**Test #6. 2012.4.19 프로그래밍언어론 01 분반- 개인 문제**

학번:

이름:

조번호:

역할 : 사회자, 일반 조원 ........................ (자신의 팀 내의 역할에 동그라미 하시오.)

1. 다음 각 항목이 참이면 O, 아니면 X 를 하시오.
2. 다음과 같은 C 변수 a와 b 가 있다고 가정하자.

struct A {

int f;

} a;

int b[1];

만일 int i=0;와 같은 선언이 추가되었을 때,

a.f 보다 b[i]를 접근하는 속도가 더 빠르게 된다.

1. Union 타입은 메모리 효율성을 더 좋게 하기 위해 사용된다.
2. 다음과 같은 코드에서 마지막 줄의 strcpy 함수 호출 시 C는 컴파일 오류를 낸다.

union number {

int value;

char data[4];

} x;

x.value = 100; ...

strcpy(x.data, “abc”);

1. 포인터간의 뺄셈은 가능하다.
2. C의 포인터는 stack의 변수를 가리킬 수 있다.
3. 일반 포인터를 void \* 타입의 포인터로 지정할 때는 (void\*)와 같은 casting이 필요하다.
4. 다음은 dangling pointer (허상포인터) 문제를 안고 있다.

char \* p1 = (char \*) malloc(sizeof(char)) ;

char \* p2 = p1;

free p1;

printf(“%c”, \*p2);

1. 다음 분실된(lost) heap-dynamic 변수 문제를 안고 있다.

char c;

char \* p1= &c;

p1 = (char\*) malloc(sizeof(char));

1. ~~실제 포인터가 직접 heap-dynamic 변수를 가리키지 않고 대신 비석(tombstone)을 가리키도록 구현하면, 나중에 heap-dynamic 변수가 반환될 때 포인터는 nil을 가리키게 된다.~~

(tombstone을 가리키므로 출제의도는 X. 그러나 tombstone이 nil값을 가지므로 대부분의 학생들이 모호함을 주장하여 무효처리)

1. Garbage collection 에서는 자신을 참조하는 포인터의 개수가 0일 때 제거된다.

2. Reference Count (참조계수)가 해결하려는 문제점이 무엇인지 쓰고, 이 방법의 단점을 적으시오.

3. 포인터에 대해 free나 delete와 같은 명시적 회수 방법을 제공하지 않는 언어에서는 dangling reference 문제가 발생할 여지가 없는가? 의견을 간단히 적으시오.

**Test #6. 2012.4.19 프로그래밍언어론 01 분반 조별 문제**

조번호:

참여자 (학번, 이름)

-

-

-

-

-

-

1. Garbage collector 가 작동하는 경우라도 stack 구조에서는 memory leak (메모리 누수)가 발생할 수 있다. 이유를 구체적으로 적어보시오.