// 선택정렬 프로그램

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#define MAX\_SIZE 10

#define SWAP(x, y, t) ( (t)=(x), (x)=(y), (y)=(t) )

int list[MAX\_SIZE];

int n;

void selection\_sort(int list[], int n)

{

int i, j, least, temp;

for (i = 0; i < n - 1; i++) {

least = i;

for (j = i + 1; j < n; j++) // 최소값 탐색

if (list[j] < list[least]) least = j;

SWAP(list[i], list[least], temp);

}

}

int main(void)

{

int i;

n = MAX\_SIZE;

srand(time(NULL));

for (i = 0; i < n; i++) // 난수 생성 및 출력

list[i] = rand() % 100; // 난수 발생 범위 0~99

selection\_sort(list, n); // 선택정렬 호출(void selection\_sort)

for (i = 0; i < n; i++) printf("%d ", list[i]);

printf("\n");

return 0;

}

// 삽입정렬 프로그램

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#define MAX\_SIZE 10

int list[MAX\_SIZE];

int n;

// 삽입정렬 프로그램 -> 가장 작은 수를 찾고 그 다음 행과 비교?

void insertion\_sort(int list[], int n)

{

int i, j, key;

for (i = 1; i < n; i++) {

key = list[i];

for (j = i - 1; j >= 0 && list[j] > key; j--)

list[j + 1] = list[j]; /\* 레코드의 오른쪽 이동 \*/

list[j + 1] = key;

}

}

int main(void)

{

int i;

n = MAX\_SIZE;

srand(time(NULL));

for (i = 0; i < n; i++) // 난수 생성 및 출력

list[i] = rand() % 100; // 난수 발생 범위 0~99

insertion\_sort(list, n); // 삽입정렬 호출

for (i = 0; i < n; i++)

printf("%d ", list[i]);

printf("\n");

return 0;

}

// 버블정렬 프로그램

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#define MAX\_SIZE 10

int list[MAX\_SIZE];

int n;

#define SWAP(x, y, t) ( (t)=(x), (x)=(y), (y)=(t) )

void bubble\_sort(int list[], int n)

{

int i, j, temp;

for (i = n - 1; i > 0; i--) {

for (j = 0; j < i; j++)

/\* 앞뒤의 레코드를 비교한 후 교체 \*/

if (list[j] > list[j + 1]) SWAP(list[j], list[j + 1], temp);

}

}

int main(void)

{

int i;

n = MAX\_SIZE;

srand(time(NULL));

for (i = 0; i < n; i++) // 난수 생성 및 출력

list[i] = rand() % 100; // 난수 발생 범위 0~99

bubble\_sort(list, n); // 삽입정렬 호출

for (i = 0; i < n; i++) printf("%d ", list[i]);

printf("\n");

return 0;

}