Homework 1C report

2014313303 홍태하(개발환경 : Linux)

1. 알고리즘

인풋이 네 자리 16진수로 들어오기 때문에 먼저 인풋들을 십진수로 바꾸어 저장한다. 이 십진수들을 Insertion sort나 merge sort를 이용해서 정렬을 한 후 다시 네 자리 16진수로 바꾸어서 출력한다.

1. 코드 설명.
2. 인풋 받기.

먼저 sort라는 변수를 통해서 1과 2중 하나의 값을 받는데 1이면 insertion sort를 이용해서 정렬을 하고 2이면 merge sort를 이용해서 내림차순 정렬을 한다.

그 후, input\_c라는 변수를 통해서 char형으로 인풋을 하나씩 받아들인다. while문의 맨 위에서 스페이스를 받으면 그냥 넘어가고 엔터를 받으면 인풋의 끝이므로 더 이상 인풋을 받지 않는다. 그 후 4개의 숫자를 받아들이면서 16진수를 십진수로 바꾸어 저장해야 한다. 16진수이므로 알파벳이 들어왔을 경우 A의 아스키 코드 값을 빼주고 10을 더해주면 A는 10, B는 11, C는 12, … , F는 15의 값으로 바뀌어 input\_i변수에 int형으로 저장된다. 숫자가 들어왔을 경우 char형으로 받았기 때문에 0의 아스키 코드 값을 빼주면 원래의 숫자 값으로 input\_i변수에 int형으로 저장할 수 있다.

그 후 16진수의 자릿값에 맞게 16을 거듭제곱 해주고 그 값들을 input이라는 변수에 더해준다. 네 자리의 모든 수들이 더해지면 dec 배열에 저장한다. 위와 같은 방법으로 인풋을 모두 받으면 dec 배열에 정렬해야 할 값들이 십진수로 저장된다.

1. Insertion\_Sort 함수

맨 처음에 받았던 sort변수의 값이 1이면 insertion sort를 이용하여 정렬을 해야 한다. 배열 arr에 정렬해야 할 값들이 들어있고 변수 index\_arr는 배열 arr의 길이이다. 배열의 앞부분부터 정렬을 진행하는데 arr[0]은 이미 정렬이 된 것으로 하고 진행하므로 1부터 정렬을 시작한다. 정렬할 칸의 값을 변수 key에 저장해놓고 정렬할 칸의 앞부분을 돌면서 key 값보다 작은 값(내림차순)을 가진 칸이 있는지 알아본다. 작은 값을 가진 칸이 있다면 그 값을 배열의 한 칸 위에 저장한다 (그 칸을 오른쪽으로 하나 민다고 생각하면 된다. 왼쪽이 낮은 index값이고 오른쪽이 높은 index값일 때). key값보다 작지 않은 칸을 만난다면 그 칸을 만나기 전 칸에 key값을 넣는다. 이런 식으로 반복하면 앞에서부터 하나씩 정렬이 되면서 정렬이 끝난다.

1. Merge\_Sort 함수

맨 처음에 받았던 sort변수의 값이 2이면 Merge sort를 이용하여 정렬을 해야 한다. 정렬할 배열을 arr라고 하고 그 배열의 첫 index를 p라고 하고 마지막 index를 r이라고 한다. 그리고 p와 r의 중간을 q라는 변수에 저장해서 정렬할 배열을 반씩 계속해서 recursive하게 나눈다. 그 후 Merge라는 함수를 통해 두 부분 배열을 정렬하면서 합친다.

1. Merge 함수

Merge함수에서는 먼저 두 개의 배열을 만들어서 합할 두 개의 배열을 옮겨 담는다. 이 두 개의 배열들은 각각 정렬이 완료 되어 있다. 배열의 마지막에는 이 배열의 값들로는 가질 수 없는 작은 수를 넣는다. 이 문제의 경우 인풋에 음수는 나오지 않으므로 -1을 넣는다. 그리고 두 배열을 앞에서부터 비교하며 큰 값들을 뽑아서 정렬한다.

1. dtoh 함수

dtoh 함수는 십진수를 네 자리 16진수로 바꾼다. 4칸짜리 char형 배열을 만들어서 각각의 칸에 16진수를 저장한다. 십진수를 16으로 나눈 나머지를 hexa라는 배열의 끝에서부터 저장하면 된다.

1. 마무리

인풋을 다 받고 정렬을 한 후에는 dec에 들어있는 각각의 십진수를 dtoh함수를 통해 16진수로 바꾸어서 바로 바로 출력해준다. 출력 후에는 숫자 간의 구분을 위해 스페이스를 출력해주는데 마지막 숫자 다음에는 스페이스를 출력하지 않기 위해 조건을 넣어준다.

1. 예시

1 0003 0010 0001 00F0 가 들어왔을 경우, 1이 변수 sort에 저장되고 0003이 십진수 3으로 바뀌어 dec[0]에 저장되고 0010이 십진수 16으로 바뀌어 dec[1]에 저장되고 0001이 십진수 1로 바뀌어 dec[2]에 저장되고 00F0이 십진수 240으로 바뀌어 dec[3]에 저장된다. 그리고 dec배열과 dec배열의 크기인 index가 매개변수로 Insertion\_Sort함수를 실행한다. dec배열에는 3 16 1 240 이렇게 4개의 숫자가 들어있고 3은 정렬이 완료된 것으로 가정하고 16부터 정렬을 시작한다. key값은 16이 되고 i는 0으로 와일문을 돈다. i는 0이고 dec[0]은 3으로 key값보다 작으므로 dec[1]은 3이 된다. i가 1줄어서 -1이 되었으므로 와일문을 나가고 dec[0]에 key값인 16이 들어가게 된다. 이제 dec배열에는 16 3 1 240 이 있고 16과 3은 정렬이 된 상태이다. 위와 같은 방법으로 1과 240도 정렬하면 240 16 3 1 의 순서로 정렬이 되게 된다. 이제 Insertion\_Sort함수를 나와서 dec배열의 값들을 16진수로 바꾸어서 출력만 하면 된다.