

Programming Language

Assignment #4

- Copy 시 소스 제공자와 카피 당사자 공히 해당 과제 전체 0점 처리.
- 반드시 자기 자신의 생각과 글로 작성할 것. 다른 곳에서 자료 인용 시 출처 반드시 기재 (미기재 시 50% 감점)
- 한글/영문 상관 없음.

1. Answer the question.

Decide which of the following identifier names is valid in C language. Support your decision.

`_Student`
`int`
`Student`
`123Student`
`Student123`

C 언어의 변수 이름은, 영어 대문자와 소문자, 숫자와 밑줄 문자(_)만 사용할 수 있고, 변수의 첫 글자는 숫자가 아닌 글자여야 한다. 또한 예약어(int, double, struct, if, else, while, for, const, static 등의)를 변수명으로 사용할 수 없다. 이름 길이에 대한 제한은 없지만, 이름이 32자 이상인 경우 일부 컴파일러에서 문제가 발생할 수 있다.

이 조건을 바탕으로 정규 표현식을 작성해보면 다음과 같이 변수 이름을 표현할 수 있다. (예약어 조건 제외)

`[a-zA-Z][a-zA-Z0-9_]{0,31}`

`_Student`

첫 글자가 밑줄 문자이고, 이 외 문자가 대소문자로 구성되어 있어 올바른 변수 이름이다.

`Int`

예약어인 "int"를 변수 이름으로 사용할 수 없으므로 올바르지 않은 변수 이름이다.

`Student`

첫 글자가 대문자이고, 이 외 문자가 소문자로 구성되어 있어 올바른 변수 이름이다.

123Student

첫 글자가 숫자이므로, 올바르지 않은 변수 이름이다.

Student123

첫 글자가 대문자이고, 이 외 문자가 숫자와 소문자로 구성되어 있어 올바른 변수 이름이다.

2. Answer the question.

Consider the following Python program:

```
x = 1;
y = 3;
z = 5;

def sub1():
    a = 7;
    y = 9;
    z = 11;
    ...

def sub2():
    global x;
    a = 13;
    x = 15;
    w = 17;
    ...

def sub3():
    nonlocal a;
    a = 19;
    b = 21;
    z = 23;
    ...
...
```

List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

범위/변수	a	b	w	x	y	z
전역	정의 X	정의 X	정의 X	전역	전역	전역
	이곳에 선언된 변수는 전역 범위에 정의됨					

sub1	sub1	정의 X	정의 X	전역	sub1	sub1
	y, z는 sub1 범위의 변수를 참조함 (sub1 범위에 선언된 변수 참조를 우선시 하기 때문에)					
sub2	sub2	정의 X	sub2	전역	전역	전역
	global 키워드를 사용한 x는 전역 범위를 참조함					
sub3	sub2	sub3	sub2	전역	전역	sub3
	nonlocal 키워드를 사용한 a는 sub3(로컬)이 아닌 sub2의 것을 참조함					

3. Answer the question.

Assume the following JavaScript program was interpreted using static-scoping rules. What value of x is displayed in function sub1? Under dynamic-scoping rules, what value of x is displayed in function sub1?

```
var x;
function sub1() {
  document.write("x = " + x + "<br />");
}
function sub2() {
  var x;
  x = 10;
  sub1();
}
x = 5;
sub2();
```

```
var x;
// 전역 범위에 변수 x를 선언
function sub1() {
  document.write("x = " + x + "<br/>")
  // 정적 영역 규칙을 사용할 때, 이곳의 x는 sub1 범위에 존재하지 않으므로, 전역 범위의 x를
  참조하게 된다
  // 동적 영역 규칙을 사용할 때, 이곳의 x는 sub2 범위에 x가 있는지를 전역 범위에 x가 있는
  지 보다 우선 확인한다
}

function sub2() {
  var x;
  x = 10;
  sub1();
}
sub2();
```

출력 (정적 영역 규칙):

x = 5

출력 (동적 영역 규칙):

x = 10

4. Answer the question.

Consider the following C program:

```
void fun(void) {
    int a, b, c; /* definition 1 */
    ...
    while (...) {
        int b, c, d; /*definition 2 */
        ... ←———— 1
        while (...) {
```

```
            int c, d, e; /* definition 3 */
            ... ←———— 2
        }
        ... ←———— 3
    }
    ... ←———— 4
}
```

For each of the four marked points in this function, list each visible variable, along with the number of the definition statement that defines it.

표시점/변수	a	b	c	d	e
1	정의문 1	정의문 2	정의문 2	정의문 2	정의 X
	b, c가 정의문 2의 것으로 은닉됨				
2	정의문 1	정의문 2	정의문 3	정의문 3	정의문 3
	c, d가 정의문 3의 것으로 은닉됨				
3	정의문 1	정의문 2	정의문 2	정의문 2	정의 X
	표시점 1과 동일한 참조 구성				
4	정의문 1	정의문 1	정의문 1	정의 X	정의 X
	fun 함수의 범위에서 정의된 변수만을 참조할 수 있음				

5. Answer the question.

```
// main program
var x, y, z;

function sub1() {
    var a, y, z;
    ...
}

function sub2() {
    var a, b, z;
    ...
}

function sub3() {
    var a, x, w;
    ...
}
```

Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last function called? Include with each visible variable the name of the function in which it was defined.

- a. main calls fun1; fun1 calls fun2; fun2 calls fun3.
- b. main calls fun1; fun1 calls fun3.
- c. main calls fun2; fun2 calls fun3; fun3 calls fun1.
- d. main calls fun3; fun3 calls fun1.
- e. main calls fun1; fun1 calls fun3; fun3 calls fun2.
- f. main calls fun3; fun3 calls fun2; fun2 calls fun1.

프로그래밍 언어론 Robert W. Sebesta 유원희, 하상호 옮김 제 10판 책의

259쪽(5장 이름, 바인딩, 영역)에 있는 11번 문제, 12번 문제가 뒤섞인 것 같아 두 문제를 모두 풀었습니다.

11. 다음 C 골격 프로그램을 생각해보자.

```
void func1(void); /* 원형 */
void func2(void); /* 원형 */
void func3(void); /* 원형 */

void main() {
```

```

        int a, b, c;
        ...
    }

    void fun1(void) {
        int b, c, d;
        ...
    }

    void fun2(void) {
        int c, d, e;
        ...
    }

    void fun3(void) {
        int d, e, f;
        ...
    }

```

11. 다음 C 골격 프로그램을 생각해보자.

호출 순서가 다음과 같이 주어져 있고 동적-영역 규칙이 사용된다고 가정하면, 마지막으로 호출된 함수가 실행되는 동안에 어떤 변수가 가시적인가? 각 가시적인 변수에 대해 그 변수가 정의된 함수 이름을 포함시켜라.

a. main은 fun1을 호출; fun1은 fun2를 호출; fun2는 fun3를 호출.

main: a

fun1: b

fun2: c

fun3: d, e, f

b. main은 fun1을 호출; fun1은 fun3을 호출.

main: a

fun1: b, c

fun3: d, e, f

c. main은 fun2를 호출; fun2는 fun3를 호출; fun3는 fun1를 호출.

main: a

fun1: b, c, d

fun3: e, f

d. main은 fun3를 호출; fun3는 fun1을 호출.

main: a

fun1: b, c, d

fun3: e, f

e. main은 fun1을 호출; fun1은 fun3을 호출; fun3은 fun2 호출.

main: a

fun1: b

fun2: c, d, e

fun3: f

f. main은 fun3을 호출; fun3는 fun2을 호출; fun2은 fun1 호출.

main: a

fun1: b, c, d

fun2: e

fun3: f

12. JavaScript-유사 구문으로 작성된 다음 프로그램을 생각해보자.

```
// main 프로그램
var x, y, z;
function sub1() {
    var a, y, z;
    ...
}
```

```

function sub2() {
    var a, b, z;
    ...
}
function sub3() {
    var a, x, w;
    ...
}

```

호출 순서가 다음과 같이 주어져 있고 동적-영역 규칙이 사용된다고 가정하면, 마지막으로 호출된 함수가 실행되는 동안에 어떤 변수가 가시적인가? 각 가시적인 변수에 대해 그 변수가 정의된 함수 이름을 포함시켜라.

a. main은 sub1을 호출; sub1은 sub2를 호출; sub2는 sub3를 호출.

sub1: y

sub2: b, z

sub3: a, x, w

b. main은 sub1을 호출; sub1은 sub3을 호출.

sub1: y, z

sub3: a, x, w

c. main은 sub2를 호출; sub2는 sub3를 호출; sub3는 sub1를 호출.

sub1: a, y, z

sub2: b

sub3: w, x

d. main은 sub3를 호출; sub3는 sub1을 호출.

sub1: a, y, z

sub3: x, w

e. main은 sub1을 호출; sub1은 sub3을 호출; sub3은 sub2 호출.

sub1: y

sub2: a, b, z

sub3: w, x

f. main은 sub3을 호출; sub3는 sub2을 호출; sub2은 sub1 호출.

sub1: a, y, z

sub2: b

sub3: x, w

<The End of the Assignment>