1. Given the array declaration

C
short a[11];

Oberon-like
VAR a : ARRAY 11 OF SHORTINT;

Mark with an A the memory location where we would find

a[7]

2. Show the memory layout of the following C struct/record definition taking into consideration the SPARC data type memory alignment restrictions discussed in class. Fill bytes in memory with the appropriate struct/record member/field name. For example, if member/field name p takes 4 bytes, you will have 4 p's in the appropriate memory locations. If the member/field is an array, use the name followed by the index number. For example, some number of p0's, p1's, p2's, etc. Place an X in any bytes of padding. Structs are padded so the total size is evenly divisible by the most strict alignment requirement of its members.

struct foo {
    char    a;
    short   b[2];
    char    c[3];
    int     d;
    double  e;
    float   f;
}

struct foo fubar;

What is the offsetof( struct foo, d )?

What is the sizeof( struct foo )?
3. Some languages (like C++) allow the programmer can define local variables anywhere in a block and that variable name is scoped to that block from that point on to the end of the block. Needless to say, this complicates the compiler's scoping and type checking mechanisms.

Consider the following valid C++ program fragment:

```cpp
int main( char *argv[], int argc )
{
    int i = 2;

    while ( i == 2 )
    {
        i--;
        int i = 2;
        i++;
        if ( i > 2 )
        {
            i--;
            int i = 2;
            i++;
            cout << i << " "; // Output the current value of i followed by a space
        }
        cout << i << " "; // Output the current value of i followed by a space
    }
    cout << i << endl; // Output the current value of i followed by a newline
    return 0;
}
```

What gets printed? _______________________

4. Indicate whether the following expressions are
   A. legal (no compiler error) or
   B. illegal (compiler error).

```cpp
int a[10];
int *iPtr;
&a[4] = iPtr;          ______  iPtr = &a[4];          ______
*(iPtr - 3) = a[3];    ______  *&a[3] = *iPtr;         ______
&a = iPtr + 420;       ______  *(a+2) = 2[iPtr];         ______
```

What question would you most like to see on the Midterm?