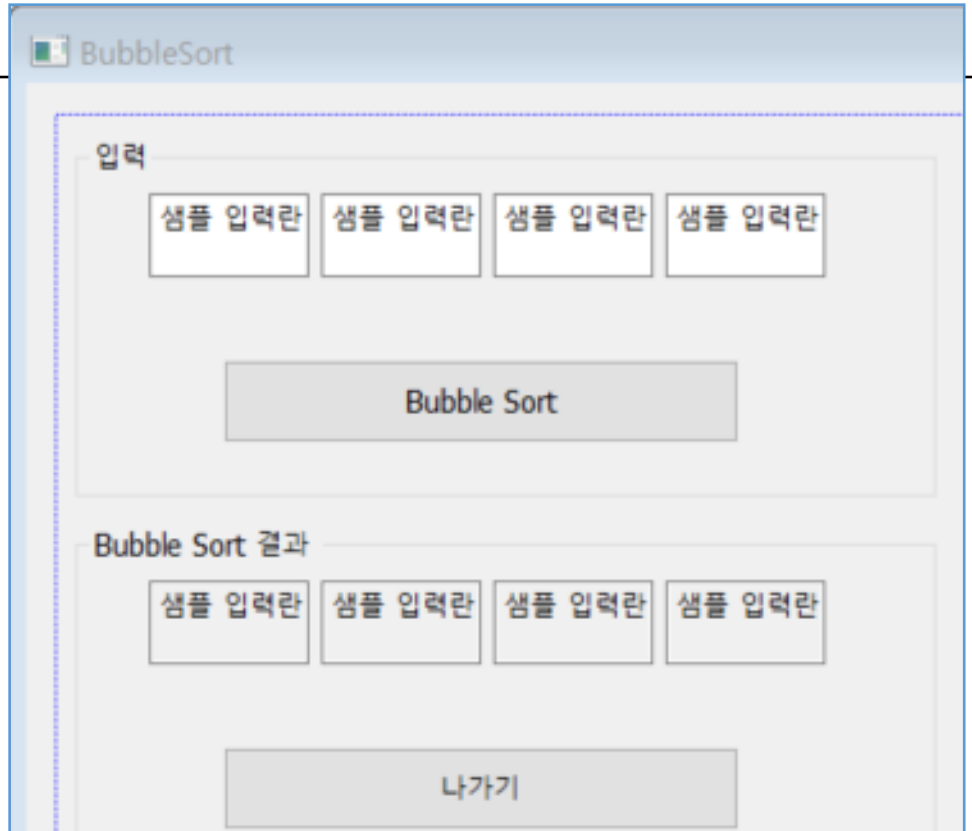
```
int n = 4;
int min; //최솟값
int minIndex; //최솟값 index
for (i = 0; i<n - 1; i++) {
        minIndex = i; min = nSource[i]; //최솟값 초기설정
        for (j = i + 1; j<n; j++) { //i 이후로 계속 최솟값 찾음
                if (min > nSource[j]) { //더 작은 값이면 최소값으로 기억
                        min = nSource[j]; minIndex = j;
                }
        }
        nSource[minIndex] = nSource[i];
        nSource[i] = min;
}
```

minindex번째 값과
i번째 값을 뒤바꿈

정렬 프로그램
(BubbleSort)

(1) 새 프로젝트 생성
("파일"–"새로만들기"–"프로젝트"
-> "Visual C++" – "MFC" - "MFC 응용프로그램")
-> 이름: BubbleSort
-> 응용 프로그램 종류: "대화상자기반" - "마침"

(2) 대화상자 편집



정렬 프로그램
(BubbleSort)

(3) 8개의 editbox에 각각 int 타입의 "Value"변수 연결.
m_n1, m_n2, m_n3, m_n4, m_n5, m_n6, m_n7, m_n8

(4) "나가기" 버튼 핸들러 구현

```
void CBubbleSortDlg::OnClickedButtonExit()
{
    OnOK();
}
```

정렬 프로그램 (BubbleSort)

(5) "Bubblesort" 버튼 핸들러 구현

```
void CBubbleSortDlg::OnClickedButtonSort() {
    UpdateData(TRUE);
    CClientDC dc(this);

    static int nSource[4];
    int i, j, temp, k = 4;

    nSource[0]=m_n1; nSource[1]=m_n2; nSource[2]=m_n3; nSource[3]=m_n4;

    for (i = 0; i <= k - 1; i++) {
        for (j = 0; j < k - 1 - i; j++) {
            if (nSource[j] > nSource[j + 1]) {
                temp = nSource[j];
                nSource[j] = nSource[j + 1];
                nSource[j + 1] = temp;
            }
        }
    }
    m_n5=nSource [0]; m_n6=nSource [1]; m_n7=nSource [2]; m_n8=nSource [3];
    UpdateData(FALSE);
}
```