Quiz #6 - Programming Language, 2017, Term 2

Name ( ) St. Id ( )

NB) 배점동일, 부분점수 인정

1. Fill in the blanks.

A. Clite 타입 체계의 완성

- 타입 규칙 10.1 모든 함수와 전역적인 식별자는 (유일)해야 한다.
- 타입 규칙 10.2 모든 함수의 (매개변수)와 지역변수는 서로 유일한 id를 가져야 한다.
- 타입 규칙 10.3 각 함수 내의 모든 문장은 함수의 지역변수, 매개변수, 접근 가능한 전역변수에 대하여 (합법적)이어야 한다.
- 타입 규칙 10.4 결과 타입이 void 가 아닌 모든 함수들은 main을 제외하고는 반환(return)문을 포함해야 하며, 반환되는 값을 나타내는 식은 해당 함수의 (타입)과 같아야 한다.
- 타입 규칙 10.5 void 함수에서는 (return)문이 나올 수 없다.
- 타입 규칙 10.6 모든 함수 호출 문장은 void 함수를 명시해야 하며, 모든 함수 호출식은 void가 아닌 함수를 명시해야 한다.
- 타입 규칙 10.7 모든 함수 호출은 명시된 함수의 매개변수의 개수와 (같은) 수의 인수값들을 사용해야 하며 각각의 인수값은 (왼쪽)부터 (오른쪽)으로 각각 대응하는 매개변수와 같은 타입이어야 한다.
- 타입 규칙 10.8 함수 호출의 타입은 호출된 함수의 타입이 되며 함수 호출을 나타낸 식은 타입 규칙 6.5와 6.6에 대하여 타당하여야 한다.

B. 다음 피보나치수의 계산 예제 코드에서의 타입 규칙들에 대해 적용 비적용 여부와 타당함을 논하시오.

```
int fibonacci (int n) {
    int fib0, fib1, temp, k;
    fib0 = 0; fib1 = 1; k = n;
    while (k > 0) {
        temp = fib0;
        fib0 = fib1;
        fib1 = fib0 + temp;
         k = k - 1;
    }
    return fib0;
}

int main () {
    int answer;
    answer = fibonacci(8);
}
```

타입 규칙 10.4 int fibonacci (int n) 함수 안에 return fib0;문장이 있음.

타입 규칙 10.5 은 적용 안됨.

타입 규칙 10.6 함수 호출 fibonacci(8)은 void가 아닌 함수 int

fibonacci (int n)를 명시함

타입 규칙 10.7 함수 호출 fibonacci(8) 은 하나의 인수값을 가지며 이 타입은 매개변수 n과 일치함

타입 규칙 10.8 함수 호출 fibonacci(8)이 포함된 계산식은 타당함.

C. Semantics of Call and Return. 의미 규칙 10.1 함수 f에 대한 호출 c의 의미는 다음과 같은 단계로 수행된다:

1. 새로운 (활성) 레코드를 만들고 f의 매개변수들과 지역변수들을 위치시킨다.
2. c의 (인수)값들을 계산하고 그 값을 f의 활성 레코드의 대응하는 매개변수에 저장한다.
3. 함수가 void 타입이 아닐 경우 활성 레코드에 함수의 이름과 (타입)이 일치하는 결과 변수를 추가한다.
4. 만들어진 활성 레코드를 실행 시간 (스택)에 추가(push)한다.
5. f 함수 몸체의 명령문들을 수행한다.
6. 실행 시간 스택으로부터 활성 레코드를 제거(pop)한다.
7. 만약 함수가 void 타입이 아닐 경우 (결과) 변수의 값을 함수 호출에 반환한다.

2. Fill in the blanks.

A. 메모리의 분할:

- (정적 영역 (static area)): 실행 전에 사용량과 내용이 결정된 공간
- (실행시간 스택 (runtime stack)): 크기나 내용이 변경되는 공간. 함수 호출의 중추 역할.
- (힙 (heap)): 크기나 내용이 변경되는 공간. 동적 자료구조를 위해 사용.

B. 메모리의 분할:

- (쓰레기(garbage))는 프로그램에서 접근 불가능한 힙 블록을 말한다. 고아(orphan)라고도 한다.
- (끊어진 참조(dangling pointer))는 더 이상 할당되어 있지 않는 힙 블록을 참조하는 것을 말한다. 과부(widow)라고도 한다.

3. Answer the questions.

A. C 언어는 프로그래머가 힙 공간을 할당받고 반납할 수 있는 malloc 함수와 free 함수를 제공한다. 이 두 함수에 대해 할당(new) 반납(delete) 연산과 비교하여 차이점을 논하라.

C functions `malloc` and `free` have the following prototypes:

```
void *malloc(size_t size);
void free(void *ptr);
```

The first allocates at least `size` bytes of memory and returns a pointer to the first byte of that block. If such a block is not available, the `null` pointer is returned.

The second deallocates that block of memory referenced by `ptr` that had been allocated by `malloc`.

These functions are thus similar to the functions *new* and *delete*, except that they exclude the idea of marking a memory block *unused* or *undef*. In C, unused blocks of memory are not so marked and the "undefined" value has no predefined meaning.

B. 시간과 공간 면에서 동적 배열을 실행시간 스택에 할당할 때의 장단점을 기술하라.

While the stack and heap space requirements would change, as noted above, no appreciable impact on the program's run-time performance would occur. That is, array arguments can still be passed by reference, whether they are originally allocated in the stack or in the heap. However, the dynamic allocation and addressing of arrays in the stack is a far more complex problem than this question suggests.

<End of the Quiz>