

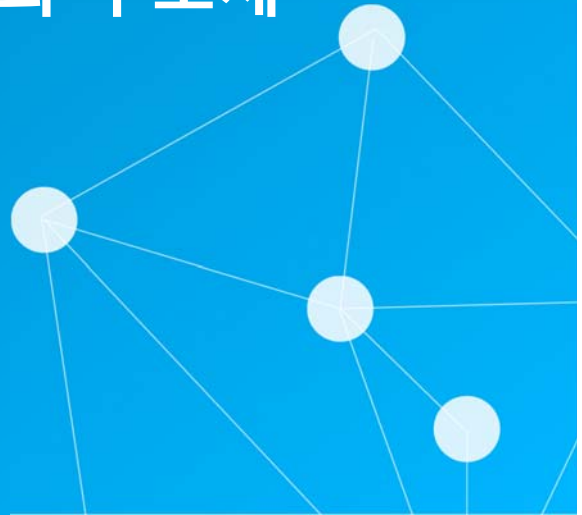
- 배열의 개념을 이해한다.
- 함수 호출에서 배열의 전달 방법을 이해한다.
- 구조체의 개념과 활용 방법을 이해한다.
- 배열과 구조체를 이용해 문제를 해결하는 능력을 배양한다.

02

CHAPTER

배열과 구조체

- 2.1 대용량 자료의 처리
- 2.2 배열
- 2.3 구조체
- 2.4 배열과 구조체의 응용: 다항식 프로그램



2.1 많은 자료의 처리?



- 배열(array), 구조체(struct)
 - 성적 처리 프로그램에서 45명의 성적을 저장하는 방법
 - 주소록 프로그램에서 친구들의 다양한 정보(이름, 전화번호, 주소, 이메일 등)를 통합하여 저장하는 방법



반 학생들의 성적 처리

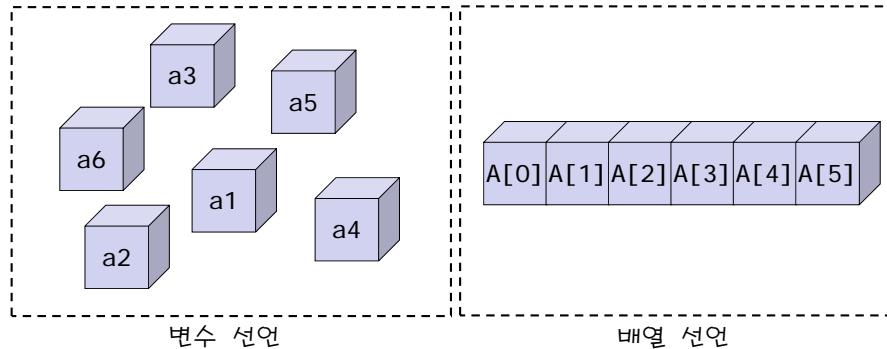


친구 주소록 만들기

2.2 배열



- 같은 형의 변수를 여러 개 만드는 경우에 사용
 - 여러 개의 변수 선언: `int A0, A1, A2, A3, ..., A5;`
 - 하나의 배열 선언: `int A[6];`



- 만약 배열이 없다면?
 - 반복문을 사용할 수 없다!

배열의 특징

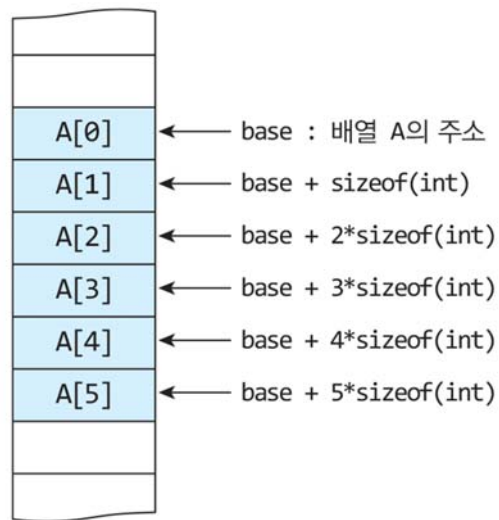


- 배열: <인덱스, 요소> 쌍의 집합
 - 인덱스(index)가 주어지면 해당되는 요소(element)가 대응되는 구조
- 직접 접근(direct access) 방식
 - 항목 접근의 시간 복잡도가 $O(1)$
 - 연결 리스트(5장)
 - 순차 접근(sequential access) 방식
 - 항목 접근의 시간 복잡도 $O(n)$
- 벡터?
 - C++의 STL에서 vector 제공
 - 배열과 벡터의 차이는?

1차원 배열



- 자료형 배열이름[배열의_크기];
- `int A[6];`



- `float a[100]`으로 선언된 배열의 시작 주소를 1000번지라고 할 때, 배열의 10번째 요소의 주소는 몇 번지인가?



배열의 복사

- 변수의 복사와 배열의 복사

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int A[5] = { 10, 20, 30 };
```

```
    int B[5], i, x = 2018, y = 0;
```

```
    y = x;
```

```
    for (i = 0; i < 5; i++)
```

```
        B[i] = A[i];
```

```
    printf("변수 복사 결과: x=%d, y=%d\n", x, y);
```

```
    printf("배열 복사 결과: \n");
```

```
    for (i = 0; i < 5; i++) {
```

```
        printf("A[%d] = %d\t", i, A[i]);
```

```
        printf("B[%d] = %d\n", i, B[i]);
```

```
    }
```

```
}
```



문자열 : 특별한 1차원 배열

- `char s[12] = "game over";`

	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]	s[11]
s	'g'	'a'	'm'	'e'	' '	'o'	'v'	'e'	'r'	'\0'		

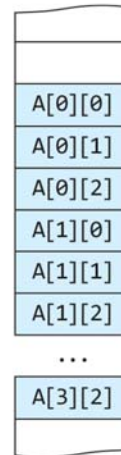
- 문자열 처리
 - 문자열의 복사나 비교를 위해 `=`나 `==` 또는 `<` 등의 연산자를 사용할 수 없다.
 - `strcmp()`, `strcpy()`, ...
 - `<string.h>` 포함

2차원 배열

- 자료형 배열이름[행의_크기][열의_크기];
 - int A[4][3];
 - int A[4][3]= { {1,2,3}, {4,5,6}, {7,8,9}, {10,11,12} };

A[0][0]	A[0][1]	A[0][2]
A[1][0]	A[1][1]	A[1][2]
A[2][0]	A[2][1]	A[2][2]
A[3][0]	A[3][1]	A[3][2]

(a) 2차원 배열



(b) 실제 메모리 안에서의 위치

- int a[5][6]과 같이 정의된 2차원 배열에서 시작 주소를 base라고 하면 a[3][2] 요소의 주소는?



- C언어에서의 배열에 대하여 다음 중 맞는 것은?
 1. 3차원 이상의 배열은 불가능하다.
 2. 배열의 이름은 포인터와 같은 역할을 한다.
 3. 배열의 인덱스는 1에서부터 시작한다.
 4. 선언한 다음, 실행 도중에 배열의 크기를 변경하는 것이 가능하다.



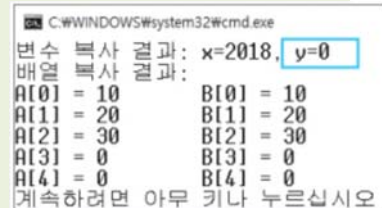
- 다음 중 배열에 관한 문장 중 문법에 맞지 않는 것은?
 1. `int a[] = {1, 2, 3, 4};`
 2. `char b[10], c[20];`
 3. `char **pa[10];`
 4. `char *pb[30] = "I am a student";`

함수의 매개변수로서의 배열

- 변수의 전달 → 값을 복사 (call by value)
- 배열의 전달 → 첫 번째 항목의 주소를 전달(주소를 복사)

```
void copy_array(int a[], int b[], int len) {  
    int i;  
    for (i = 0; i < len; i++)  
        b[i] = a[i];  
}  
void copy_variable(int a, int b) {  
    b = a;  
}  
...
```

```
int A[5] = { 10, 20, 30 };  
int B[5], i, x = 2018, y = 0;  
copy_variable(x, y);  
copy_array(A, B, 5);
```



C:\WINDOWS\system32\cmd.exe
변수 복사 결과: x=2018, y=0
배열 복사 결과:
A[0] = 10 B[0] = 10
A[1] = 20 B[1] = 20
A[2] = 30 B[2] = 30
A[3] = 0 B[3] = 0
A[4] = 0 B[4] = 0
계속하려면 아무 키나 누르십시오

13

배열에서의 주의사항

- 매개 변수로 배열의 길이도 전달해야 함.
int findMaxValue(int a[], int len)
int arr[10] = {3, ...};
int maxVal = findMaxValue(arr, 10);
- 2차원 이상의 다차원 배열의 매개 변수 전달에 조심.
– int findMaxPixel(int a[][5], int h, int w)

14

실습

- 크기가 n 인 배열 array에서 임의의 위치 loc에 정수 value를 삽입하는 함수 insert()를 작성하라. 정수가 삽입되면 그 뒤에 있는 정수들은 한 칸씩 뒤로 밀려야 한다. 현재 배열에 들어있는 원소의 개수는 items개라고 하자 (items는 n 에 비해 충분히 작다고 가정한다).

```
void insert (int array[], int loc, int value) {
```

```
...
```

```
}
```

- 크기가 n 인 배열 array에서 임의의 위치 loc에 있는 정수를 삭제하는 함수 delete()를 작성하라. 정수가 삭제되면 그 뒤에 있는 정수들은 한 칸씩 앞으로 이동하여야 한다. 현재 배열에 들어있는 원소의 개수는 items개라고 하자 (items는 n 에 비해 충분히 작다고 가정한다).

```
void delete (int array[], int loc) {
```

```
...
```

```
}
```

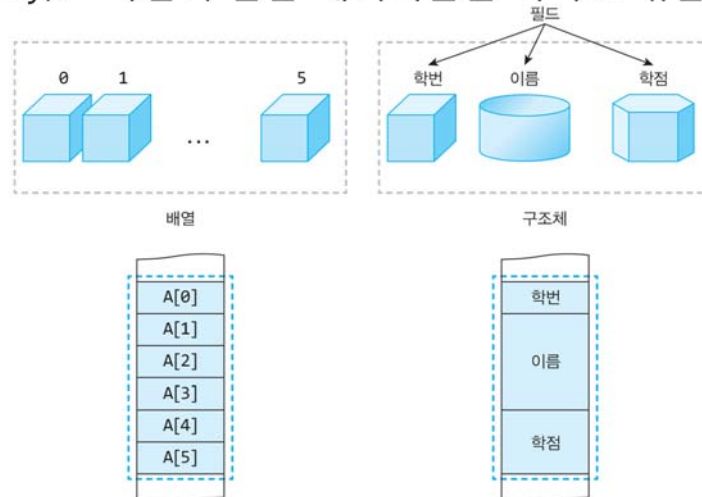
- 구현한 insert(), delete() 함수의 시간 복잡도를 구하여라.

15

2.3 구조체

- 기존의 자료형들을 조합해 새로운 자료형을 만드는 방법
- 배열과의 차이

- 구조체(structure): 타입이 다른 데이터를 하나로 묶음
- 배열(array): 타입이 같은 데이터들을 하나로 묶음



16

구조체의 정의와 선언



- 정의

```
struct Student {  
    int id;  
    char name[20];  
    double score;  
};
```

```
typedef struct Student_t {  
    int id;  
    char name[20];  
    double score;  
} Student;
```

- 선언

```
struct Student a;
```

```
Student a;
```

```
Student a = { 201803156, "홍길동", 96.3 };
```

- 멤버 접근: 항목 연산자(membership operator) '.'

```
a.id = 30830;  
a.score = 92.3;  
strcpy(a.name, "Jinyoung");  
// a.name = "Jinyoung";은 오류 발생
```

17

구조체와 연산자



- 대입 연산자만 가능

```
int x, y=10;  
Student a, b={ 201803156, "홍길동", 96.3 };  
x = y;           // OK: int 변수의 복사  
a = b;         // OK: 구조체 변수의 복사
```

- 다른 연산자 사용 불가

```
if( a > b )      // 오류: 구조체의 비교연산 불가  
    a += b;      // 오류: 구조체의 다른 대입 연산도 불가  
  
int compare(Student a, Student b) {  
    return a.id - b.id;  
}
```

18

구조체와 함수



- 함수의 매개 변수나 반환형으로 사용할 수 있음.
 - Call by value
- 다음 함수의 동작은?

```
void print_complex(Complex c) {  
    printf("%4.1f + %4.1fi\n", c.real, c.imag);  
}  
void reset_complex(Complex c) {  
    c.real = c.imag = 0.0;  
}
```



19

- 구조체에 관한 내용 중 틀린 것은?
 1. 구조체의 배열이 가능하다.
 2. 구조체 변수에 대하여 비교 연산이나 산술 연산을 할 수 있다.
 3. 구조체 안에 배열을 넣을 수 있다.
 4. 서로 다른 데이터 형도 묶을 수 있다.



2.4 배열과 구조체의 응용 : 다항식

- 다항식의 일반적인 형태

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

- 처리를 위한 다항식의 자료구조가 필요
- 어떤 자료구조가 다항식의 연산을 편리하게 할까?

- 다항식을 위한 자료구조?

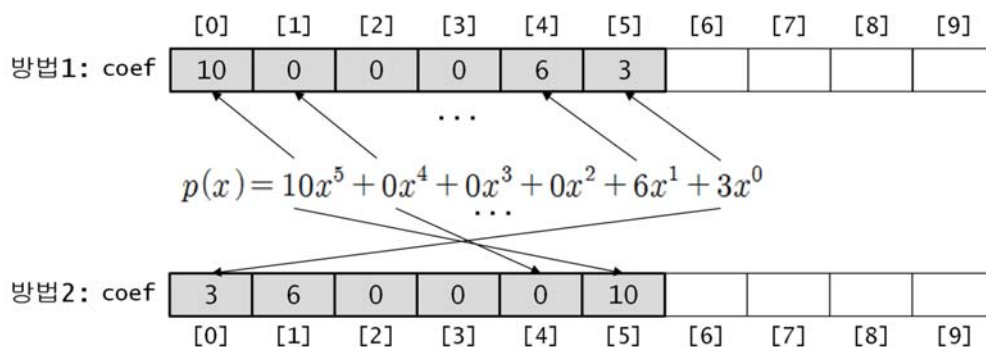
- 배열을 사용하는 방법

- 모든 항의 계수를 배열에 저장
- 다항식의 0이 아닌 항만을 배열에 저장: **희소 다항식**

21

다항식 구조체

```
#define MAX_DEGREE    101    // 다항식의 최고 차수 + 1
typedef struct {
    int degree;
    float coef[MAX_DEGREE];
} Polynomial ;
```



22

다항식 입출력 연산(방법 1)



```
void print_poly(Polynomial p, char str[])
{
    int i;
    printf("\t%s", str);
    for( i=0 ; i<p.degree ; i++)
        printf("%5.1f x^%d + ", p.coef[i], p.degree-i);
    printf(" %4.1f\n", p.coef[p.degree] );
}
```

```
Polynomial read_poly()
{
    int i;
    Polynomial p;
    printf("다항식의 최고 차수를 입력하시오: ");
    scanf( "%d", &p.degree );
    printf("각 항의 계수를 입력하시오 (총 %d개): ", p.degree+1);
    for( i=0 ; i<=p.degree ; i++)
        scanf( "%f", p.coef+i );
    return p;
}
```

23

다항식의 덧셈 연산



- 다항식 덧셈 알고리즘?
- 단순화 방법? $c=a+b \rightarrow c = a; c += b;$

```
Polynomial add_poly(Polynomial a, Polynomial b)
{
    int i;
    Polynomial p;
    if (a.degree > b.degree) {
        p = a;
        for( i=0 ; i<=b.degree ; i++ )
            p.coef[i+(p.degree-b.degree)] += b.coef[i];
    }
    else {
        p = b;
        for( i=0 ; i<=a.degree ; i++ )
            p.coef[i+(p.degree-a.degree)] += a.coef[i];
    }
    return p;
}
```

24

다항식 프로그램



```
void main()
{
    Polynomial a, b, c;
    a = read_poly();
    b = read_poly();
    c = add_poly(a, b);
    print_poly(a, " A = ");
    print_poly(b, " B = ");
    print_poly(c, "A+B= ");
}
```

```
C:\WINDOWS\system32\cmd.exe
다항식의 최고 차수를 입력하시오: 3
각 항의 계수를 입력하시오 (총 4개): 1 2 3 4
다항식의 최고 차수를 입력하시오: 5
각 항의 계수를 입력하시오 (총 6개): 1 2 3 4 5 6
A = 1.0 x^3 + 2.0 x^2 + 3.0 x^1 + 4.0
B = 1.0 x^5 + 2.0 x^4 + 3.0 x^3 + 4.0 x^2 + 5.0 x^1 + 6.0
A+B= 1.0 x^5 + 2.0 x^4 + 4.0 x^3 + 6.0 x^2 + 8.0 x^1 + 10.0
계속하려면 아무 키나 누르십시오 . . .
```

25

실습



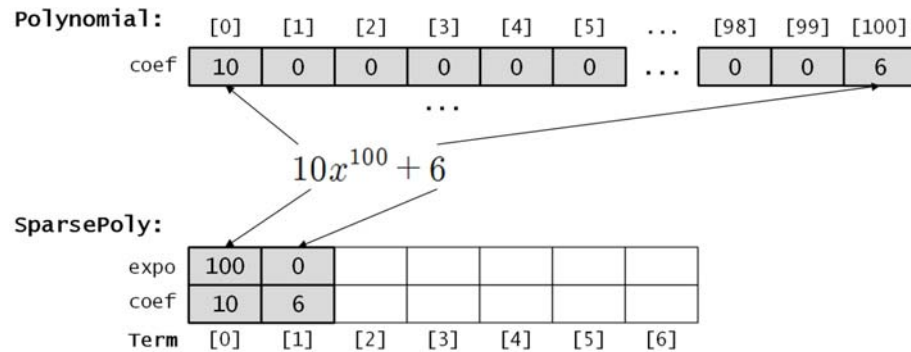
- 다음 연산들을 프로그램 2.7에 추가하라.
- 1. 두 다항식의 뺄셈 ($c=a-b$)을 구하는 함수 `sub_poly()`를 구현하라.
`Polynomial sub_poly(Polynomial a, Polynomial b)`
- 2. 두 다항식의 곱셈 ($c=a*b$)을 구하는 함수 `mult_poly()`를 구현하라.
`Polynomial mult_poly(Polynomial a, Polynomial b)`
- 3. 다항식의 연산 결과 최고차항의 계수가 0으로 변할 수 있다. 다항식의 계수를 분석해 최고차항의 계수가 0이 아닌 값이 나오도록 다항식의 멤버 값들을 변경하는 함수 `trim_poly()`를 구현하라.
`void trimPoly (Polynomial * p);`
- 4. 다항식의 출력함수 `print_poly()`를 수정하여 계수가 0인 항은 출력되지 않도록 변경하라. 또한 계수가 1인 경우에는 1이 출력되지 않도록 변경하라.
`void print_poly(Polynomial p, char str[]);`

26

희소 다항식의 표현

- 희소 다항식(Sparse Polynomial) 이란?

- 대부분 항의 계수가 0인 다항식



```
typedef struct {  
    int      nTerms;  
    Term     term[MAX_TERMS];  
} SparsePoly;
```

```
typedef struct {  
    int      expon;  
    float    coef;  
} Term;
```

27

실습

- 희소 다항식을 위한 구조체와 관련 함수를 구현하라.
각 항을 나타내는 Term 구조체와 희소 다항식을 위한 구조체 SparsePoly를 설계하고 관련 함수를 구현하라.
SparsePoly read_spoly();
SparsePoly add_spoly(SparsePoly a, SparsePoly b);
void printSPoly (SparsePoly p, char str[]);
- 희소행렬 표현방법 중 두번째 방법(0이 아닌 요소들만 저장하는 방법)으로 주어진 행렬의 전치 행렬을 구하는 연산인 transpose()를 구현하라.
 - 행번호와 열번호가 순서대로 저장되게 구현
 - 전치 전과 후를 출력하라.

28