5645640 제갈건 0324

1. 실행코드

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include <stdlib.h>

#include<malloc.h>

typedef int element;

typedef struct {

int max\_size, size; // max\_size : 리스트의 최대크기, size : 현재 리스트의 개수

element \*pA; // 리스트 포인터

}ArryListType;

void creat(ArryListType\* L) { // 리스트 동적 할당 ADT

printf("어떤 크기의 배열을 만들 것인가요? : ");

scanf("%d", &L->max\_size);

L->pA = (element\*)malloc(sizeof(element) \* L->max\_size);

L->size = 0;

}

void error(const char\* message) { // 에러 출력 ADT

fprintf(stderr, "%s\n", message);

exit(1);

}

void init(ArryListType\* L) { // 생성자

L->size = 0;

}

int is\_empty(ArryListType\* L) { // 리스트가 비었는지를 검사한다.

return L->size == 0;

}

int is\_full(ArryListType\* L) { // 리스트가 꽉 찼는지를 검사한다.

return L->size == L->max\_size;

}

element get\_entry(ArryListType\* L, int pos) { // pos 위치의 요소를 반환한다.

if (pos < 0 || pos >= L->size) {

error("error");

}

return L->pA[pos];

}

void print\_list(ArryListType\* L) { //리스트의 모든 요소를 표시한다.

printf("리스트 출력 : ");

for (int i = 0; i < L->size; i++) printf("%d->", L->pA[i]);

printf("null 종료\n");

printf("\n");

}

void insert\_last(ArryListType\* L, element item) {

if (L->size >= L->max\_size) error("리스트 오버플로우");

L->pA[L->size++] = item;

}

void insert(ArryListType\* L, int pos, element item) {

if (!is\_full(L) && (pos <= L->size)) {

for (int i = (L->size - 1); i >= pos; i--) L->pA[i + 1] = L->pA[i];

L->pA[pos] = item;

L->size++;

}

}

element Delete(ArryListType\* L, int pos) { // pos위치의 요소를 제거한다

element item;

if (pos < 0 || pos >= L->size) error("위치오류");

item = L->pA[pos];

for (int i = pos; i<(L->size - 1); i++) L->pA[i] = L->pA[i + 1];

L->size--;

return item;

}

int main() {

ArryListType\* list = (ArryListType\*)malloc(sizeof(ArryListType)); // 리스트 생성

creat(list); // 리스트의 크기를 입력받아 생성

insert(list, 0, 10);

print\_list(list);

insert(list, 0, 20);

print\_list(list);

insert(list, 0, 30);

print\_list(list);

int a = Delete(list, 0);

print\_list(list);

return 0;

}

1. 실행화면

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 고찰

처음에 정적 배열을 포인터로 바꾸는 것이 어려웠습니다. 무엇을, 왜 바꿔야 하는지는 알고 있었지만 어떻게 바꿔야 하는지 감이 오지 않았습니다. 이후 교수님의 코드를 보고 한 번에 이해가 되었고, 포인터의 사용 이유도 깨달았습니다.

void creat(ArryListType\* L) { // 리스트 동적 할당 ADT

printf("어떤 크기의 배열을 만들 것인가요? : ");

scanf("%d", &L->max\_size);

L->pA = (element\*)malloc(sizeof(element) \* L->max\_size);

L->size = 0;

}

이후 프로그램의 배열을 포인터로 바꾼 후 실행을 하니 정상 작동 하였다.