|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C2\_실습과제\_12주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

* **강의내용**

1. 메모리 할당 : calloc(), realloc()

#include <stdio.h>

#include <stdlib.h> //realloc(), calloc()

void display(int \*, int, char \*);

int main(void) {

int \*reary, \*cary;

int i = 0;

if ((cary = (int \*)calloc(3, sizeof(int))) == NULL){

printf("메모리 할당에 문제가 있습니다.\n");

exit(EXIT\_FAILURE);

}

display(cary, 3, "calloc() 초기화 생략");

for (i = 0; i < 3; i++)

\*(cary + i) = i + 10;

display(cary, 3, "calloc() 초기화 후");

if ((reary = (int \*)realloc(cary, 4 \* sizeof(int))) == NULL){

printf("메모리 할당이 문제가 있습니다.\n");

exit(EXIT\_FAILURE);

}

display(reary, 4, "ralloc() 초기화 생략");

free(reary);

return 0;

}

void display(int \*pt, int size, char \*str){

int i;

printf("=== %s ====\n", str);

for (i = 0; i < size; i++)

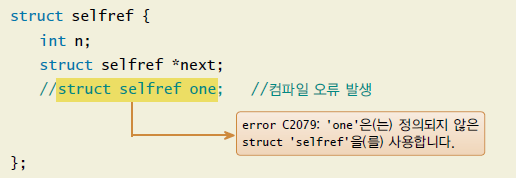
printf("[%d] : %3d\n",i, \*(pt + i));

printf("\n");

}

|  |
| --- |
| * 실행결과 |

1. 자기 참조 구조체 : 구조체의 멤버 중의 하나가 자기 자신의 구조체 포인터 변수를 갖는 구조체



/\* 링크트 리스트를 따라 가면서 내용을 출력하는 예제 프로그램 \*/

struct list {

char \*name;

char sex;

int age;

struct list \*next; //자신과 동일한 구조체 형에 대한 포인터, 자기참조구조체

} \*ptr, x = {"Kim", 'M', 36}, y = {"Jeong", 'F', 35}, z = {"Choi", 'M', 41};

int main(){

x.next = &y;

y.next = &z;

z.next = '\0';

printf("이름 성 나이\n");

printf("------------------\n");

for (ptr=&x; ptr; ptr=ptr->next)

printf("%-8s %-6c %2d\n", ptr->name, ptr->sex, ptr->age);

return 0;

}

|  |
| --- |
| * 실행결과 |

1. 링크드 리스트

**======== linkedlist.h ==========**

#include <stdio.h> //<시스템헤더파일.h> 기술

#include <stdlib.h>

#include <string.h>

struct linked\_list {

char \*name;

struct linked\_list \*next;

};

typedef struct linked\_list NODE;

typedef NODE \* LINK;

**============ linkedlist.c ===============**

#include "linkedlist.h" //사용자 정의 헤더 파일

LINK createNode(char \*name); //노드 생성

LINK append(LINK head, LINK cur); //노드 추가

void printList(LINK head); //연결 리스트 출력

int main(void){

char name[30];

LINK head = NULL;

LINK cur;

printf("이름을 입력하고 Enter를 누르세요. >> \n");

while (gets\_s(name, 30) != NULL){

cur = createNode(name); //노드 동적 할당

if (cur == NULL)

exit(1);

head = append(head, cur); //맨 뒤에 노드추가

printList(head); //연결 리스트 모두 출력

}

return 0;

}

**============ listlib.c ===============**

#include "linkedlist.h" // 사용자 정의 헤더 파일

LINK createNode(char \*name){ //노드를 생성하는함수

LINK cur;

cur = (NODE \*)malloc(sizeof(NODE));

if (cur == NULL) {

printf("노드 생성을 위한 메모리할당에 문제가 있습니다.\n");

return NULL;

}

//언어 이름을 저장 할 문자 배열을 동적 할당하여 name에 저장

cur->name = (char \*)malloc(sizeof(char) \* (strlen(name) + 1));

strcpy\_s(cur->name, strlen(name)+1, name); //name에 저장된 문자열 복사

cur->next = NULL;

return cur; //할당 된 메모리 주소 반환

}

LINK append(LINK head, LINK cur) { //노드를 추가하는 함수

LINK nextNode = head;

if (head == NULL){ //첫 노드 추가일 경우

head = cur; //헤드 노드는 첫 번째 노드가 됨

return head;

}

//두 번째 노드 추가 부터

while (nextNode->next != NULL){ //마지막 노드 찾기

nextNode = nextNode->next;

}

nextNode->next = cur; //마지막 노드에 추가

return head;

}

void printList(LINK head) { //연결 리스트의 모든 노드 출력 함수

int cnt = 0;

LINK nextNode = head;

while (nextNode != NULL) {

printf("%3d번째 노드는%s\n", ++cnt, nextNode->name);

nextNode = nextNode->next;

}

//return cnt;

}

|  |
| --- |
| * 실행결과 |

* 중간 점검

1. **다음에서 설명하는 문장을 작성하시오.**
2. int형 포인터 변수 p에 int형 자료를 저장할 수 있는 공간을 할당하여 값 10을 저장하는 문장

int \*p = NULL;

if((p = (int \*)malloc(sizeof(int)))==NULL)

{

printf(“메모리 할당에 문제가 있습니다.”);

exit(1);

}

\*p = 10;

1. int 형 포인터 변수 pi에 int 형 원소 3개를 저장할 수 있는 배열 공간을 할당하여 각각 1, 3, 5를 저장하는 문장

int \*pi = NULL;

int i;

if((pi = (int \*)calloc(3, sizeof(int))) == NULL)

{

printf(“메모리 할당에 문제가 있습니다.”);

exit(1);

}

\*(p+0) = 1;

\*(p+1) = 3

\*(p+2) = 5;

1. 위에서 할당된 메모리를 원소 5개로 확장하여 메모리를 할당하고 확장된 마지막 두 원소에 각각 7, 9를 저장하는 문장

int \*ary = NULL;

if((ary = (\*int)realloc(pi, sizeof(int)\*5))==NULL)

{

printf(“메모리 할당에 문제가 있습니다.”);

exit(1);

}

\*(p+3) = 7;

\*(p+4) = 9;

1. 함수 realloc()을 이용하여 int 형 포인터 변수 p에 int 자료를 저장할 수 있는 공간을 할당하여 각각 100을 저장하는 문장

int \*p = NULL;

if((p = (int \*)realloc(ary, sizeof(int)\*6)) = NULL)

{

printf(“메모리 할당에 문제가 있습니다.”);

exit(1);

}

\*(p+5) = 100;

1. 함수 calloc()을 이용하여 int 형 포인터 변수 pc에 int 형 원소 2개를 저장하는 배열 공간을 할당하는 문장

int \*pc = NULL;

if((pc = (int \*)calloc(2, sizeof(int))) == NULL)

{

printf(“메모리 할당에 문제가 있습니다.”);

exit(1);

}

1. 구조체 멤버로 char 형인 ch 변수와 자신의 포인터 변수인 next를 갖는 자기 참조 구조체 struct data를 정의하는 문장

struct data

{

char ch;

struct data \*next;

};

1. 위 구조체 struct data를 NODE로 재정의하는 문장

typedef struct data NODE;

1. NODE 포인터 head를 선언하고 NULL을 저장하는 문장

typedef \*NODE head;

1. **다음 문장에서 문법오류를 찾아 수정하시오.**
2. int p = malloc( sizeof(int) );

int \*p = (int \* )malloc(sizeof(int))

1. double \*p = (int) malloc( sizeof(int) );

double \*p = (double \* ) malloc(sizeof(double));

or

int \*p = (int \*)malloc(sizeof(int));

1. int \*pi = malloc( sizeof(int) ); pi = 3;

int \*pi = (int \*)malloc(sizeof(int));

\*pi = 3;

1. int \*pi = malloc(3, sizeof(int) );

int \*pi = (int \*)malloc(sizeof(int)\*3);

1. double \*pd = calloc(sizeof(double), 5);

double \*pd = (double \*)calloc(5, sizeof(double));

1. double \*pd = (double \*) realloc( 3\*sizeof(int) );

double \*pd = (double \*)realloc(3, sizeof(int));

1. struct mydata {

int data;

struct mydata link;

};

struct mydata {

int data;

struct mydata \*link;

}

1. struct node {

int data;

node \* link;

};

1. struct node {

int data;

struct node \* link;

};

* 프로그램 과제

1. 다음 결과를 참고로 문자열을 입력하여 출력하도록 함수 calloc()을 사용하여 프로그램을 작성하시오.

* 입력하는 문자열을 모두 동적 할당한 메모리에 저장 후 다음과 같이 출력

입력할 영문자의 갯수를 입력 >> 11

11개의 영문자 입력 >> Objective-C

입력한 문자열: Objective-C

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

char \*ary = NULL;

int n = 0;

printf("입력할영문자의갯수를입력>> ");

return 0;

}

|  |
| --- |
| <소스>  #include <stdio.h>  #include <stdlib.h>  int main(void)  {  char \*ary = NULL;  int n = 0;  int i;  printf("입력할영문자의갯수를입력>> ");  scanf("%d", &n);    if ((ary = (char \*)calloc(n, sizeof(char))) == NULL)  {  printf("메모리 할당에 실패했습니다. \n");  exit(1);  }  printf("%d개의 영문자 입력 >> ", n);    scanf\_s("%s", ary, 11);    printf("입력된 문자: ");  for (i = 0; i < n; i++)  {  printf("%c", \*(ary+i));  }  printf("\n");  return 0;  } |
| <실행결과> |

1. 입력 받은 레코드를 링크드 리스트에 저장한 후 출력하는 프로그램을 작성하시오. 제시된 구조체를 사용하며 합은 두 개의 성적을 입력 받은 후 계산하여 저장한다.

리스트 출력 후 입력한 문자열과 일치하는 레코드를 리스트에서 검색하여 출력한다.

struct linked\_list {

int num;

char name[10];

int sc1;

int sc2;

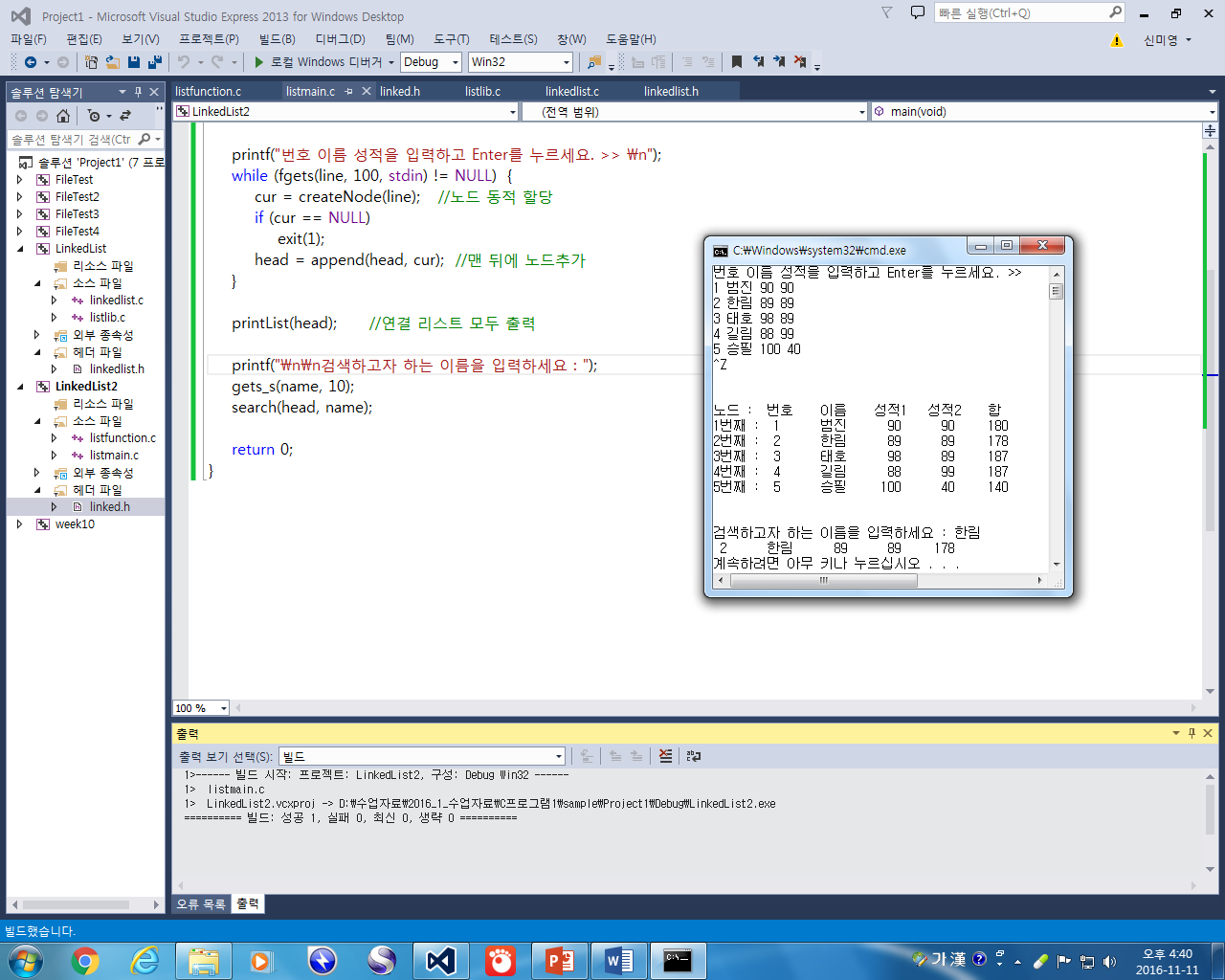
int hap;

struct linked\_list \*next;

};

typedef struct linked\_list NODE;

typedef NODE \* LINK;



char line[100];

gets\_s(line, 100);

sscanf\_s(line, “%d %s %d %d %d ”, &(p->d1), p->name).

|  |
| --- |
| <소스>  =====================Node.h============================  #pragma once  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  struct linked\_list {  int num;  char name[10];  int sc1;  int sc2;  int hap;  struct linked\_list \*next;  };  typedef struct linked\_list NODE; //linked\_list 구조체를 NODE로 재정의  typedef NODE \* LINK; //구조체 포인터 LINK를 정의  =====================Main.c============================  #include "Node.h"  LINK createNode(char \*line);  LINK append(LINK cur, LINK head);  void printData(LINK head);  void findStudent(char \*, LINK head);  int main()  {  char line[100];  LINK head = NULL;  LINK cur;  char findName[10];  printf("번호 이름 성적을 입력하고 Enter을 누르세요. >> \n");  while (gets\_s(line, 100) != NULL)  {  cur = createNode(line);  if (cur == NULL)  {  exit(1);  }  head = append(cur, head);    }  printData(head);  printf("검색하고자 하는 이름을 입력하세요 : ");  scanf\_s("%s", findName, 10);  findStudent(findName, head);  }  =====================Fuction.c============================  #include "Node.h"  LINK createNode(char \*line)  {  LINK cur; //새로운 노드를 나타내는 구조체 포인터 생성 (== NODE \*cur)    if ((cur = (NODE \*)malloc(sizeof(NODE))) == NULL) //  {  printf("메모리 할당을 실패하였습니다. \n");  return NULL;  }    sscanf\_s(line, "%d %s %d %d", &(cur->num), cur->name, 10, &(cur->sc1), &(cur->sc2)); //line에 저장된 정보를 구조체 데이터에 저장  cur->hap = (cur->sc1) + (cur->sc2);  cur->next = NULL;    return cur;  }  LINK append(LINK cur, LINK head)  {  LINK temp = head; // head의 주소를 temp에 넣어준다.  if (head == NULL)  {  head = cur;  return head;  }  while ((temp->next) != NULL)  {  temp = temp->next;  }    temp->next = cur;  return head;  }  void printData(LINK head)  {  LINK cur;  printf("노드 : 번호 이름 성적1 성적2 합 \n");  for (cur = head; cur; cur = cur->next)  {  printf("%d번째 : %d %s %d %d %d \n", cur->num, cur->num, cur->name, cur->sc1, cur->sc2, cur->hap);  }  }  void findStudent(char \*name, LINK head)  {  int flag = 0; // 찾았는지 못찾았는지 확인해주는 변수  LINK cur;  for (cur = head; cur; cur = cur->next)  {  if (strcmp(name, cur->name) == 0)  {  printf("%d %s %d %d %d \n", cur->num, cur->name, cur->sc1, cur->sc2, cur->hap);  flag = 1;  }  }  if (flag == 0)  {  printf("찾고자하는 학생의 정보가 없습니다. \n");  }  } |
| <실행결과> |