Exercise

Basic concepts (1)

Exercise 1

- Selection sort 를 다음과 같은 구성으로 구현하라

1. 전역변수 int list[MAX_SIZE] 사용
2. 파일을 열어 파일에 있는 정수들을 읽고 순서대로 list에 저장
   - 파일명은 argument로 전달
3. void selection_sort(): list 에 있는 수들을 오름차순으로 정렬
4. 정렬된 결과 표준 출력

Input:
- Text파일로 된 정수 리스트, 각 정수는 공백으로 구분
  ex) 5 9 10 2 4

Output:
- 정렬된 리스트를 표준출력, 각 정수는 공백으로 구분
  ex) 2 4 5 9 10

Exercise 2

- Binary search 를 다음과 같은 구성으로 구현하라

1. Exercise 1 에서 작성한 코드를 재사용하여 selection sort 수행
2. int binsearch(int searchnum, int left, int right): list 에서 searchnum 을 찾아 있으면 해당 인덱스를, 없으면, -1 을 반환
3. 찾을 정수 searchnum은 txt 파일에 정수 리스트와 함께 전달
4. Searchnum에 대한 binsearch 함수를 수행하고 결과를 표준출력

Input:
– 두 줄로 구성된 txt 파일
• 첫 번째 줄 : 찾을 정수 값
• 두 번째 줄 : 정수 리스트
Ex) 5
4 7 1 3 5

Output:
– 찾는 값이 있으면 인덱스를, 없으면 -1 을 표준출력
Ex) 3

Exercise 3(option)

- Binary search algorithm 을 다음과 같은 구성으로 재구현하라
  1. int read_data(char infile[], int list[]): infile 을 열어 파일에 있는 정수 리스트를 list 배열에 순서대로 저장하고 리스트의 크기를 반환
  2. selection_sort를 호출하여 배열을 오름차순으로 정렬한다
     - 이때, 배열의 크기는 read_data 에서 반환된 정수값을 사용
  3. void write_data(int list[], char outfile[], int size): list 에 있는 수들을 outfile 에 쓴다
  4. int binsearch(int list[], int searchnum, int right, int left): list 에서 searchnum 을 찾아 있으면 해당 인덱스를, 없으면, -1 을 반환한다.

  Input:
  – 두 줄로 구성된 txt 파일 (첫 번째 줄 : 찾을 정수 값, 두 번째 줄 : 정수 리스트)
    Ex) 5
    4 7 1 3 5

  Output:
  – 정렬된 리스트가 저장된 파일(두 번째 argument로 저장할 파일의 이름 전달)
  – 찾는 값이 있으면 인덱스를, 없으면 -1 을 표준출력
    Ex) 3