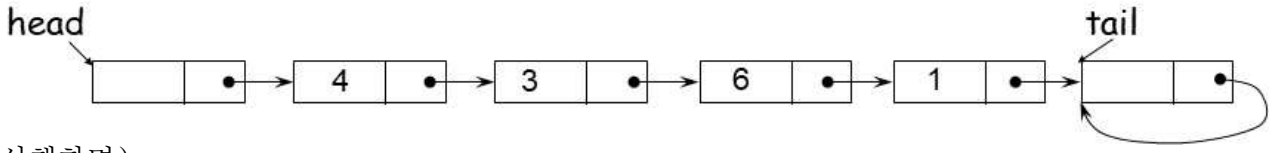
자료구조 3주차 과제

제출일:
학번/이름:

문1) 다음과 같은 단순연결리스트를 만들고, 리스트 내의 원소들을 출력한 후 주어진 리스트에서 가장 큰 정수값을 출력하는 프로그램을 작성하시오. 정수 단순연결리스트의 헤드 노드 주소를 받아 가장 큰 원소의 값을 반환하는 함수인 int max_element(node *head)를 작성하여 사용하시오. (비어있는 리스트이면 -999999를 반환함)



(실행화면)
리스트 생성 완료!
리스트 원소: 4 3 6 1
가장 큰 원소의 값: 6

(참고 프로그램)

```
typedef struct _node
{
    int key;
    struct _node *next;
} node;

node* init_list(void){
    node *head = (node*)malloc(sizeof(node));
    node *tail = (node*)malloc(sizeof(node));
    // malloc이 NULL을 리턴할 경우 처리 방법은??
    head->next = tail;
    tail->next = tail;

    return head;
}

node *insert_after(int k, node* t){
    node *s;
    if (!t || t->next == t) return t;
```

```
    s = (node*)malloc(sizeof(node));
    if (s==NULL) return s;
    s->key = k;
    s->next = t->next;
    t->next = s;
    return s;
}

void print_list(node* h) {
    h = h->next;
    while (h != h->next){
        printf("%d ", h->key);
        h = h->next;
    }
}

int main() {
        node *head = init_list();
        node *p;
        insert_after(4,head);
        insert_after(3,head);
        insert_after(6,head);
        insert_after(1,head);

        print_list(head);
}
```

문2) 숫자를 입력받아 이 숫자로 이루어진 단순연결 리스트를 만들고, 리스트 원소들의 합을 계산하는 프로그램을 작성하시오.(비어있는 리스트이면 0을 리턴함)

(실행화면)
원소의 개수를 입력하세요: 5
숫자를 입력하세요: 4 5 6 7 10
리스트 생성 완료! (리스트 원소 : 4 5 6 7 10)
원소의 합: 30

문3) 1부터 10까지의 숫자로 이루어진 이중연결 리스트를 만들고, 이중연결 리스트의 노드들의 순서를 반대로 하는 reverse_list 함수를 작성하여 위 리스트에 적용한 후 결과를 출력하시오.

(head와 tail을 전역변수로 사용)

(실행화면)
리스트 생성 완료!
리스트 원소 : 1 2 3 4 5 6 7 8 9 10
reverse_list 결과: 10 9 8 7 6 5 4 3 2 1

(참고 프로그램)

```
#include <stdio.h>
#include <memory.h>
#include <stdlib.h>

typedef struct _dnode
{
    int key;
    struct _dnode *prev;
    struct _dnode *next;
} dnode;

dnode *head, *tail;

void init_dlist(void)
{
    head = (dnode*)malloc(sizeof(dnode));
    tail = (dnode*)malloc(sizeof(dnode));
    head->next = tail;
    head->prev = head;
    tail->next = tail;
    tail->prev = head;
}

dnode *insert_dnode_ptr(int k, dnode *t)  /* insert k, before t */
{
    dnode *i;
    if (t == head)
        return NULL;
    i = (dnode*)malloc(sizeof(dnode));
    i->key = k;
    t->prev->next = i;
    i->prev = t->prev;
```

```
    t->prev = i;
    i->next = t;
    return i;
}

void print_dlist(dnode *p)
{
    while (p != tail)
    {
        printf("%d ", p->key);
        p = p->next;
    }
}

void main(void)
{
    int i;
    init_dlist();

    for (i=1; i<=10; i++) {
        insert_dnode_ptr(i, tail);

    }
    print_dlist(head->next);
}
```