문제해결기법(13967005)

202135592 한웅재

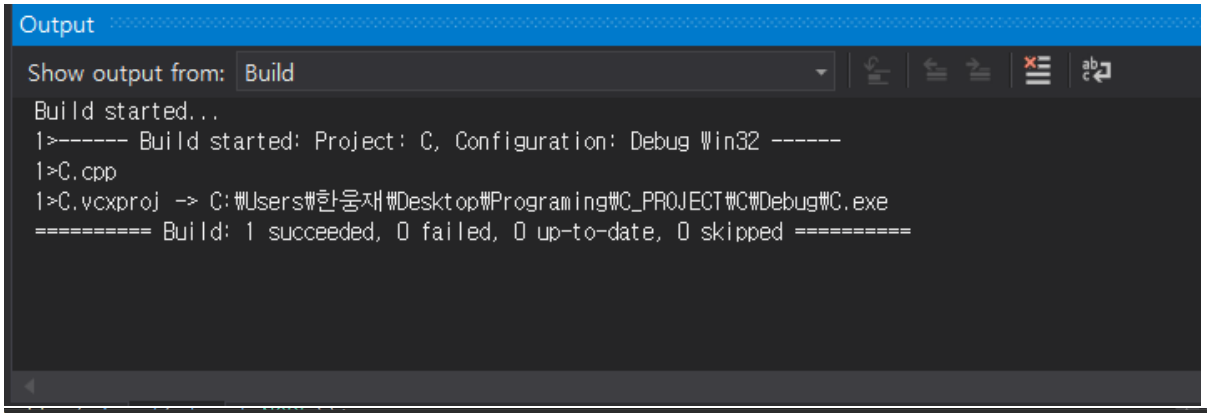
소프트웨어

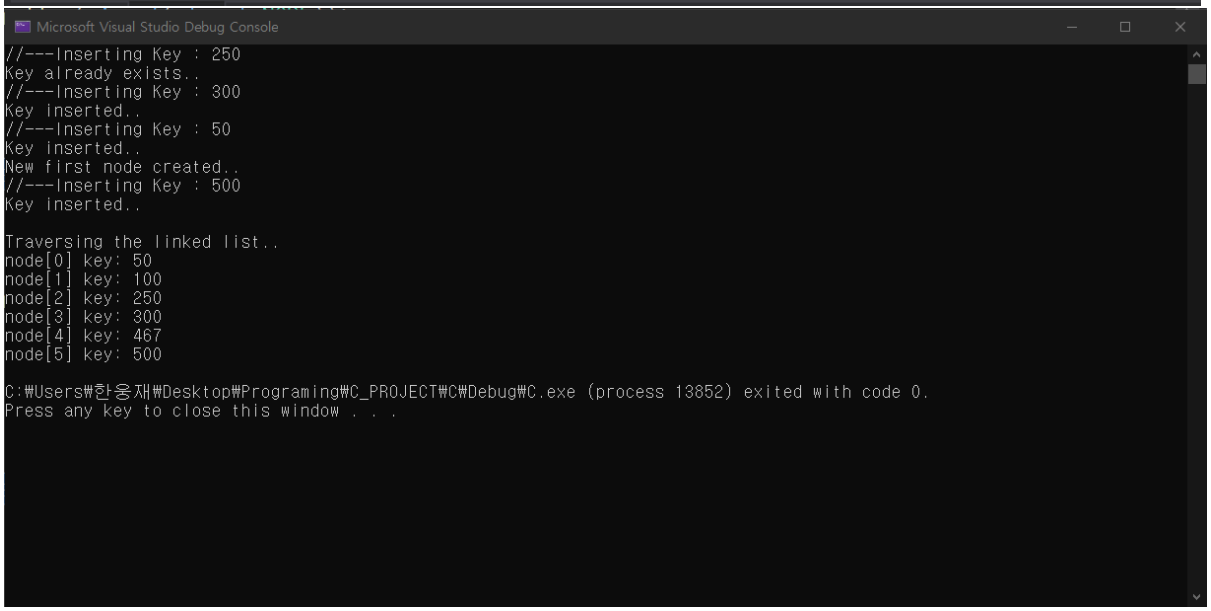
제출일: 2021. 11. 20

Q1. P1 (p. 66)

```c
#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;
}
```

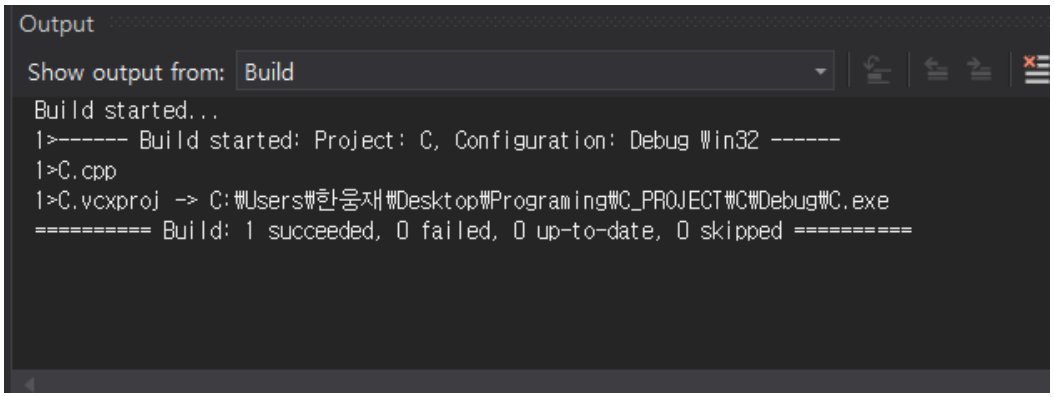

Output
Show output from: Build
Build started...
1>------ Build started: Project: C, Configuration: Debug Win32 ------
1>C.cpp
1>C.vcxproj -> C:₩Users₩한웅재₩Desktop₩Programing₩C_PROJECT₩C₩Debug₩C.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========

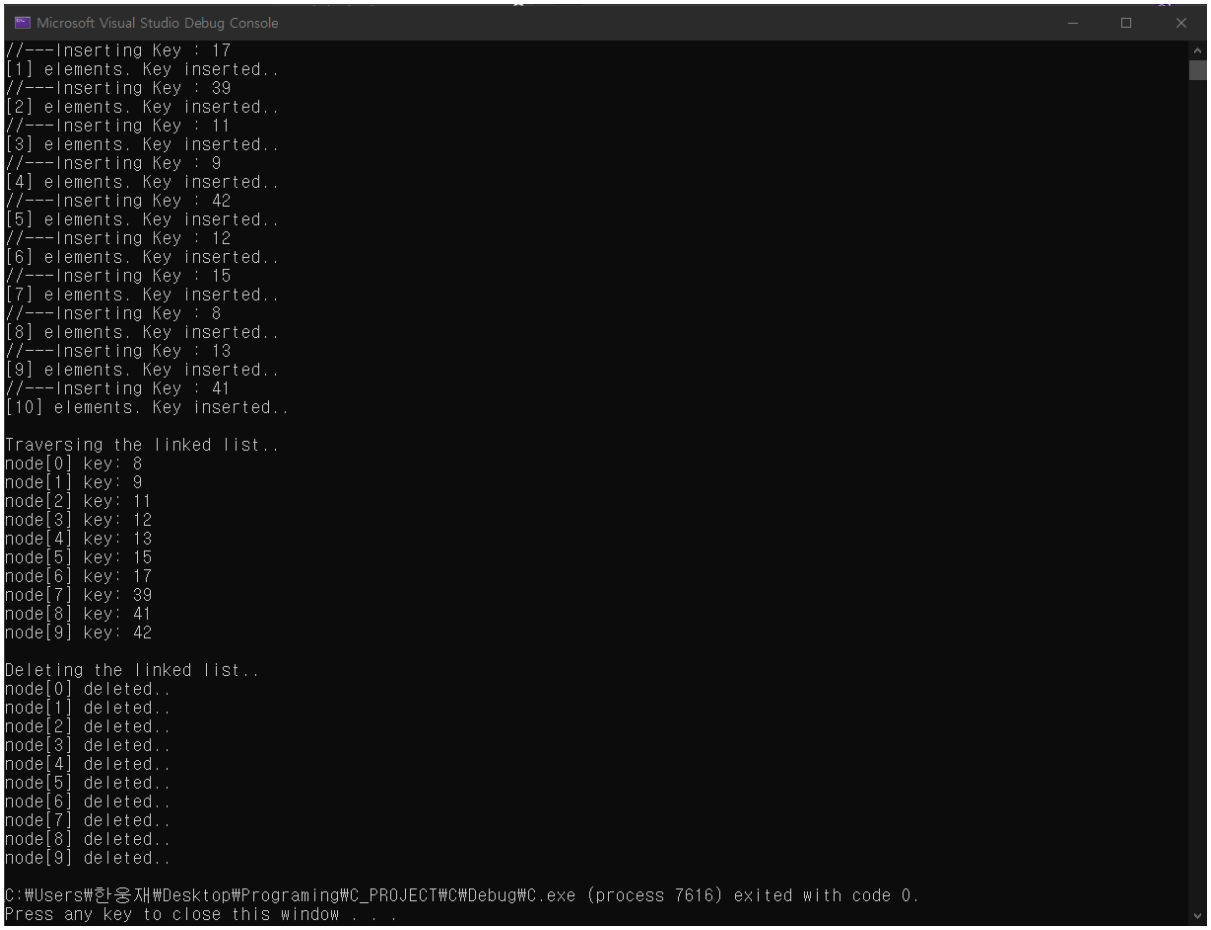

Microsoft Visual Studio Debug Console
//---Inserting Key : 250
Key already exists..
//---Inserting Key : 300
Key inserted..
//---Inserting Key : 50
Key inserted..
New first node created..
//---Inserting Key : 500
Key inserted..
Traversing the linked list..
node[0] key: 50
node[1] key: 100
node[2] key: 250
node[3] key: 300
node[4] key: 467
node[5] key: 500
C:₩Users₩한웅재₩Desktop₩Programing₩C_PROJECT₩C₩Debug₩C.exe (process 13852) exited with code 0.
Press any key to close this window . . .

Q2. P2 (p. 67)

```c
#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;
}
```




Microsoft Visual Studio Debug Console
//---Inserting Key : 17
[1] elements. Key inserted..
//---Inserting Key : 39
[2] elements. Key inserted..
//---Inserting Key : 11
[3] elements. Key inserted..
//---Inserting Key : 9
[4] elements. Key inserted..
//---Inserting Key : 42
[5] elements. Key inserted..
//---Inserting Key : 12
[6] elements. Key inserted..
//---Inserting Key : 15
[7] elements. Key inserted..
//---Inserting Key : 8
[8] elements. Key inserted..
//---Inserting Key : 13
[9] elements. Key inserted..
//---Inserting Key : 41
[10] elements. Key inserted..
Traversing the linked list..
node[0] key: 8
node[1] key: 9
node[2] key: 11
node[3] key: 12
node[4] key: 13
node[5] key: 15
node[6] key: 17
node[7] key: 39
node[8] key: 41
node[9] key: 42
Deleting the linked list..
node[0] deleted..
node[1] deleted..
node[2] deleted..
node[3] deleted..
node[4] deleted..
node[5] deleted..
node[6] deleted..
node[7] deleted..
node[8] deleted..
node[9] deleted..
C:₩Users₩한웅재₩Desktop₩Programing₩C_PROJECT₩C₩Debug₩C.exe (process 7616) exited with code 0.
Press any key to close this window . . .