5645640 제갈건 0328

1. 실행코드

#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\_s("%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) {

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) { // 리스트의 마지막에 item을 추가합니다.

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++;//L->pA[L->size++] = a;//pA[size]에 item을 넣고 size++한다.

}

}

void insert\_new(ArryListType\* L) { // 사용자의 입력을 받아 리스트의 마지막에 Item을 추가한다.

int a;

printf("input an interger number : ");

scanf\_s("%d", &a);

if (!is\_full(L)){

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

//pA[size]에 item을 넣고 size++한다 무조건 마지막에 입력

}

}

void find(ArryListType\* L, int item) {

for (int i = (L->size - 1); i >= 0; i--) {

if (L->pA[i] == item) {

printf("find %d the position %d\n", L->pA[i], i);

system("pause");

exit(0);

}

}

printf("not find");

}

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

element item;

if (pos < 0 || pos >= L->max\_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

for (int i = 0; i <= list->max\_size - 1; i++) { // 순차적으로 데이터 입력

insert\_new(list);

print\_list(list);

}

int a = Delete(list, list->max\_size - 1);// 리스트의 마지막item을 추가한다.

print\_list(list);

find(list, 3);

free(list->pA);

free(list);

return 0;

}

1. 실행화면



1. 고찰

Insert\_new()함수를 만드는 것은 금방 하였습니다. 하지만 find함수를 구현하는 것에 있어서 조금 고민을 하였습니다. 리스트를 순회하는 것에서 위치를 찾는 것을 고민하였지만 이내 pA를 리스트로 쓴다는 것을 기억해 내어 for문과 if문을 사용하여 구현하였습니다.