

프로그래밍언어론

2008년 시행 행정고등고시(기술직) 제2차시험

응시번호 :

성명 :

제 1 문. 프로그래밍 언어의 변수(variable) 사용에 관한 다음 물음에 답하시오. (총 10점)

- 대부분의 프로그래밍 언어는 변수를 다양한 종류로 분류하고 있다. 예를 들어, C 언어에서는 대표적으로 변수를 전역(global), 자동(automatic), 정적(static) 변수 등으로 나눈다. 이렇게 변수를 종류별로 나눌 때의 장점을 간단히 설명하시오. (4점)
- 다음 C 언어로 작성된 프로그램의 수행 중 printf 함수를 통해 출력되는 결과를 모두 기술하시오. (6점)

```
int a, b, c;

void foo(int x) {
    static int b = 1;
    int c = 1;
    a = a + b + c;
    b++;
    c = c + a * b;
    printf("%d %d %d\n", a, b, c);
    if (x > 0) foo(--x);
}

main() {
    a = b = c = 0;
    foo(5);
    printf("%d %d %d", a, b, c);
}
```

제 2 문. 다음은 큐(queue)를 추상 데이터 타입(abstract data type)의 대수적 명세(algebraic specification)로 기술한 것이다.

(단, 공리(axiom) 앞의 번호는 공리를 구분하기 위한 것으로 실제 명세에는 사용되지 않는다)

```
type queue(element) imports boolean
operations :
    create :    → queue
    enqueue :  queue × element → queue
    dequeue :  queue → queue
    front :    queue → element
    empty :    queue → boolean
variables :
    q : queue
    x : element
axioms :
    ① empty(create) = true
    ② empty(enqueue(q, x)) = false
    ③ front(create) = error
    ④ front(enqueue(q, x)) = if empty(q) then x else front(q)
    ⑤ dequeue(create) = error
    ⑥ dequeue(enqueue(q, x)) = if empty(q) then q else enqueue(dequeue(q), x)
```

대수적 명세의 연산(operation)에서 사용되는 용어는 다음과 같다.

- 생성자(constructor) : 명세에 의해 정의되는 타입으로 사상시키는 연산
- 소거자(destructor) : 생성자 중 이용 가능한 정보의 양을 감소시키는 연산
- 점검자(Inspector) : 다른 타입으로 사상시키는 연산
- 술어(predicate) : 점검자 중 논리값(boolean value)을 돌려주는 연산
- 선택자(selector) : 점검자 중 논리값이 아닌 값을 돌려주는 연산

다음 물음에 답하십시오. (총 15점)

- 1) 연산들 중에서 생성자, 소거자, 점검자, 술어, 선택자는 각각 어느 것인지 모두 기술하십시오. (4점)
- 2) 아래 식이 참임을 증명하십시오. (4점)

front(enqueue(enqueue(create, x), y)) = x

- 3) 하나의 원소를 가진 큐에 dequeue 연산을 수행하면 공백 큐가 됨을 위에서 기술한 공리를 이용하여 증명하십시오. (4점)
- 4) 어떤 대수적 명세를 위한 적절한 공리 집합을 찾는 것은 매우 어려운 일이지만, 일반적으로 소거자가 아닌 생성자와 점검자의 조합에 대해 각각 한 개씩, 소거자와 소거자가 아닌 생성자의 조합에 대해 각각 한 개씩의 공리가 필요하다. 이 방법에 근거하여 위에서 기술한 큐의 공리 6개가 유도된 이유를 설명하십시오. (3점)

제 3 문. 문맥자유문법(context-free grammar)에 관한 다음 물음에 답하십시오. (총 10점)

- 1) 언어 L이 아래의 문법 $G = (N, T, P, S)$ 를 통해 정의되어 있을 때, G가 표현하고자 하는 언어 L을 우리말로 설명하십시오. (3점)
(단, N과 T는 각각 비단말자(nonterminal)와 단말자(terminal)의 집합이고, P는 G의 생성규칙(production)을 정의하며, S는 시작 비단말자를 나타내고, ϵ 은 빈 문자열(empty string)을 나타낸다)

$N = \{ \langle p \rangle, \langle q \rangle \}$
 $T = \{ x, y, z \}$
 $P = \{ \langle p \rangle := x \langle p \rangle y \mid \langle q \rangle, \langle q \rangle := z \mid \epsilon \}$
 $S = \langle p \rangle$

- 2) 두 개의 문자열 xyzxy와 xxyy 중에서 L에 속한 문자열을 찾고 그 문자열에 대한 파스트리(parse tree)를 그리시오. (3점)
- 3) 열린 괄호와 닫힌 괄호만으로 구성된 문자열에서, 임의의 왼쪽 열린 괄호는 항상 오른쪽에 대응되는 닫힌 괄호를 가지고 있고, 그렇게 대응되는 열린 괄호와 닫힌 괄호의 쌍이 다른 쌍에 완전히 포함(properly nested)되어 있으면 이 문자열을 ‘괄호가 균형잡힌 문자열(balanced parentheses)’이라고 한다. 괄호가 균형잡힌 문자열의 예는 $()$, $()()$, $((()))()$ 등이다. 아래 생성규칙의 우변 빈 칸을 채워 괄호가 균형잡힌 가능한 문자열을 모두 생성하는 문법 $G' = (N', T', P', S')$ 를 완성하십시오. (4점)

$N' = \{ \langle s \rangle \}$
 $T' = \{ (,) \}$
 $P' = \{ \langle s \rangle := \text{ } \}$
 $S' = \langle s \rangle$

제 4 문. 아래 C 함수는 정수 배열, 배열의 길이(0이 아닌 정수), 정수 값을 인자로 받는다.
다음 물음에 답하시오. (총 15점)

```
int accum(int a[], int n, int b) {
    int i;
    int sum = b;

    for (i=0; a[i]>0 && i<n; i++) {
        sum = sum + a[i];
        a[i] = sum;
    }
    return sum;
}
```

- 1) 위 함수가 주어진 길이 n의 배열 a에 대하여 어떤 작업을 하는 함수인지 작업 내용과 결과 값을 설명하시오. (5점)
- 2) 위 함수의 수행 중 배열 범위 밖의 값에 대한 읽기 접근이 발생할 수 있다. 어떤 경우에 이런 일이 발생할 수 있는지 설명하고, 이 문제를 short-circuit 기법을 사용하여 해결하시오. (5점)

3) 다음은 위 함수와 똑같은 의미를 갖되 반복문을 사용하지 않는 재귀 함수를 C 언어로 작성한 것이다. 두 개의 빈 칸(㉠과 ㉡)을 채워 올바른 함수를 완성하시오. (5점)
(단, 전역변수는 사용하지 않는다)

```
int accum(int a[], int n, int b) {
    int sum = b;
    if ( ㉠ ) {
        return sum;
    }
    else {
        sum = sum + a[0];
        a[0] = sum;
        return ㉡ ;
    }
}
```

행정안전부 시험출제과장