

Test #6. 2015.4.13 프로그래밍언어론 03 분반

학번:

이름:

* 문제에 답을 직접 적지 마시기 바랍니다.*

글씨크기를 작게 하되 또박또박 적어주시기 바랍니다.

1. 다음 각 항목이 참이면 O, 아니면 X 를 선택하여 1줄로 적으시오. (예를 들어 (1)O (2)O (3)O (4)O (5)X 라면, OOOOX 로 적는다)

- (1) 같은 코드에서 마지막 줄의 strlen 함수 호출 시 C는 컴파일 오류를 낸다.

```
union number { int value; char data[4]; } x;  
x.value = 100; ...  
strlen(x.data);
```

- (2) 다음과 같은 C 변수 a와 b 가 있다고 가정하자.

```
struct A { int f; int g; int h; } a;  
int b[3];
```

만일 변수 i의 값이 2일때, a.h를 접근하는 속도는 b[i]를 접근하는 속도보다 느리다.

- (3) malloc(sizeof(char)) 의 결과값은 (void*) 이다.

- (4) garbage collection에서는 메모리 두 개가 서로 참조하고 있는 순환 참조의 경우에 프로그램에서 사용이 끝난 경우라도 메모리가 제거되지 않는다.

- (5) C#에서는 포인터와 reference 가 둘 다 제공된다.

1. 2. 3.

2. 평면 좌표상의 한 점을 나타내는 다음과 같은 자료구조 struct Point 가 있다고 할 때

```
struct Point{  
    int x; int y;  
};
```

아래 변수 ps은 스택에 할당되어 사용된다.

```
struct Point ps[2];  
ps[0].x = 0;      ps[0].y = 0;  
ps[1].x = 50;     ps[1].y = 50;
```

이것을 동일한 내용으로 변수를 heap에 할당한 코드가 아래와 같다면, 밑줄을 채우시오.

```
struct Point * ph = (struct Point *) malloc(sizeof(struct Point));  
ph->x = 0;  ph->y =0;  
  
_____  
ph->x = 50;  ph->y = 50;
```

3. 다음 코드에 대한 설명으로 가장 올바른 것을 하나만 고르시오. (틀리면 감점 있음)

```
char c;  
char * p1= &c;  
p1 = (char*) malloc(sizeof(char));
```

- (1) 비석 방법으로 해결될 수 있다.

- (2) garbage collection으로 해결될 수 있다.

- (3) dangling pointer 문제나 lost heap dynamic variables 문제가 발생되지 않는다.