문제해결기법(13967005)

202135592 한웅재

소프트웨어

제출일: 2021. 11. 20

Q1. P1 (p. 66)

#define \_CRT\_SECURE\_NO\_WARNINGS// or scanf\_s

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <ctype.h>

#include <stdbool.h>

struct NODE {

int key;

struct NODE\* next;

};

int InsertKey(struct NODE\* head, int value, struct NODE\*\* address\_of\_new\_first\_node)

{

/\* Start from head->next instead of head \*/

struct NODE\* p = head->next, \* prev = head;

struct NODE\* new\_node;

int found=0;

int count = 0;

while (p) {

if (p->key > value)

{

found = 0;

break;

}

else if (p->key == value)

{

found = -1;

return found;

}

count++;//to found new head

prev = p;

p = p->next;

}

if (count) {

new\_node = (struct NODE\*)malloc(sizeof(struct NODE));

new\_node->key = value;

prev->next = new\_node; /\* adjust next pointers \*/

new\_node->next = p;

\*address\_of\_new\_first\_node = NULL;

}

else if (count == 0) {

new\_node = (struct NODE\*)malloc(sizeof(struct NODE));

new\_node->key = value;

prev->next = new\_node; /\* adjust next pointers \*/

new\_node->next = p;

\*address\_of\_new\_first\_node= new\_node;

}

return found;

}

void ScanList(struct NODE\* node) {

printf("\nTraversing the linked list..\n");

struct NODE\* ptr= node->next;

int count = 0;

while (ptr != NULL) {

printf("node[%d] key: %d\n",count,ptr->key);

ptr = ptr->next;

count++;

}

}

int main() {

struct NODE nodes[7];

struct NODE \*head= (struct NODE\*)malloc(sizeof(struct NODE));

struct NODE\*\* address\_of\_head;

struct NODE\* new\_head = head;

address\_of\_head = &new\_head;

for (int i = 0; i < 7; i++) {

nodes[i].next = NULL;

}

nodes[0].key = 100;

nodes[1].key = 250;

nodes[2].key = 467;

nodes[0].next = &nodes[1];

nodes[1].next = &nodes[2];

head->next = &nodes[0];

int keys[4] = {250,300,50,500};

//insertion

for (int i = 0; i < sizeof(keys)/sizeof(int); i++) {

printf("//---Inserting Key : %d\n", keys[i]);

if (InsertKey(head, keys[i], address\_of\_head) == 0) {

printf("Key inserted..\n");

if (new\_head)

printf("New first node created..\n");

}

else {

printf("Key already exists..\n");

}

}

//

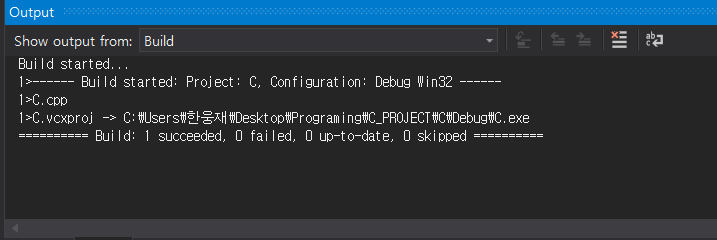
//travel

ScanList(head);

//

return 0;

}

텍스트이(가) 표시된 사진

자동 생성된 설명

Q2. P2 (p. 67)

#define \_CRT\_SECURE\_NO\_WARNINGS// or scanf\_s

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <ctype.h>

#include <stdbool.h>

struct NUM {

int key;

struct NUM\* next;

};

void Insert(struct NUM\* head, int value)

{

/\* Start from head->next instead of head \*/

struct NUM\* p = head->next, \* prev = head;

struct NUM\* new\_node;

int count = 0;

new\_node = (struct NUM\*)malloc(sizeof(struct NUM));

new\_node->key = value;

if (p->key < 0) {

prev->next = new\_node;

new\_node->next = p->next;

}

while (p) {

if (p->key > value) break;

prev = p;

p = p->next;

count++;

}

//if (count == 0) {

// new\_node = (struct NUM\*)malloc(sizeof(struct NUM));

// new\_node->key = value;

//

//}

prev->next = new\_node; /\* adjust next pointers \*/

new\_node->next = p;

head->key++;

}

void ClearList(struct NUM\* LIST) {

struct NUM\* temp = LIST->next;

struct NUM\* prev = NULL;

printf("\nDeleting the linked list..\n");

for (int i = 0; i < LIST->key;i++) {

prev = temp;

temp = temp->next;

free(prev);

printf("node[%d] deleted..\n", i);

}

}

int main() {

int nums[10] = { 17, 39, 11, 9, 42, 12, 15, 8, 13, 41 };

struct NUM\* nodes = (struct NUM\*)malloc(10\*sizeof(struct NUM));

//insertion

struct NUM\* head=(struct NUM\*)malloc(sizeof(struct NUM));

head->next = nodes;

head->key = 0;

for (int i = 0; i < 7; i++) {

nodes[i].next = NULL;

}

for (int i = 0; i < 10; i++) {

printf("//---Inserting Key : %d\n", nums[i]);

Insert(head, nums[i]);

printf("[%d] elements. Key inserted..\n",head->key);

}

printf("");

//travels

printf("\nTraversing the linked list..\n");

struct NUM\* ptr = head->next;

int count = 0;

while (ptr != NULL) {

printf("node[%d] key: %d\n", count, ptr->key);

ptr = ptr->next;

count++;

}

//

ClearList(head);

return 0;

}

텍스트이(가) 표시된 사진

자동 생성된 설명