구조체

레코드

하나의 객체를 활용하기 위하여 정의된 여러 타입의 객체 속성으로 구성된 자료 구조

◆ 구조체의 개념

구조체 : 여러 형의 데이터를 하나의 객체로 선언하여 사용함

레코드 자료구조를 구현함

```
typedef struct member {
        int id;
        char name[20];
        float score;
} example;
```

에 의하여 struct member 라는 구조체 형이 선언됨
동시에 typedef 에 의하여 example이라는
데이터 형을 정의함

```
struct member {
    int id;
    char name[20];
    float score;
};
typedef struct member example;
```

구조체

◆ 구조체의 활용

example onep, exmember[20];

와 같은 선언문에 의하여 구조체를 프로그램에서 변수로 사용할 수 있게 됨.

```
onep.id = 2102;
strcpy(onep.name, “Hong Gildong” );
onep.score = 4.23;
```

// exmember중에 성적이 4.00이상인 학생의 id를 출력하시오.

```
for (k=0; k < 20; k++)
        if (exmember[k].score >= 4.0)
            printf( “%d\n” , exmember[k].id);
```

구조체 활용 실습

```
typedef struct member {
    int id;
    char name[20];
    float score;
} example;


void  over40(example *a, int n);
example  find_max(example *a, int n);
```

```
main()
{
    example onep, exmember[20];
    int i, n;

    printf( “Input the number of data :  “);
    scanf( “%d” , &n);

    for (i=0; i < n; i++) {
        printf("데이터 입력 : ");
        scanf("%d  %s  %f",
        &exmember[i].id, exmember[i].name,
        &exmember[i].score);
    }
    over40(exmember, n);
    onep = find_max(exmember, n);

    printf("Information of the highest score student
    = %d : %s : %.2f\n",
   onep.id, onep.name, onep.score);
}
```

구조체 활용 실습

```
void over40(example a[], int n)
{
    int k;

    printf("List of high score
            students(over 4.0)\n");
    for (k=0; k < n; k++)
        if (a[k].score > 4.0)
            printf("%d -- %s\n",
            a[k].id, a[k].name);
}
```

k	id	name	score
0	121	Park	4.01
1	170	Lee1	3.98
2	200	Kim1	3.33
3	100	Lee2	4.32
4	300	Kim2	4.11
5	210	Lim	3.88

```
example find_max(example a[], int n)
{
    int k, max_index;

    max_index = 0;

    for (k=1; k < n; k++)
        if (a[k].score > a[max_index].score)
            max_index = k;

    return a[max_index];
}
```

파일과 구조체 정의

다음의 예제 입력파일과 같이 준비된 수강신청 데이터를 활용한 예제 프로그램을 작성해 보면서 파일의 데이터를 구조체로 읽어 활용하는 예제를 살펴봅니다

[입력파일 예]

강좌코드	담당교수이름	수강생수	강의실번호
db01	YSKIM	37	1217
dmath	DHLEE	42	2301
arch01	HJCHOI	33	1108
ds01	HSRHEE	45	1307
db02	HJCHO	28	2311
algo01	HSRHEE	35	1305
ds02	HSRHEE	37	1307
arch02	HJCHOI	36	1108
algo02	HSRHEE	43	1305
network	HJYOON	22	2200
web	YSKIM	43	2100

```
typedef struct course {
    char courseid[10];
    char tname[20];
    unsigned snum;
    unsigned roomnum;
} cinform;
```

1. 수강인원이 40명이상인 강좌코드를 출력하시오

2. 교수이름이 HSRHEE인 교수가 강의하는 강좌코드와 강의실번호를 출력하시오

3. 강의실 번호 1108 에서 강의하는 강좌코드와 담당교수를 출력하시오

스택의 정의 및 구조

◆ 스택(stack)

- 선형리스트의 끝부분에서만 데이터의 입력과 출력이 가능한 자료구조

- 마지막에 삽입(Last-In)한 원소는 맨 위에 쌓여 있다가
  가장 먼저 삭제(First-Out) 되는

  ☞ **후입선출 구조** (LIFO, Last-In-First-Out)

- 스택을 운영하기 위하여 끝 부분에 대한 정보가 필요함.

  ☞ **top**

스택에서의 데이터 처리과정

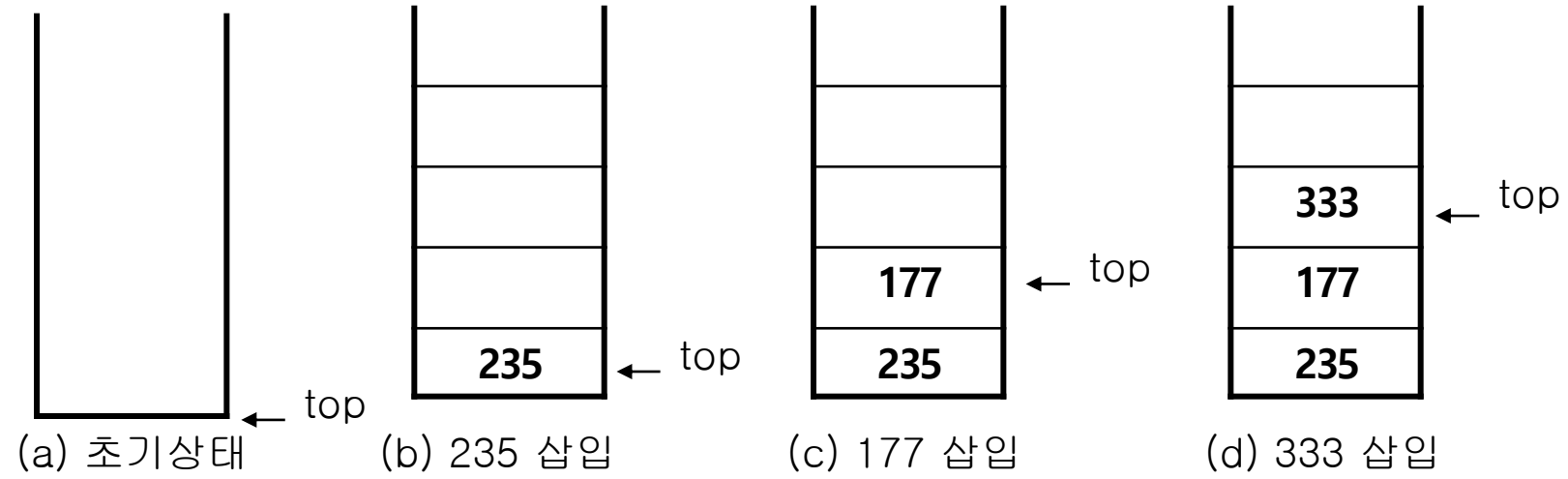
◆ 스택(stack)의 구현

- 배열에 스택 기능을 구현하는 경우

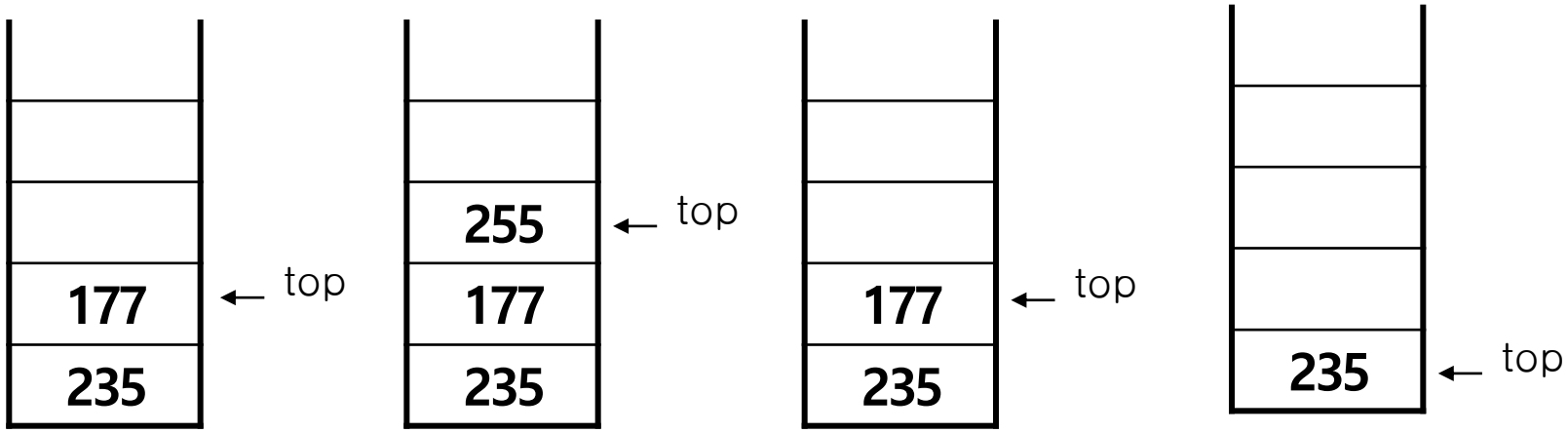
  top을 배열의 첨자로 사용하고 스택안의 데이터가 int 인 경우

  top = -1로 초기화

  int stack[MAX_STACK_SIZE];



스택에서의 데이터 처리과정



(e) 삭제 (f) 255 삽입 (g) 삭제 (h) 삭제

스택에 데이터 삽입

◆ 데이터 삽입 함수 push

```
void push(int item)
{
     /* 전역변수 stack에 item을 삽입, top도 또한 전역변수 */

    if (top >= MAX_STACK_SIZE -1)
        printf("Stack is overflow !!!\n");
    else {
        top++;
        stack[top] = item;
    }
}
```

스택으로부터 데이터 삭제

◆ 스택으로부터 데이터 삭제 함수 pop

```
int pop()
{
    /* stack의 top이 가리키는 데이터를 리턴해 준다  */

    if (top == -1)
        printf("Stack is empty!!!");
    else return stack[top--];
}
```

```
else  {
    item = stack[top];
    top--;
    return item;
}
```

스택 연산 활용 실습

```
typedef struct {
    int key;
    char grade;
} element;

int top = -1;
element stack[MAX_STACK_SIZE];
void push(element data);
element pop();
```

100	B
150	A
200	C
250	B
300	A
350	C

```
main()
{
    element data;
    int i, n, cond = 1;
    i = 0;
    while (cond) {
        printf("데이터 입력 : ");
        scanf("%d %c",
                &data.key, &data.grade);
        if (data.key != 0) {
            push(data);
            i++;
        }
        printf( “스택에 데이터를 계속
                 입력하실래요?(1/0)");
        scanf("%d” , &cond);
    }
    printf( “입력한 데이터의 개수 : %d\n” , i);
```

스택 연산 활용 실습

```
    printf("스택에서 몇개의 데이터가 필요하세요 ? ");
    scanf("%d", &n);
    printf("스택에서 삭제한 데이터 : \n");
    for (i=0; i < n; i++) {
        data = pop();
        printf("%d\t%c\n", data.key, data.grade);
    }
}
```

```
100   B
150   A
200   C
250   B
300   A
350   C
```

스택의 응용 예

(1) 부프로그램의 호출과 복귀

```
main()          f1()            f2()            f3()
{               {               {               {
    f1()            f2()            f3()
R0:             R1:             R2:

}               }               }               }
```

스택의 응용 예

(2) 스택을 사용하여 입력된 중위표기식을 후위표기식으로 변환

• 변환 알고리즘(괄호 없는 수식의 경우)

1) 초기상태의 빈 스택에는 무조건 연산자를 스택에 push한다.

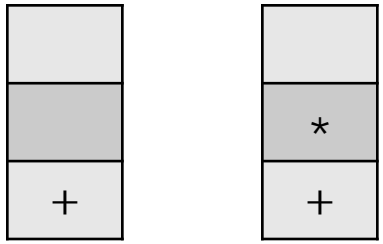
2) 피연산자를 만나면 출력한다.

3) If 들어오는 연산자의 우선순위 > 스택의 top연산자 우선순위
   then 연산자를 스택에 push한다.
   else top의 연산자 pop하여 출력하고 새로운 스택의 연산자와 비교한다.

4) 식을 다 읽으면 스택의 연산자를 pop하여 출력한다.

수식의 후위표기법 변환

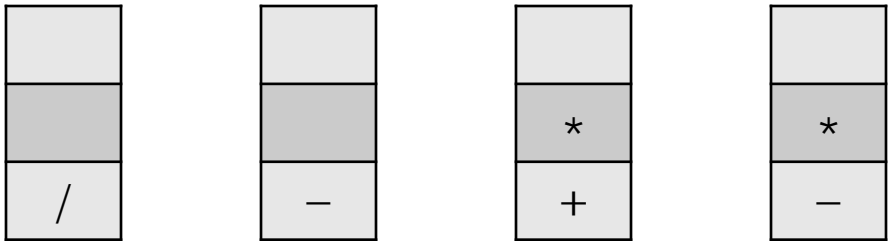
◆ 중위 표기식 a+b*c 의 후위 표기식으로의 변환



a b c * +

수식의 후위표기법 변환

◆ 중위 표기식 a/b-c+d*e-a*c의 후위 표기식으로의 변환



a b / c - d e * + a c * -

스택의 응용 예

(3) 후위표기식의 연산

- 연산 방법

1) 피연산자를 만나면 스택에 push 한다.

2) 연산자를 만나면 필요한 만큼의 피연산자를 스택에서 pop하여 연산하고,
   연산결과를 다시 스택에 push 한다.

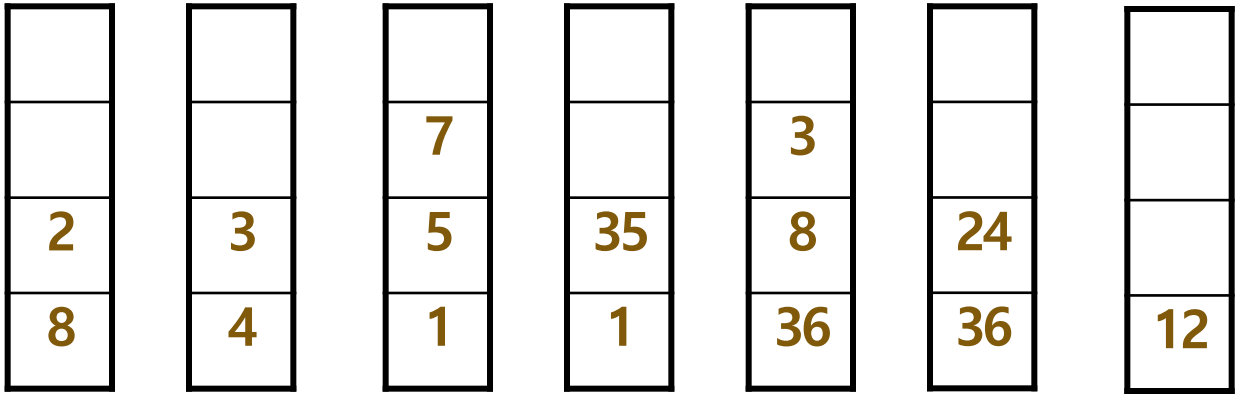
3) 1)-2)를 반복하여 수식이 끝나면,
   마지막으로 스택을 pop하여 출력한다.

수식의 후위표기법 변환과 계산의 예

◆ 중위 표기식 a/b-c+d*e-a*c의 후위 표기식으로의 변환
a(8), b(2), c(3), d(5), e(7)인 경우

a b / c - d e * + a c * -

8 2 / 3 - 5 7 * + 8 3 * -



스택에 데이터 삽입

스택을 활용하여 다음 수식(중위표기식)을 후위표기식으로 바꾸고 그 값을 계산하는 과정에서 스택의 변화를 그리시오.

8 - 9 / 3 + 6 – 3 * 4