

자료구조

1차과제 - 연결리스트구현 소스코드

컴퓨터공학과

2019305059

이현수

[연결리스트 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 nodeRecord {
7      int Data; //노드내부의데이터
8      struct nodeRecord* Next; //노드타입을 가리킴
9  }node;
10
11  typedef node* Nptr;
12
13  typedef struct {
14      int Count; //리스트안에있는 의미있는 값의 개수
15      Nptr Head; //헤드포인터
16  }listType;
17
18  //Lptr이 가리키고 있는 리스트를 초기화
19  void Inint(listType* Lptr) {
20      Lptr->Count = 0; //개수를 0으로 초기화
21      Lptr->Head = NULL; //헤드가 NULL을 가리키게함
22  }
23
24  //Lptr이 가리키고 있는 리스트가 비어있으면 1, 아니면 0반환
25  bool IsEmpty(listType* Lptr) {
26      if (Lptr->Count == 0) return true; //개수가 0이면 1반환
27      else return false; //아니면 0반환
28  }
29
30  //Lptr이 가리키고 있는 리스트에 있는 의미있는 값의 개수 반환
31  int Length(listType* Lptr) {
32      return Lptr->Count; //개수 반환
33  }
34
35  //Lptr이 가리키고 있는 리스트의 Position번째에 Item값 삽입
36  void Insert(listType* Lptr, int Position, int Item) {
37      int i; //c언어에서는 변수선언을 중간에 못함
38      if ((Position > (Lptr->Count + 1)) || (Position < 1))
39      { //Position이 이격발생 혹은 음수, 0일경우
40          printf("Position out of Range\n");
41      }
42      else {
43          Nptr p = (node*)malloc(sizeof(node)); //삽입할 노드 동적할당
44          p->Data = Item; //데이터 삽입
45          p->Next = NULL; //삽입노드의next가 NULL가리킴
46          if (Position == 1) { //첫위치에 삽입할 경우
47              p->Next = Lptr->Head; //삽입node가 현재 첫node 가리킴
48              Lptr->Head = p; //헤드가 삽입노드를 가리킴
49          }
50          else { //첫위치가 아닌경우, 중간혹은 마지막
51              Nptr Temp = Lptr->Head; //Temp가 첫노드 가리킴
52              for (i = 1; i < (Position - 1); i++) {
53                  Temp = Temp->Next; //Temp가 삽입직전 노드를 가리키게
54              }
55              p->Next = Temp->Next; //삽입node의next와 삽입후 뒤에있을 노드와연결
56              Temp->Next = p; //삽입앞노드next를 삽입노드에 연결
```

```

57     }
58     Lptr->Count += 1; //개수 1증가
59 }
60 }
61
62 //Lptr이 가리키고 있는 리스트의 Position번째 내용을 삭제
63 void Delete(listType* Lptr, int Position) {
64     int i;
65     if (IsEmpty(Lptr)) { //리스트가 비어었을 경우
66         printf("list empty\n");
67     }
68     else if ((Position > (Lptr->Count)) || (Position < 1))
69     { //삭제할 위치가 비어있는 공간 혹은 음수, 0일 경우
70         printf("Position out of Range\n");
71     }
72     else {
73         if (Position == 1) { //첫번째 노드 삭제할 경우
74             Nptr p = Lptr->Head; //p가 첫노드가리킴
75             Lptr->Head = Lptr->Head->Next; //헤드가 삭제할노드 다음노드가리킴
76             free(p); //공간해체
77         }
78         else { //그 외
79             Nptr Temp = Lptr->Head; //Temp가 첫노드 가리킴
80             for (i = 1; i < (Position - 1); i++) {
81                 Temp = Temp->Next; //Temp가 삭제할노드 직전노드가리킴
82             }
83             Nptr p = Temp->Next; //p가 삭제할 노드 가리킴
84             Temp->Next = p->Next; //삭제앞노드가 삭제뒤노드와 연결
85             free(p); //공간해체
86         }
87         Lptr->Count -= 1; //개수 1감소
88     }
89 }
90
91 //Lptr이 가리키고 있는 리스트의 Position번째 내용을 Item내용으로 변경
92 void Update(listType* Lptr, int Position, int Item) {
93     int i;
94     if (IsEmpty(Lptr)) { //리스트 비어있는 경우
95         printf("list empty\n");
96     }
97     else if ((Position > (Lptr->Count)) || (Position < 1))
98     { //변경하려는 위치가 빈공간 혹은 음수, 0일 경우
99         printf("Position out of Range\n");
100     }
101     else {
102         Nptr Temp;
103         Temp = Lptr->Head; //노드포인터Temp가 첫노드가리킴
104         for (i = 0; i < Position-1; i++)
105         { //Temp가 Position번째(변경할노드) 노드가리킴
106             Temp = Temp->Next;
107         }
108         Temp->Data = Item; //Temp가 가리키는 노드에 Item으로변경
109     }
110 }
111
112 //Lptr이 가리키고 있는 리스트의 Position번째 내용을 찾은 후 이를 반환
113 int Retrieve(listType* Lptr, int Position) {
114     int i;
115     Nptr Temp;

```

```

116 Temp = Lptr->Head; //노드포인터 Temp가 첫노드가리킴
117 for (i = 0; i < Position - 1; i++)
118 { //Temp가 Position번째(반환할노드) 노드가리킴
119     Temp = Temp->Next;
120 }
121 return Temp->Data; //Temp가 가리키는 노드 Data 반환
122 }
123
124 int main(void) {
125     listType list; //리스트 생성
126
127     int sel; //기능선택 시 입력받는 변수
128     int em; //리스트가 비었는지 알기위해 함수로부터 숫자를 반환받는 변수
129     int num; //삽입숫자, 삭제숫자, 갱신숫자 입력받는변수, 숫자반환받는변수
130     int loc; //위치를 입력받는 변수
131     char ch; //각 기능수행 후 재수행여부 확인 시 사용
132     char str[10]; // (3)리스트원소등록시 사용자로부터 입력받을 변수
133     while (1) { //무한반복문
134         AAA:printf("\n");
135         printf("(1) 리스트 초기화\n");
136         printf("(2) 리스트 공백여부 확인\n");
137         printf("(3) 리스트 원소 등록\n");
138         printf("(4) 리스트 원소 개수 확인\n");
139         printf("(5) 리스트 원소 삽입\n");
140         printf("(6) 리스트 원소 삭제\n");
141         printf("(7) 리스트 원소 변경\n");
142         printf("(8) 리스트 원소 검색\n");
143         printf("(9) 프로그램 종료\n");
144         printf("--- 기능선택: ");
145         scanf("%d", &sel); //기능을 선택함
146         getchar(); //버퍼제거
147         if (sel == 1) { // (1) 리스트 초기화
148             linit(&list);
149
150             // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
151             printf("Continue: ");
152             scanf("%c", &ch); //문자 입력 받음
153             if (ch == 'Y') goto AAA;
154             else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
155         }
156         else if (sel == 2) { // (2) 리스트 공백여부 확인
157             em = IsEmpty(&list); //em에 숫자반환, true=1, false=0
158             if (em == 1) printf("list empty\n"); //1이면 비어있는 리스트
159             else printf("list is not empty\n"); //아니면 비어있지않는 리스트
160
161             // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
162             printf("Continue: ");
163             scanf("%c", &ch); //문자 입력 받음
164             if (ch == 'Y') goto AAA;
165             else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
166         }
167         else if (sel == 3) { // (3) 리스트 원소 등록
168             //Length(listType *Lptr)함수 이용해 리스트에 저장된 데이터개수 반환
169             printf("현재 리스트에 %d개 입력됨.\n", Length(&list));
170             printf("X를 입력하면 원소등록 종료\n");
171             while (1) {
172                 scanf("%s", &str); // 'X'입력되면 중단해야되서 'X'읽기위해 문자열로 입력받음
173                 getchar(); //버퍼제거
174                 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");
    }
}

```

```

234 }
235 else if ((loc > (Length(&list))) || (loc < 1))
236 { //반환하려는 위치가 빈공간 혹은 0,음수일때
237     printf("Position out of Range\n");
238 }
239 else { //반환 위치가 정상일 때
240     num = Retrieve(&list, loc); //loc번째 데이터 num에 반환
241     printf("리스트 %d번째 데이터 : %d\n", loc, num);
242 }
243
244 // 'Y' 입력 시 AAA로 이동해 새로운기능 선택가능
245 printf("Continue: ");
246 scanf("%c", &ch); //문자 입력 받음
247 if (ch == 'Y') goto AAA;
248 else break; // 'Y'이외 문자 입력시 무한반복문 빠져나감
249 }
250 else if (sel == 9) { //(9) 프로그램 종료
251     break; //반복문 빠져나감.
252 }
253 }
254 }

```

[연결리스트 구현 소스코드 - 소스복사]

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

typedef struct nodeRecord {
    int Data; //노드내부의데이터
    struct nodeRecord* Next; //노드타입을 가리킴
}node;

typedef node* Nptr;

typedef struct {
    int Count; //리스트안에있는 의미있는 값의 개수
    Nptr Head; //헤드포인터
}listType;

//Lptr이 가리키고 있는 리스트를 초기화
void Inint(listType* Lptr) {
    Lptr->Count = 0; //개수를 0으로 초기화
    Lptr->Head = NULL; //헤드가 NULL을 가리키게함
}

//Lptr이 가리키고 있는 리스트가 비어있으면 1, 아니면 0반환
bool IsEmpty(listType* Lptr) {
    if (Lptr->Count == 0)return true; //개수가 0이면 1반환
    else return false; //아니면 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; //c언어에서는 변수선언을 중간에 못함
    if ((Position > (Lptr->Count + 1)) || (Position < 1))
    { //Position이 이격발생 혹은 음수,0일경우
        printf("Position out of Range\n");
    }
    else {
        Nptr p = (node*)malloc(sizeof(node)); //삽입할 노드 동적할당
        p->Data = Item; //데이터 삽입
        p->Next = NULL; //삽입노드의next가 NULL가리킴
        if (Position == 1) { //첫위치에 삽입할 경우
            p->Next = Lptr->Head; //삽입node가 현재 첫node 가리킴
            Lptr->Head = p; //헤드가 삽입노드를 가리킴
        }
        else { //첫위치가 아닌경우. 중간혹은 마지막
            Nptr Temp = Lptr->Head; //Temp가 첫노드 가리킴
            for (i = 1; i < (Position - 1); i++) {
                Temp = Temp->Next; //Temp가 삽입직전 노드를 가리키게
            }
            p->Next = Temp->Next; //삽입node의next와 삽입 후 뒤에있을 노드와연결
            Temp->Next = p; //삽입앞노드next를 삽입노드에 연결
        }
        Lptr->Count += 1; //개수 1증가
    }
}
```

```

}

//Lptr이 가리키고 있는 리스트의 Position번째 내용을 삭제
void Delete(listType* Lptr, int Position) {
    int i;
    if (IsEmpty(Lptr)) { //리스트가 비어있을 경우
        printf("list empty\n");
    }
    else if ((Position > (Lptr->Count)) || (Position < 1))
    { //삭제할 위치가 비어있는 공간 혹은 음수, 0일 경우
        printf("Position out of Range\n");
    }
    else {
        if (Position == 1) { //첫번째 노드 삭제할 경우
            Nptr p = Lptr->Head; //p가 첫노드가리킴
            Lptr->Head = Lptr->Head->Next; //헤드가 삭제할 노드 다음노드가리킴
            free(p); //공간해체
        }
        else { //그 외
            Nptr Temp = Lptr->Head; //Temp가 첫노드 가리킴
            for (i = 1; i < (Position - 1); i++) {
                Temp = Temp->Next; //Temp가 삭제할 노드 직전노드가리킴
            }
            Nptr p = Temp->Next; //p가 삭제할 노드 가리킴
            Temp->Next = p->Next; //삭제 앞노드가 삭제 뒤노드와 연결
            free(p); //공간해체
        }
        Lptr->Count -= 1; //개수 1감소
    }
}

//Lptr이 가리키고 있는 리스트의 Position번째 내용을 Item내용으로 변경
void Update(listType* Lptr, int Position, int Item) {
    int i;
    if (IsEmpty(Lptr)) { //리스트 비어있는 경우
        printf("list empty\n");
    }
    else if ((Position > (Lptr->Count)) || (Position < 1))
    { //변경하려는 위치가 빈공간 혹은 음수, 0일 경우
        printf("Position out of Range\n");
    }
    else {
        Nptr Temp;
        Temp = Lptr->Head; //노드포인터 Temp가 첫노드가리킴
        for (i = 0; i < Position-1; i++)
        { //Temp가 Position번째(변경할노드) 노드가리킴
            Temp = Temp->Next;
        }
        Temp->Data = Item; //Temp가 가리키는 노드에 Item으로 변경
    }
}

//Lptr이 가리키고 있는 리스트의 Position번째 내용을 찾은 후 이를 반환
int Retrieve(listType* Lptr, int Position) {
    int i;
    Nptr Temp;
    Temp = Lptr->Head; //노드포인터 Temp가 첫노드가리킴
    for (i = 0; i < Position - 1; i++)
    { //Temp가 Position번째(반환할노드) 노드가리킴
        Temp = Temp->Next;
    }
    return Temp->Data; //Temp가 가리키는 노드 Data 반환
}

```



```
}
```

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

    int sel;        //기능선택 시 입력받는 변수
    int em;         //리스트가 비었는지 알기위해 함수로부터 숫자를 반환받는 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 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 == 2) { //(2) 리스트 공백여부 확인
                em = lsempty(&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 비어있지않은 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 { //반환 위치가 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.
}
}
}
```