

1. InsertionSort

```

void insert(element e, element a[], int i)
/* insert e into the ordered list a[i:i] such that the
resulting list a[i:i+1] is also ordered, the array a
must have space allocated for at least i+2 elements */
a[i+1] = e;
while (e.key < a[i].key)
{
    a[i+1] = a[i];
    i--;
}
a[i+1] = e;

```

Program 7.4: Insertion into a sorted list.

다음 입력 리스트에 대해 insertionSort(Program 7.5)의 for문에서 insert() 함수 실행 이 후의 배열상태를 순서대로 기술하라.(2점) **한 번 수행한 자료에 변화가 생김**
 입력 리스트 (12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18)

```

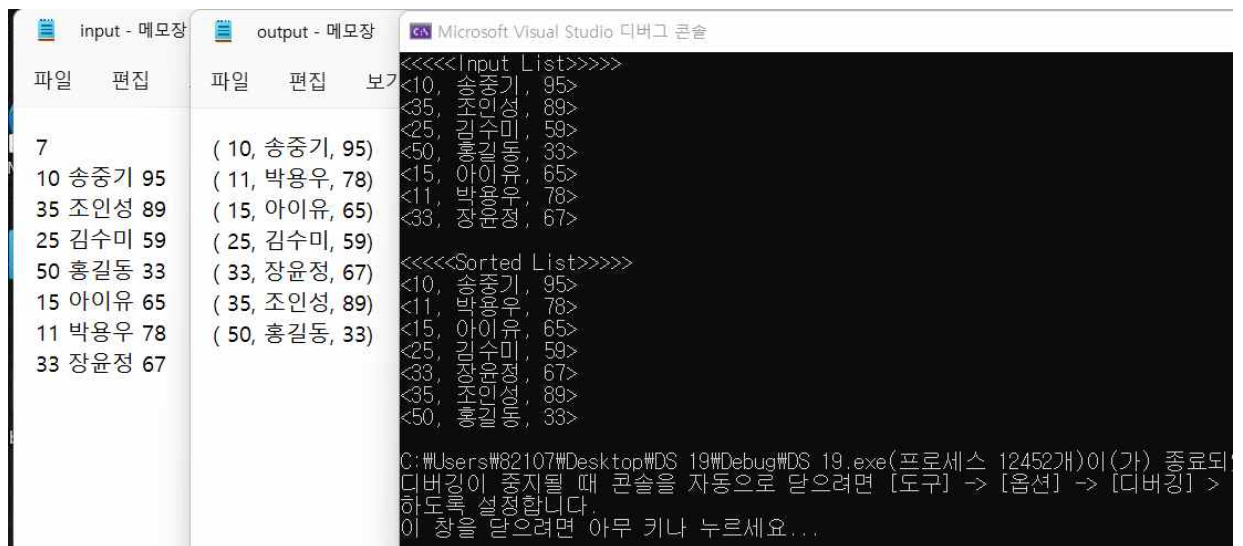
void insertionSort(element a[], int n)
/* sort a[1:n] into nondecreasing order */
int j;
for (j = 2; j <= n; j++) {
    element temp = a[j];
    insert(temp, a, j-1);
}

```

Program 7.5: Insertion sort

* 연습장에 적은 후 사진을 찍어도 되며 그 결과를 보고서에 넣을 것

	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	12	2	16	30	8	28	4	10	20	6	18	
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	2	2	12	16	30	8	28	4	10	20	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	16	2	12	16	30	8	28	4	10	20	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	30	2	12	16	30	8	28	4	10	20	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	8	2	8	12	16	30	28	4	10	20	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	28	2	8	12	16	28	30	4	10	20	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	4	2	4	8	12	16	28	30	10	20	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	10	2	4	8	10	12	16	28	30	20	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	20	2	4	8	10	12	16	20	28	30	6	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	6	2	4	6	8	10	12	16	20	28	30	18
→	Index	0	1	2	3	4	5	6	7	8	9	10	11
	값	18	2	4	6	8	10	12	16	18	20	28	30

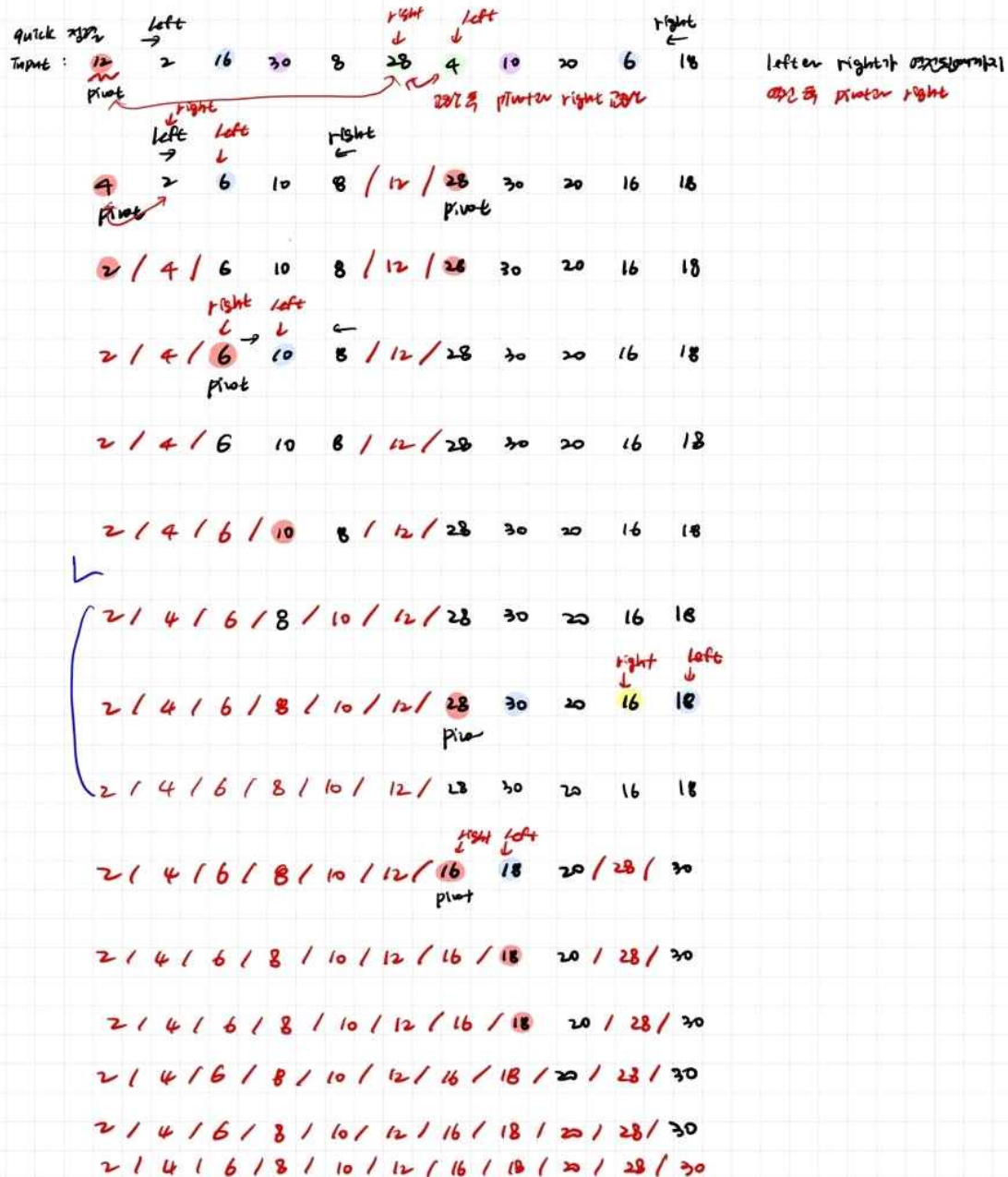


입력 리스트 (12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18) *정렬 후* *2* *제거*

R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	left	right
[26	5	37	1	61	11	59	15	48	19]	1	10
[11	5	19	1	15]	26	[59	61	48	37]	1	5
[1	5]	11	[19	15]	26	[59	61	48	37]	1	2
1	5	11	[19	15]	26	[59	61	48	37]	4	5
1	5	11	15	19	26	[59	61	48	37]	7	10
1	5	11	15	19	26	[48	37]	59	[61]	7	8
1	5	11	15	19	26	37	48	59	[61]	10	10
1	5	11	15]	19	26	37	48	59	61		

Figure 7.1: Quick sort example

* 연습장에 적은 후 사진을 찍어도 되며 그 결과를 보고서에 넣을 것



input - 메모장

파일

편집

보기

11
12 2 16 30 8 28 4 10 20 6 18

output - 메모장

파일

편집

보기

11
2 4 6 8 10 12 16 18 20 28 30

선택 Microsoft Visual Studio 디버그 콘솔

```

<<<<<Input List>>>>>
12 2 16 30 8 28 4 10 20 6 18

executive of quick sort...
[ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9] [10] [11] [left right]
12 2 16 30 8 28 4 10 20 6 18: 1 11
4 2 6 10 8 12 28 30 20 16 18: 1 5
2 4 6 10 8 12 28 30 20 16 18: 1 1
2 4 6 10 8 12 28 30 20 16 18: 3 5
2 4 6 10 8 12 28 30 20 16 18: 3 2
2 4 6 10 8 12 28 30 20 16 18: 4 5
2 4 6 8 10 12 28 30 20 16 18: 4 4
2 4 6 8 10 12 28 30 20 16 18: 6 5
2 4 6 8 10 12 28 30 20 16 18: 7 11
2 4 6 8 10 12 16 18 20 28 30: 7 9
2 4 6 8 10 12 16 18 20 28 30: 7 6
2 4 6 8 10 12 16 18 20 28 30: 8 9
2 4 6 8 10 12 16 18 20 28 30: 8 7
2 4 6 8 10 12 16 18 20 28 30: 9 9
2 4 6 8 10 12 16 18 20 28 30: 11 11

calls of quick sort: 15
C:\Users\82107\Desktop\WDS 19\Debug\02.exe(프로세스 8364개)이(가) 종료되었습니다(코드: 0개)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지될 때 콘솔을 닫습니다.]

```