

자료구조응용

10. 연결리스트 : 기초 (10점)

2022.4.6.

1. 다음과 같이 정렬되지 않는 정수 데이터를 입력하면서 정렬된 체인(singly linked list or chain)를 만들고 실행 예와 같이 수행되는 프로그램을 작성하라. (10점)

[실행 순서]

- ① 입력파일("input.txt")로 부터 데이터를 입력받으면서 정렬된 체인을 만든다. 입력데이터는 정렬되지 않은 값으로 중복 가능하다.

input.txt										
50	80	30	20	19	90	30	55	77	30	
99	45	55	89	91	10	20	66	38	59	
22	55	88	22	66	29	50	95	78	83	

- ② 체인의 처음부터 끝까지 노드 데이터를 출력한다.
③ 성적이 50점 이하인 노드를 연결리스트에서 삭제한다.
④ 체인의 처음부터 끝까지 노드 데이터를 출력한다.
⑤ 체인의 모든 노드를 삭제한다.

[구현 세부사항]

- ① 구조체 정의문

```
typedef struct listNode *listPointer;
typedef struct listNode {
    int data;
    listPointer link;
} listNode;
listPointer first = NULL;
```

- ② 정수 데이터 시퀀스 파일입력 및 정렬된 리스트 만들기

```
int main(void)
{
    int data;
    listPointer x, trail;

    // data input for each node
    FILE *fp;
    if((fp = fopen("input.txt", "r")) == NULL )
    {
        fprintf(stderr, "cannot open the file");
        exit(EXIT_FAILURE);
    }
}
```

```

// 입력방법 1
// 파일 끝의 white character(엔터키, 탭키, 한 칸 띄우기)에 상관없이 마지막 정수 입력 됨
while (fscanf(fp, "%d", &data) != EOF)
{
    find(first, &x, data);        // find insert position
    insert(&first, x, data);      // insert data first after node x.
}

// 입력방법 2
/*
// 파일 끝에 white character(엔터키, 탭키, 한 칸 띄우기) 필요
fscanf(fp, "%d", &data);
while( !feof(fp) )
{
    find(first, &x, data);        // find insert position
    insert(&first, x, data);      // insert data first after node x.
    fscanf(fp, "%d", &data);
}
*/

:

fclose(fp);

:

return 0;
}

```

③ 함수정의

```

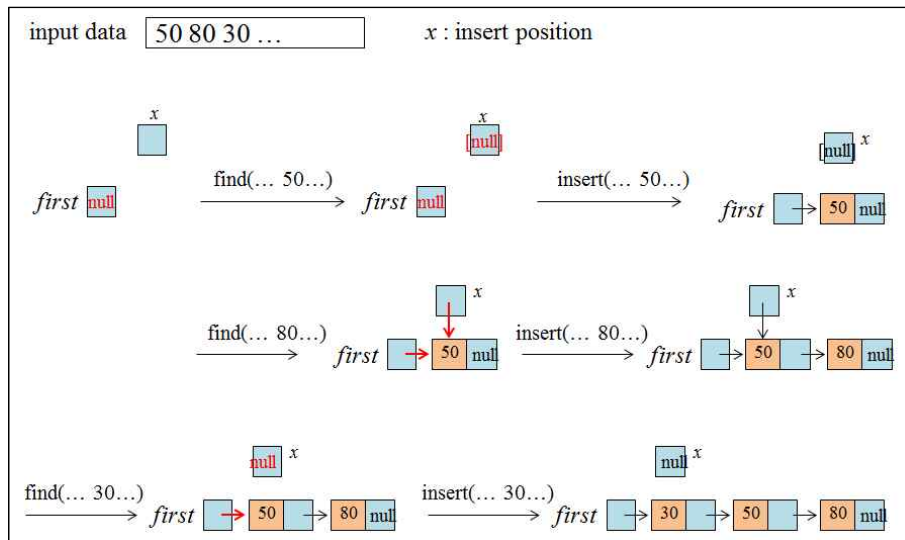
void find(...)
{
    ...
    if( 빈 리스트 )
    { // empty list
        ...
    }
    else
    { //non-empty list
        ...
    }
}

```

```

void insert(...)
{
    ...
    if( 빈 리스트 )
    { // empty list
        ① 첫 노드로 추가
    }
    else
    { //non-empty list
        if( 첫 노드 추가 )
            ② 첫 노드로 추가
        else
            ③ 두 번째 이후 노드로 추가
    }
}

```



<정렬된 연결리스트 생성 과정>

```

void delete(listPointer *first, listPointer trail,
            listPointer x)
{
    /* delete x from the list, trail is the preceding node
       and *first is the front of the list */
    if (trail)
        trail->link = x->link;
    else
        *first = (*first)->link;
    free(x);
}

```

Program 4.3: Deletion from a list

※ 주의: delete는 C++ 의 keyword!, 소스파일 확장자가 cpp이면 컴파일 에러

```

void printList(listPointer first)
{
    printf("The list contains: ");
    for (; first; first = first->link)
        printf("%4d", first->data);
    printf("\n");
}

```

Program 4.4: Printing a list

※ 한 줄에 10개씩 출력되게 수정하라.

[실행예]

```

C:\WINDOWS\system32\cmd.exe
The ordered list contains:
10 19 20 20 22 22 29 30 30 30
38 45 50 50 55 55 59 66 66
77 78 80 83 88 89 90 91 95 99

After deleting nodes with data less than and equal to 50
The ordered list contains:
55 55 55 59 66 66 77 78 80 83
88 89 90 91 95 99
계속하려면 아무 키나 누르십시오 . . .

```

■ 제출 형식

- 솔루션 이름 : DS 10
- 프로젝트 이름 : 1
- 각 소스파일에 주석처리
“학번 이름”

“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

- 실행화면을 캡처한 보고서를 작성 후 pdf 파일로 변환하여 솔루션 폴더에 포함
- 솔루션 정리 메뉴를 수행 후 전체 솔루션을 “학번.zip”으로 압축하여 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 1차 마감 (LMS 과제 마감일) : 수업일 자정
- 2차 마감 (LMS 과제 열람기한) : 수업 익일 자정(만점의 50%, 반올림)
- 1, 2차 마감 이외의 제출은 허용하지 않습니다. (이메일 제출 불가!)