

## 과제 9-1. 배열의 입력과 변환

- 입력할 배열의 크기를 입력 받아 이에 필요한 정수형 메모리를 할당하고, 필요한 정수들을 입력하고, 입력된 배열의 주소값을 return 하는 입력 함수를 만든다.
  - 함수 원형: *int \* input\_integer (int \*arrSize);*
- 주어진 숫자 만큼의 배열의 순서를 뒤집어 저장하는 역 변환 함수를 작성한다.
  - 함수 원형: *void inverse\_reorder(int \*arr, int arrSize);*
  - 예를 들어 배열의 이름이 arr 이고, 주어진 크기가 10 이라고 할 때, 변경 전 arr[9]에 저장된 값은 arr[0]에, arr[8]에 저장되었던 값은 arr[1]에 저장된다.
- 주어진 정수 배열을 라인당 지정된 숫자 만큼 출력하는 출력 함수를 만든다.
  - 함수 원형 *void print\_array (int \*, int arrSize, int lineNo);*
- 메인 프로그램에서는
  - 정수 배열을 입력 받을 정수형 포인터 변수를 선언하고, 해당되는 배열에 정수의 숫자들을 입력하는 *input\_integer* 함수를 호출하고,
  - 입력된 배열을 *print\_array* 함수를 이용하여 출력하고,
  - 함수 순서를 바꾸는 함수 *inverse\_reorder*를 호출한 후 변경 후의 배열에 저장된 값을 순서대로 출력한다.
  - 배열에 저장된 값에 대한 접근 및 변경은 포인터를 이용한 함수를 작성하여 수행한다 .

# 과제 9-1. 배열의 입력과 변환

라인당 출력 숫자를 5로 정하고 실행한 결과 임

```

입력할 정수배열의 크기 = ? 10
10 20 30 40
50 60 70 80
90 100

입력한 정수 배열data :
10 20 30 40 50
60 70 80 90 100

역으로 변환한 정수 배열 data :
100 90 80 70 60
50 40 30 20 10

-----
Process exited after 18.02 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
    
```

```

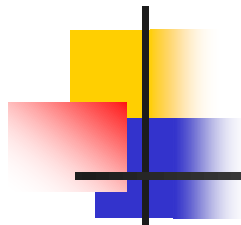
입력할 정수배열의 크기 = ? -10
입력할 정수배열의 크기 = ? 0
입력할 정수배열의 크기 = ? 5
10 20
30 40
50

입력한 정수 배열data :
10 20 30 40 50

역으로 변환한 정수 배열 data :
50 40 30 20 10

-----
Process exited after 14.1 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
    
```

배열의 크기를 잘못 지정하면 다시 물어봄

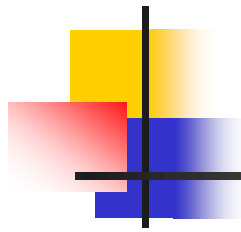


## 과제 9-2. 주어진 행렬에서 특정 row의 sorting

- main함수에서 행의 개수가 4, 열의 개수가 8인 2차원 integer 배열을 선언하고 다음과 같이 초기화 한다.

```
int data[4][8] = { {838, 758, 113, 515, 51, 627, 10, 419},  
                  {212, 86, 749, 767, 84, 60, 225, 543},  
                  {89, 183, 137, 566, 966, 978, 495, 311},  
                  {367, 54, 31, 145, 882, 736, 524, 505} };
```

- 사용자로부터 원하는 행 번호를 입력 받고, 입력 받은 행을 오름차순으로 정렬한 뒤 해당되는 행을 출력하는 프로그램을 작성한다.
- 행 번호를 잘못 입력하면 다시 입력을 받고, -1을 입력하면 전체 배열을 출력한다.
- 행의 정렬은 배열에서 배운 selection sorting 함수를 이용한다.  
함수 원형 : *void selectionSort (int \* arr, int arrSize);*
  - 배열access는 index 표시를 이용하지 하지 말고 포인터 변수를 이용하여 수정할 것



# 과제 9-2. 주어진 행렬에서 특정 row의 sorting

■ Example)

```
Before sorting
838 758 113 515 51 627 10 419
212 86 749 767 84 60 225 543
89 183 137 566 966 978 495 311
367 54 31 145 882 736 524 505

Enter the row index (0~3) : 0
0_th row after sorting
10 51 113 419 515 627 758 838

Enter the row index (0~3) : 1
1_th row after sorting
60 84 86 212 225 543 749 767

Enter the row index (0~3) : 2
2_th row after sorting
89 137 183 311 495 566 966 978

Enter the row index (0~3) : 3
3_th row after sorting
31 54 145 367 505 524 736 882

Enter the row index (0~3) : -1
After sorting
10 51 113 419 515 627 758 838
60 84 86 212 225 543 749 767
89 137 183 311 495 566 966 978
31 54 145 367 505 524 736 882
아래면 아무 키나 누르십시오 . . .
```

## 과제 9-3. 행렬의 생성과 transpose(전치)

수학, 특히 선형대수학에서 어떤 행렬의 전치행렬(Transposed Matrix)은 원래 행렬의 열은 행으로, 행은 열로 바꾼 것이다. 따라서  $M \times N$  matrix의 전치행렬은  $N \times M$  matrix가 된다.

- 주어진 행렬의 전치행렬을 반환하는 함수를 작성하시오.

함수 원형 : *int\*\* transpose (int \*\*matrix, int m, int n);*

- 함수에서 필요한 공간의 메모리를 할당 받아서 전치시키고,
- 전치된 행렬의 결과를 가지는 pointer를 반환한다.

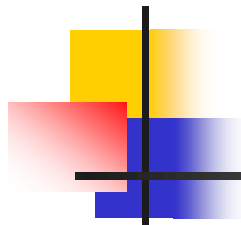
- 주어진 행렬을 출력하는 함수를 작성하시오

함수 원형 : *void printMatrix (int \*\*matrix, int m, int n);*

- Main 함수에서는 row 크기와 column 크기를 입력 받아서,

- 해당되는 크기의 matrix가 저장 가능한 공간의 메모리를 할당 받고,
- 행렬의 원소는 rand() 함수를 통하여 1~100 을 생성하며, seed번호는 학생의 학번으로 한다.
- 원래 행렬과 전치행렬을 차례대로 출력한다.

- 단, 전역 변수의 사용을 금지한다.



# 과제 9-3. 행렬의 생성과 transpose(전치)

- 아래와 같은 결과가 나오도록 한다.

*Example*

```
Number of Rows : 3
Number of Cols : 4
seed번호 20119999로 생성된 Matrix
86 86 55 25
82 63 47 23
40 12 97 81
Transpose된 Matrix
86 82 40
86 63 12
55 47 97
25 23 81
계속하려면 아무 키나 누르십시오 . . .
```

```
Number of Rows : 6
Number of Cols : 4
seed번호 20119999로 생성된 Matrix
86 86 55 25
82 63 47 23
40 12 97 81
27 79 75 19
75 43 67 59
86 46 61 56
Transpose된 Matrix
86 82 40 27 75 86
86 63 12 79 43 46
55 47 97 75 67 61
25 23 81 19 59 56
계속하려면 아무 키나 누르십시오 . . .
```