SHIN,CHAE UN (신채운) / 202135789

## Source codes & Comment

// 아래 네모 안에 코드를 복사하여 붙일 것

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <string.h>  #include <stdlib.h>  /\*  name:Shin Chae Un  Date:11/18  Student number: 202135789  description:  \*/  struct NODE {  int key;  struct NODE\* next;  };  void insert();  void main() {  insert();  }  void insert() {  struct NODE\* node;  node = (struct NODE\*)malloc(3 \* sizeof(struct NODE));  if (node == NULL) {  printf("malloc failed");  exit(1);  }  if (node != NULL) {  (node[0]).next = &node[1];  (node[1]).next = &node[2];  (node[2]).next = NULL;  (node[0]).key = 100;  (node[1]).key = 250;  (node[2]).key = 467;  }  struct NEW\_NODE {  int key=50;  int \* next;  };  struct NEW\_NODE\* new\_node;  struct NODE\* ptr,\*prv\_ptr = NULL;  int newkey = 50;  ptr = &node[0];  while (ptr)  {  if ((\*ptr).key == newkey)  {  printf("key already exists");  break;  }  if ((\*ptr).key < newkey) {  prv\_ptr = ptr;  ptr = (\*ptr).next;  }  else {  (\*prv\_ptr).next = &node[3];  node[3].next = ptr;  printf("key inserted");  break;  }  }  }; |

## Inspect the program (testing)

// 자가 점검 후 네모 안에 v표시

**🗹** Check loop, if else, switch, function.

**🗹** Check variable initialization.

**🗹** Check pointers.

## Test cases & Output (Screenshots)

|  |  |
| --- | --- |
| # | [스크랜 샷 설명] |
| (유의사항: 전체 화면을 캡쳐 하지 말고, 실행 화면만 잘라 붙여 결과를 잘 확인할 수 있게 처리바람) | |

SHIN,CHAE UN (신채운) / 202135789

## Source codes & Comment

// 아래 네모 안에 코드를 복사하여 붙일 것

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <string.h>  #include <stdlib.h>  /\*  name:Shin Chae Un  Date:11/18  Student number: 202135789  description:  \*/  struct NODE {  int key;  struct NODE\* next;  };  void Delete( );  void main() {  Delete( );  }  void Delete( ) {  struct NODE\* node;  node = (struct NODE\*)malloc(3 \* sizeof(struct NODE));  if (node == NULL) {  printf("malloc failed");  exit(1);  }  if (node != NULL) {  (node[0]).next = &node[1];  (node[1]).next = &node[2];  (node[2]).next = NULL;  (node[0]).key = 100;  (node[1]).key = 250;  (node[2]).key = 467;  }  struct NODE\* ptr, \* prv\_ptr = NULL;  int delkey, deleted = 0;  delkey = 100;  ptr = &node[0];  while (ptr) {  if ((\*ptr).key == delkey) {  if (prv\_ptr != NULL)  (\*prv\_ptr).next = (\*ptr).next;  (\*ptr).key = -1;  (\*ptr).next = NULL;  deleted = 1;  break;  }  prv\_ptr = ptr;  ptr = (\*ptr).next;  }  if (deleted)  printf("node deleted.");  else  printf("key not found");  }; |

## Inspect the program (testing)

// 자가 점검 후 네모 안에 v표시

**🗹** Check loop, if else, switch, function.

**🗹** Check variable initialization.

**🗹** Check pointers.

## Test cases & Output (Screenshots)

|  |  |
| --- | --- |
| # | [스크랜 샷 설명] |
| (유의사항: 전체 화면을 캡쳐 하지 말고, 실행 화면만 잘라 붙여 결과를 잘 확인할 수 있게 처리바람) | |

SHIN,CHAE UN (신채운) / 202135789

## Source codes & Comment

// 아래 네모 안에 코드를 복사하여 붙일 것

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <string.h>  #include <stdlib.h>  /\*  name:Shin Chae Un  Date:11/18  Student number: 202135789  description:  \*/  struct NODE {  int key;  struct NODE\* next;  };  void insert(int key);  void ScanList();  void main() {  struct HEAD {  int key;  int\* next;  };  struct HEAD \*head;  insert(250);  insert(300);  insert(50);  insert(500);  }  void insert(int key) {  struct NODE\* node;  node = (struct NODE\*)malloc(3 \* sizeof(struct NODE));  if (node == NULL) {  printf("malloc failed");  exit(1);  }  if (node != NULL) {  (node[0]).next = &node[1];  (node[1]).next = &node[2];  (node[2]).next = NULL;  (node[0]).key = 100;  (node[1]).key = 250;  (node[2]).key = 467;  }  struct NEW\_NODE {  int key = 50;  int\* next;  };  struct NEW\_NODE\* new\_node;  struct NODE\* ptr, \* prv\_ptr = NULL;  int newkey = key;  ptr = &node[0];  while (ptr)  {  if ((\*ptr).key == newkey)  {  printf("key already exists");  break;  }  if ((\*ptr).key < newkey) {  prv\_ptr = ptr;  ptr = (\*ptr).next;  }  else {  (\*prv\_ptr).next = &node[3];  node[3].next = ptr;  printf("key inserted");  break;  }  }  };  void ScanList() {    } |

## Inspect the program (testing)

// 자가 점검 후 네모 안에 v표시

**🗹** Check loop, if else, switch, function.

**🗹** Check variable initialization.

**🗹** Check pointers.

## Test cases & Output (Screenshots)

|  |  |
| --- | --- |
| # | [스크랜 샷 설명] |
| (유의사항: 전체 화면을 캡쳐 하지 말고, 실행 화면만 잘라 붙여 결과를 잘 확인할 수 있게 처리바람) | |