

자료구조

1차과제 - 배열구현 소스코드

컴퓨터공학과

2019305059

이현수

[배열로 구현 소스코드 - 화면 캡처]

```
1  #include<stdio.h>
2  #include<stdlib.h> //atoi()함수를 사용하기위한 헤더파일
3  #include<stdbool.h> //bool을 사용하기위한 헤더파일
4  #define MAX 100
5
6  typedef struct {
7      int Count; //리스트안에있는 의미있는 값의 개수
8      int Data[MAX]; //정수데이터가 MAX개 저장될 배열
9  }listType;
10
11  //Lptr이 가리키고 있는 리스트를 초기화
12  void Inint(listType* Lptr) {
13      Lptr->Count = 0; //개수를 0으로 초기화
14  }
15
16  //Lptr이 가리키고 있는 리스트가 비어있으면 1, 아니면 0반환
17  bool IsEmpty(listType* Lptr) {
18      if (Lptr->Count == 0) return true; //개수가 0이면 1반환
19      else return false; //그외 0반환
20  }
21
22  //Lptr이 가리키고 있는 리스트에 있는 의미있는 값의 개수 반환
23  int Length(listType* Lptr) {
24      return Lptr->Count; //개수 반환
25  }
26
27  //Lptr이 가리키고 있는 리스트의 Position번째에 Item값 삽입
28  void Insert(listType* Lptr, int Position, int Item) {
29      int i; //반복문에 사용할 변수, 0언어는 변수초기화를 처음에 해야함
30      if (Lptr->Count == MAX) //배열이 꽉 차있을경우
31      {
32          printf("List Full\n");
33      }
34      else if ((Position > (Lptr->Count + 1)) || (Position < 1))
35      { //저장하려는 위치가 이격이 발생하는 공간 혹은 0,음수일 경우
36          printf("Position out of Range\n");
37      }
38      else //그 외
39      { //마지막 데이터부터 삽입위치데이터까지 한칸씩 뒤로 배치
40          for (i = (Lptr->Count - 1); i >= (Position - 1); i--)
41          {
42              Lptr->Data[i + 1] = Lptr->Data[i];
43          } //0,1,2번째가 아닌 1,2,3번째이므로 Position번째는 인덱스상 Position-1이다
44          Lptr->Data[Position - 1] = Item; //Position-1번째 인덱스에 Item저장
45          Lptr->Count += 1; //개수 1추가
46      }
47  }
48
49  //Lptr이 가리키고 있는 리스트의 Position번째 내용을 삭제
50  void Delete(listType* Lptr, int Position) {
51      int i;
52      if (IsEmpty(Lptr)) //리스트가 비어있을경우
53      {
54          printf("list empty\n");
55      }
56      else if ((Position > (Lptr->Count)) || (Position < 1))
57      { //삭제하려는 위치가 빈공간 혹은 0,음수일경우
58          printf("Position out of Range\n");
59      }
60      else//그 외
61      { //삭제할 위치로 뒤에있는 값들을 한칸씩 앞쪽으로 배치
```

```

62     for (i = Position - 1; i < (Lptr->Count - 1); i++) {
63         Lptr->Data[i] = Lptr->Data[i + 1];
64     }
65     Lptr->Count -= 1; //개수 -1
66 }
67 }
68
69 //Lptr이 가리키고 있는 리스트의 Position번째 내용을 Item내용으로 변경
70 void Update(listType* Lptr, int Position, int Item) {
71     if (IsEmpty(Lptr)) //리스트가 비어있을경우
72     {
73         printf("list empty\n");
74     }
75     else if ((Position > (Lptr->Count)) || (Position < 1))
76     { //변경하려는 위치가 빈공간 혹은 0,음수일경우
77         printf("Position out of Range\n");
78     }
79     else {
80         Lptr->Data[Position - 1] = Item; //Position-1 인덱스에 Item 데이터 삽입
81     }
82 }
83
84 //Lptr이 가리키고 있는 리스트의 Position번째 내용을 찾은 후 이를 반환
85 //0,1,2번째가 아닌 1,2,3번째이므로 Position번째는 인덱스상 Position-1이다
86 int Retrieve(listType* Lptr, int Position) {
87     return Lptr->Data[Position - 1]; //Position-1번째 인덱스 데이터 반환
88 }
89
90 int main(void) {
91     listType list; //리스트 생성
92
93     int sel; //기능선택 시 입력받는 변수
94     int em; //리스트가 비었는지 알기위해 함수로부터 숫자를 반환받는 변수
95     int num; //삽입숫자, 삭제숫자, 갱신숫자 입력받는변수, 숫자반환받는변수
96     int loc; //위치를 입력받는 변수
97     char ch; //각 기능수행 후 재수행여부 확인 시 사용
98     char str[10]; // (3)리스트원소등록시 사용자로부터 입력받을 변수
99     while (1) { //무한반복문
100         AAA:printf("\n");
101         printf("(1) 리스트 초기화\n");
102         printf("(2) 리스트 공백여부 확인\n");
103         printf("(3) 리스트 원소 등록\n");
104         printf("(4) 리스트 원소 개수 확인\n");
105         printf("(5) 리스트 원소 삽입\n");
106         printf("(6) 리스트 원소 삭제\n");
107         printf("(7) 리스트 원소 변경\n");
108         printf("(8) 리스트 원소 검색\n");
109         printf("(9) 프로그램 종료\n");
110         printf("--- 기능선택: ");
111         scanf("%d", &sel); //기능을 선택함
112         getchar(); //버퍼제거
113         if (sel == 1) { //(1) 리스트 초기화
114             linit(&list);
115
116             // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능5
117             printf("Continue: ");
118             scanf("%c", &ch); //문자 입력 받음
119             if (ch == 'Y') goto AAA;
120             else break; //'Y'이외 문자 입력시 무한반복문 빠져나감
121         }
122         else if (sel == 2) { //(2) 리스트 공백여부 확인
123             em = IsEmpty(&list); //em에 숫자반환, true=1, false=0
124             if (em == 1) printf("list empty\n"); //1이면 비어있는 리스트
125             else printf("list is not empty\n"); //아니면 비어있지않는 리스트

```

```

// 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
printf("Continue: ");
scanf("%c", &ch); //문자 입력 받음
if (ch == 'Y') goto AAA;
else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}

else if (sel == 3) { //(3) 리스트 원소 등록
//Length(listType *Lptr)함수 이용해 리스트에 저장된 데이터개수 반환
printf("현재 리스트에 %d개 입력됨.\n", Length(&list));
printf("X를 입력하면 원소등록 종료\n");
while (1) {
    scanf("%s", &str); // 'X'가 입력되면 중단해야되서 'X'를 읽기위해 문자열로 입력받음
    getchar(); //버퍼제거
    if (str[0] == 'X') break; // 'X'가 입력되면 while문 빠져나감.
    num = atoi(str); //입력된 문자열을 정수로 자료형을 변환해서 num에 대입
    Insert(&list, Length(&list) + 1, num); //데이터를 마지막에 삽입
}

// 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
printf("Continue: ");
scanf("%c", &ch); //문자 입력 받음
if (ch == 'Y') goto AAA;
else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}

else if (sel == 4) { //(4) 리스트 원소 개수 확인
//Length(listType *Lptr)함수로 저장된 데이터개수 반환
printf("리스트에 %d개 원소가 있습니다.\n", Length(&list));

// 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
printf("Continue: ");
scanf("%c", &ch); //문자 입력 받음
if (ch == 'Y') goto AAA;
else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}

else if (sel == 5) { //(5) 리스트 원소 삽입
printf("몇번째? 삽입할 정수값? ");
scanf("%d %d", &loc, &num); //저장할위치, 저장할데이터를 입력받음
getchar(); //버퍼제거
Insert(&list, loc, num); //loc번째에 num숫자 삽입

// 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
printf("Continue: ");
scanf("%c", &ch); //문자 입력 받음
if (ch == 'Y') goto AAA;
else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}

else if (sel == 6) { //(6) 리스트 원소 삭제
printf("몇번째 원소 삭제? ");
scanf("%d", &loc); getchar(); //삭제할위치 입력받음
Delete(&list, loc); //loc번째 데이터 삭제

// 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
printf("Continue: ");
scanf("%c", &ch); //문자 입력 받음
if (ch == 'Y') goto AAA;
else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}

else if (sel == 7) { //(7) 리스트 원소 변경
printf("몇번째? 변경할 정수값? ");
scanf("%d %d", &loc, &num); getchar(); //위치와 변경할 값 입력받음
Update(&list, loc, num); //loc번째 데이터를 num으로 대체

// 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능

```

```

190     printf("Continue: ");
191     scanf("%c", &ch); //문자 입력 받음
192     if (ch == 'Y') goto AAA;
193     else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
194 }
195 else if (sel == 8) { //(8) 리스트 원소 검색
196     printf("몇번째 원소 검색? ");
197     scanf("%d", &loc); getchar(); //위치입력받음
198     if (IsEmpty(&list)) { //리스트가 비어있을경우
199         printf("list empty\n");
200     }
201     else if ((loc > (Length(&list))) || (loc < 1))
202     { //반환하려는 위치가 빈공간 혹은 0,음수일때
203         printf("Position out of Range\n");
204     }
205     else { //반환위치가 정상일 때
206         num = Retrieve(&list, loc); //loc번째 데이터 num에 반환
207         printf("리스트 %d번째 데이터 : %d\n", loc, num);
208     }
209
210     // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
211     printf("Continue: ");
212     scanf("%c", &ch); //문자 입력 받음
213     if (ch == 'Y') goto AAA;
214     else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
215 }
216 else if (sel == 9) { //(9) 프로그램 종료
217     break; //반복문 빠져나감.
218 }
219 }
220 }

```

[배열로 구현 소스코드 - 소스복사]

```
#include<stdio.h>
#include<stdlib.h> //atoi()함수를 사용하기위한 헤더파일
#include<stdbool.h> //bool을 사용하기위한 헤더파일
#define MAX 100

typedef struct {
    int Count; //리스트안에있는 의미있는 값의 개수
    int Data[MAX]; //정수데이터가 MAX개 저장될 배열
}listType;

//Lptr이 가리키고 있는 리스트를 초기화
void Init(listType* Lptr) {
    Lptr->Count = 0; //개수를 0으로 초기화
}

//Lptr이 가리키고 있는 리스트가 비어있으면 1, 아니면 0반환
bool IsEmpty(listType* Lptr) {
    if (Lptr->Count == 0)return true; //개수가 0이면 1반환
    else return false; //그외 0반환
}

//Lptr이 가리키고 있는 리스트에 있는 의미있는 값의 개수 반환
int Length(listType* Lptr) {
    return Lptr->Count; //개수 반환
}

//Lptr이 가리키고 있는 리스트의 Position번째에 Item값 삽입
void Insert(listType* Lptr, int Position, int Item) {
    int i; //반복문에 사용할 변수, C언어는 변수초기화를 처음에 해야함
    if (Lptr->Count == MAX)//배열이 꽉 차있을경우
    {
        printf("List Full\n");
    }
    else if ((Position > (Lptr->Count + 1)) || (Position < 1))
    { //저장하려는 위치가 이격이 발생하는 공간 혹은 0,음수일 경우
        printf("Position out of Range\n");
    }
    else //그 외
    { //마지막 데이터부터 삽입위치데이터까지 한칸씩 뒤로 배치
        for (i = (Lptr->Count - 1); i >= (Position - 1); i--)
        {
            Lptr->Data[i + 1] = Lptr->Data[i];
        } //0,1,2번째가 아닌 1,2,3번째이므로 Position번째는 인덱스상 Position-1이다
        Lptr->Data[Position - 1] = Item; //Position-1번째 인덱스에 Item저장
        Lptr->Count += 1; //개수 1추가
    }
}

//Lptr이 가리키고 있는 리스트의 Position번째 내용을 삭제
void Delete(listType* Lptr, int Position) {
    int i;
    if (IsEmpty(Lptr)) //리스트가 비어있을경우
    {
        printf("List empty\n");
    }
    else if ((Position > (Lptr->Count)) || (Position < 1))
    { //삭제하려는 위치가 빈공간 혹은 0,음수일경우
        printf("Position out of Range\n");
    }
}
```

```

else//그 외
{
    //삭제할 위치로 뒤에있는 값들을 한칸씩 앞으로 배치
    for (i = Position - 1; i < (Lptr->Count - 1);i++) {
        Lptr->Data[i] = Lptr->Data[i + 1];
    }
    Lptr->Count -= 1; //개수 -1
}
}

//Lptr이 가리키고 있는 리스트의 Position번째 내용을 Item내용으로 변경
void Update(listType* Lptr, int Position, int Item) {
    if (IsEmpty(Lptr)) //리스트가 비어있을경우
    {
        printf("list empty\n");
    }
    else if ((Position > (Lptr->Count)) || (Position < 1))
    {
        //변경하려는 위치가 빈공간 혹은 0,음수일경우
        printf("Position out of Range\n");
    }
    else {
        Lptr->Data[Position - 1] = Item; //Position-1 인덱스에 Item 데이터 삽입
    }
}

//Lptr이 가리키고 있는 리스트의 Position번째 내용을 찾은 후 이를 반환
//0,1,2번째가 아닌 1,2,3번째이므로 Position번째는 인덱스상 Position-1이다
int Retrieve(listType* Lptr, int Position) {
    return Lptr->Data[Position - 1]; //Position-1번째 인덱스 데이터 반환
}

int main(void) {
    listType list; //리스트 생성

    int sel;        //기능선택 시 입력받는 변수
    int em;         //리스트가 비었는지 알기위해 함수로부터 숫자를 반화받는 변수
    int num;        //삽입숫자, 삭제숫자, 갱신숫자 입력받는변수, 숫자반환받는변수
    int loc;        //위치를 입력받는 변수
    char ch;        //각 기능수행 후 재수행여부 확인 시 사용
    char str[10]; //((3)리스트원소등록시 사용자로부터 입력받을 변수
    while (1) { //무한반복문
        AAA:printf("\n");
        printf("(1) 리스트 초기화\n");
        printf("(2) 리스트 공백여부 확인\n");
        printf("(3) 리스트 원소 등록\n");
        printf("(4) 리스트 원소 개수 확인\n");
        printf("(5) 리스트 원소 삽입\n");
        printf("(6) 리스트 원소 삭제\n");
        printf("(7) 리스트 원소 변경\n");
        printf("(8) 리스트 원소 검색\n");
        printf("(9) 프로그램 종료\n");
        printf("---- 기능선택:");
        scanf("%d", &sel); //기능을 선택함
        getchar(); //버퍼제거
        if (sel == 1) { //(1) 리스트 초기화
            linit(&list);

            // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능5
            printf("Continue: ");
            scanf("%c", &ch); //문자 입력 받음
            if (ch == 'Y') goto AAA;
            else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
        }
    }
}

```

```

else if (sel == 2) { //(2) 리스트 공백여부 확인
    em = IsEmpty(&list); //em에 숫자반환, true=1, false=0
    if (em == 1) printf("list empty\n"); //1이면 비어있는 리스트
    else printf("list is not empty\n"); //아니면 비어있지않는 리스트

    // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
    printf("Continue: ");
    scanf("%c", &ch); //문자 입력 받음
    if (ch == 'Y') goto AAA;
    else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}
else if (sel == 3) { //(3) 리스트 원소 등록
    //Length(listType *Lptr) 함수 이용해 리스트에 저장된 데이터개수 반환
    printf("현재 리스트에 %d개 입력됨.\n", Length(&list));
    printf("X를 입력하면 원소등록 종료\n");
    while (1) {
        scanf("%s", &str); // 'X' 입력되면 중단해야되서 'X' 읽기위해 문자열로 입력받음
        getchar(); //버퍼제거
        if (str[0] == 'X') break; // 'X'가 입력되면 while문 빠져나감.
        num = atoi(str); //입력된 문자열을 정수로 자료형을 변환해서 num에 대입
        Insert(&list, Length(&list) + 1, num); //데이터를 마지막에 삽입
    }

    // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
    printf("Continue: ");
    scanf("%c", &ch); //문자 입력 받음
    if (ch == 'Y') goto AAA;
    else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}
else if (sel == 4) { //(4) 리스트 원소 개수 확인
    //Length(listType *Lptr) 함수로 저장된 데이터개수 반환
    printf("리스트에 %d개 원소가 있습니다.\n", Length(&list));

    // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
    printf("Continue: ");
    scanf("%c", &ch); //문자 입력 받음
    if (ch == 'Y') goto AAA;
    else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}
else if (sel == 5) { //(5) 리스트 원소 삽입
    printf("몇번째? 삽입할 정수값? ");
    scanf("%d %d", &loc, &num); //저장할위치, 저장할데이터를 입력받음
    getchar(); //버퍼제거
    Insert(&list, loc, num); //loc번째에 num숫자 삽입

    // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
    printf("Continue: ");
    scanf("%c", &ch); //문자 입력 받음
    if (ch == 'Y') goto AAA;
    else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}
else if (sel == 6) { //(6) 리스트 원소 삭제
    printf("몇번째 원소 삭제? ");
    scanf("%d", &loc); getchar(); //삭제할위치 입력받음
    Delete(&list, loc); //loc번째 데이터 삭제

    // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
    printf("Continue: ");
    scanf("%c", &ch); //문자 입력 받음
    if (ch == 'Y') goto AAA;
    else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
}
}

```



```

else if (sel == 7) { //(7) 리스트 원소 변경
    printf("몇번째? 변경할 정수값? ");
    scanf("%d %d", &loc, &num); getchar();//위치와 변경할 값 입력받음
    Update(&list, loc, num);//loc번째 데이터를 num으로 대체

    // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
    printf("Continue: ");
    scanf("%c", &ch); //문자 입력 받음
    if (ch == 'Y') goto AAA;
    else break; //'Y'이외 문자 입력시 무한반복문 빠져나감
}
else if (sel == 8) { //(8) 리스트 원소 검색
    printf("몇번째 원소 검색? ");
    scanf("%d", &loc); getchar(); //위치입력받음
    if (IsEmpty(&list)) { //리스트가 비어있을경우
        printf("list empty\n");
    }
    else if ((loc > (Length(&list))) || (loc < 1))
    { //반환하려는 위치가 빈공간 혹은 0,음수일때
        printf("Position out of Range\n");
    }
    else { //반환위치가 정상일 때
        num = Retrieve(&list, loc);//loc번째 데이터 num에 반환
        printf("리스트 %d번째 데이터 : %d\n", loc, num);
    }

    // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
    printf("Continue: ");
    scanf("%c", &ch); //문자 입력 받음
    if (ch == 'Y') goto AAA;
    else break; //'Y'이외 문자 입력시 무한반복문 빠져나감
}
else if (sel == 9) { //(9) 프로그램 종료
    break; //반복문 빠져나감.
}
}
}
}

```