

프로그래밍 언어론

Assignment #5

전기컴퓨터공학부 정보컴퓨터공학전공

201524582 정희석

1. Give the accessing formula for computing the location of component $A[I, J]$ of a matrix A declared as:

$A: \text{array}[LB1:UB1, LB2:UB2]$

where A is stored in column-major order.

A.

$A: \text{array}[LB1:UB1, LB2:UB2] \rightarrow \text{array 는 } m \times n \text{ array 이다}$ $m = UB1 - LB1 + 1$ 이다.
 $n = UB2 - LB2 + 1$ 이다.

$A[I, J]$ 의 Location!

ex) $I=2, J=3$

$A + 3m + 2$

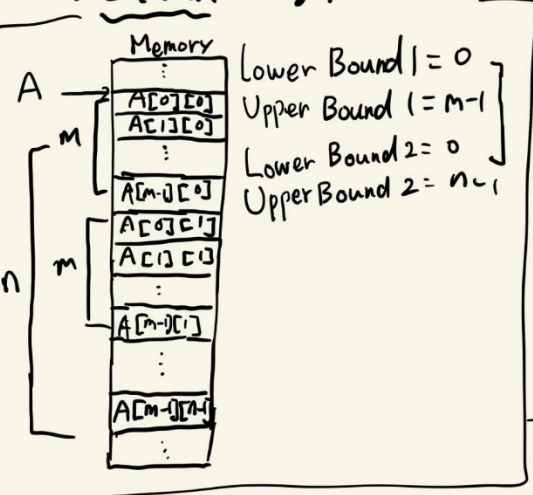
$= A + (J(UB1+1) + I) \rightarrow \text{인 경우, } LB1, LB2 = 0$

ex) $LB1 = 0$ $UB1 = 14$ $LB2 = 0$ $UB2 = 10$ $A[10, 8]$ $L_{address}$

$= A + 8 \times 15 + 10$

$\rightarrow \text{address formula}$

$\Rightarrow A + [(I - LB1) + (J - LB2) \times (UB1 - LB1 + 1)] \times \text{sizeof(element)}$



Address = $A + [(I - LB1) + (J - LB2) \times (UB1 - LB1 + 1)] \times \text{sizeof(element)}$

$LB1, LB2 = 0$ 이면 Address = $A + \{I + J \times (UB1 + 1)\} \times \text{sizeof(element)}$

$M \times N$ array 이면 Address = $A + (I + J \times M) \times \text{sizeof(element)}$

2. For a language that allows variant records without tag fields (free-union types) such as Pascal, write a procedure

```
procedure GIGO (I: integer; var R: real;)
```

that uses a record with two variants whose only purpose is to compromise the type-checking system. GIGO takes an argument I of type integer and returns the same bit pattern as a result R of type real without actually converting the value of I to a real number.

- C 언어로 작성하되 float 를 int 로 변환하는 형태로 작성한다.

A. GIGO => Integer 를 parameter 로 받아서 Float 를 출력하는 함수. 이때 bit pattern 은 바뀌지 않는 procedure 이다. 이를 C 언어에서 같은 작업을 하지만 float 를 받아서 int 로 변환하도록 하는 함수를 작성. => UNION 을 사용

```
union Data{
    int d;
    float f;
};
int GIGO(float real){ //garbage in garbage out
    union Data convert; //이때 Data 는 어떤 형식인지 정해지지 않는다.
    convert.f = real;
    return convert.d;
}
```

Test) 15.34772F 를 넣고 돌렸을 때 bit pattern 은 01000001 01110101 10010000 01000011 이었고 이를 decimal 로 환산했을 때 1098223683 이 나왔으며 이를 printf 로 확인한 결과 모두 나왔다.

3. Assume that language BL includes a stack data structure and three operations:

NewTop(S, E), which adds the element E to the top of stack S,

PopTop(S), which deletes the top element of stack S,

GetTop(S), which returns a pointer to the location in stack S of the current top element.

What is wrong with the design of these three operations? How could they be redefined to correct the problem?

A. BL 언어는 stack data 구조와 3 개의 operation 을 가지고 있다고 가정한다.

-> 먼저 stack 구조는 LIFO(Last in First Out)형식으로 Top 과 Bottom 에서 먼저 들어올수록 Bottom 으로, element 를 뺄 때는 Top 에서 빼는 방식이다.

=> <operation 분석>

1) NewTop operation 에서 element 를 stack 의 top 에 추가하는 것은 좋으나 이전 Top 과 새로운 Top 간의 연결이 구현, 그리고 stack 의 크기에 관한 구현이 동반되어야 한다. (stack 이 linked-list 기반이면 pointer 를 추가, array 기반이면 모든 element 를 한 index 씩 뒤로 미루는 것과 같은 동작이 수반되어야 함.) 그리고 array 기반에서는 stack 이 가득 찼을 때의 이 operation 을 어떻게 수행해야 할 지 역시 구현되어야 한다. (그리고 NewTop 보다 PushTop 으로 이름을 바꾸는 것이 좋다고 생각한다.)

2) PopTop operation 에서 element 를 stack 의 top 에서 제거하는 것은 좋으나 그 다음 element 를 Top 으로 만드는 구현이 동반되어야 한다. (linked-list 기반이면 pointer 연결, array 기반이면 index 조정 동작이 수반되어야 함.) 그리고 stack 이 empty 상태 일 때 PopTop operation 을 동작했을 때의 error 를 출력하는 동작이나 아무런 동작이 없는 것과 같은 예외 처리가 구현되어야 한다.

3) GetTop operation 에서 Top 의 pointer 를 반환하는 것도 좋지만 Top 의 element 를 반환하는 것이 좋다고 생각한다. 역시 stack 이 empty 일 때의 예외처리 역시 구현되어야 한다.

4) 위의 operation 에 추가로 stack 을 생성, 제거, 상태, element 탐색, 저장 개수 출력 등 추가적인 operation 이 구현하는 것도 좋다고 생각한다.

4. Pascal 의 discriminated variant record 의 문제점을 정리하고, 이 문제를 해결하기 위한 방법을 제시하라.

A. discriminated variant record 는 record 에서 discriminator 를 사용하여 record 내의 어떤 값을 저장할 지 구분하도록 하는데 이 때 발생하는 문제는 이 discriminator 가 한번 정해지면 변하지 않는 값이 아니라 계속해서 변할 수 있는 값이라는 것이 문제이다.

즉 tag(type)을 통해 저장되는 값의 형식을 정하고 값을 저장한 뒤 이 tag(type)을 바꿔서 값을 출력하게 되면 원하지 않는 값이 출력 되는 문제가 발생할 수 있다. 이는 Pascal 언어에서 프로그래머에게 사용할 때 주의하도록 하는 역할을 한다.

이를 해결하기 위해서는 사용할 때 주의해서 사용하는 것이 제일이지만 type 별로 분리하여 여러 record 를 만드는 것이 근본적인 해결 방안이라고 생각한다. 만약 그대로 사용하게 된다면

1)procedure 중간에 tag(type)을 검사하여 tag 에 따른 데이터의 일치 여부를 확인하는 과정을 추가하는 방법

2>tag 를 한번 정하면 바꿀 수 없도록 하는(ex)tag 가 아무것도 없을 때만 assign 이 가능, record 를 선언할 때만 tag(type)을 정함 등등) 과정을 추가하는 것 역시 가능할 것이라고 생각한다.