Jeon, sihyeon (전시현) 202135574

# 과제: HW-1

* Code Test: https://www.acmicpc.net/problem/2750
* 해당 프로그램에 주석 (comment)를 한글로 단다. (추가할 수 있는 것은 최대한 추가)

# #HW 1-1

|  |  |
| --- | --- |
|  | Solving the problem using insertion sort and uploading a report using  submission template |

**Requirements**

* < 요구조건 #1: insertion sort>

## SOURCE CODE

|  |
| --- |
| /\*\*\*\*      \*file : insertionsort.c      \*      \* Author : Jeon Sihyeon (wjstlgus13@gachon.ac.kr)      \*      \* Date : 09/07/2022      \*      \* Partner : I worked alone      \* Course : Algorithm (14465\_001)      \*      \* Summary of File:      \*   This file contains code which rearrange numbers in ascending order with insertion sort.      \*  \*/  #include <stdio.h>  #include <stdlib.h>  void getnumbers(int arr[], int size);  void insertionSort(int arr[], int size);    //함수 선언  int main(){      int numofnumbers;      scanf("%d", &numofnumbers);             //num 어레이의 개수      int \*num = (int \*)malloc(numofnumbers\*sizeof(int)); //동적 할당      getnumbers(num, numofnumbers);      //calls getnumbers function      insertionSort(num, numofnumbers);   //calls insertionSort function      for (int i = 0; i<numofnumbers; i++){          printf("%d\n", num[i]);          //답 출력      }        free(num);      //동적 할당 해제  }  /\*\*   \* void getnumbers(int arr[], int size)   \*   \* getnumbers :   \*      arr에 size만큼의 숫자를 받아 저장한다.   \*   \* parameters : int 어레이, integer 형   \*   \* return value : 없음  NOTE : Modifies the array "in place"   \*   \*/  void getnumbers(int arr[], int size){      for (int i = 0; i<size; i++){          scanf("%d", &arr[i]);      }  }  /\*\*   \* void insertionSort(int arr[], int size)   \*   \* insertionSort 함수 :   \*      insertion sort 알고리즘을 이용하여 정렬한다   \*   \* parameter : int array, integer 형   \*   \* return value : 없음      Note : modifies the array "in place"   \*   \* Description :   \*   \*      이 함수는 insertion sort 알고리즘을 활용하였다.   \*   \*/  void insertionSort(int arr[], int size){      int index = 0;      int key = arr[index];       //initializes key with the value of arr[index]      index++;      while(index<size){  //index = size일 때 while문 escape.          key = arr[index];          for (int i = index-1; i >= 0; i--){              arr[i+1] = arr[i];      //arr[i+1]에 arr[i] 값을 저장              if(arr[i] < key){       //key 값과 비교하여 key의 위치 선정                  arr[i+1] = key;                  break;              }              if(i==0){                  arr[i] = key;       //arr[i]에 key 값을 저장              }          }          index++;      }  } |

## OUTPUT (Screen Shots)

|  |  |
| --- | --- |
| #1 | [제출 결과] |
| (유의사항:제출 결과에 대한 분석 및 간단 설명) | |

//(본인이 실행한 화면을 출력)  
//(필요한 만큼 더 생성하여 실행 결과를 잘 파악할 수 있도록 제시)   
// (Screen shot이 없는 경우 생략 가능)

// 제출하는 문제 수에 맞게 위에 양식을 복사하여 사용

텍스트이(가) 표시된 사진

자동 생성된 설명

Array = {2, 8, 5, 3, 9, 4}일 때 {2, 3, 4, 5, 8, 9}로 정렬되었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Array = {3, 8, 2, 6, 2, 1}로 2가 2번 중복되는 경우에도 {1, 2, 2, 3, 6, 8}로 정렬된 것을 볼 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Array = {8, 5, 2, 8, 2, 1}로 8이 2번, 2가 2번 중복되는 경우에도 {1, 2, 2, 5, 8, 8}로 잘 정렬된 것을 볼 수 있다.