Princeton University
COS 217: Introduction to Programming Systems
Fall 2001
Midterm I Answers

Question 1

(a) average is a pointer to a pointer to a double.

(b) sqrt is an array of 5 pointers to functions. Each of those functions takes a pointer to an integer and returns a pointer to an integer.

(c) a is an array of 10 pointers to integers.

(d) k is a function that takes an int and returns a pointer to a character.

(e) j is a pointer to an array of 10 integers.

Question 2

(a) int (*lolla)(char **);

(b) const double **b;

(c) unsigned int *(*list_scores)(struct assignment*);

(d) char f(int(*)());

Question 3

(a) True. The variable q is a constant pointer. It is legal to change the contents of the storage to which q is pointing.

(b) False. Since q is a constant pointer, it is illegal to change the value of q such that it points to different storage.

Question 4

```
0
8
```

Explanation:

```
(**funcs)(array)
= (**(funcs+0)(array)
= (*funcs[0])(array)
= (*(&f0)(array)
= f0(array)
= **array
= **(array+0)
= *(array[0])
```

```
= *(array[0]+0)
= array[0][0]
= "0"

(**(funcs+1))(array+1)
= (*funcs[1])(array+1)
= (*&f1)(array+1)
= f1(array+1)
= *(*(array+1+1)+2)
= *(*(array+2)+2)
= *(array[2]+2)
= array[2][2]
= "8"
```

Question 5

```
3
4
6
8
12
16
```

Explanation:

f is a function. The variable b within f refers to the global b,
whose value is 2.

m is a macro, and so the calls to m within main are expanded inline
within main. The variable b in the calls to m thus refers to the b
that is declared within main, whose value is 4.

Question 6

```
5 6 7
0 0 7 0
10 10 17 10
5 10 7
5 10 3
0 0 3 10
10 10 13 20
5 10 3
```

Explanation:

The variable a in file sub.c is a global static variable. Thus
the variable a in file main.c refers to different storage from the
variable a in file sub.c. Changing the value of a in main.c does
not affect the value of a in sub.c, and vice-versa.

The variable b is a global non-static variable. Thus the variable
b in file main.c and the variable b in file sub.c refer to the same
storage. Changing the value of b in main.c affects the value of b
in sub.c, and vice-versa.

The variable c as used in file main.c is a local variable that is

declared at multiple levels.

The variable c as used in file sub.c is a formal parameter of the function f. Since the C language uses call by value, changing the value of c in f does not affect the corresponding actual parameter in main.

The variable d is a static variable that is local to function f. It is initialized once at program startup. Thereafter it retains its value between calls to f.

Question 7

32
400

Explanation:

sizeof(struct student):

```
   struct student {
      char **name;                        (4 bytes)
                                          (4 bytes padding)

      union u {
         double f;
         int i;
      } lucky_number;                     (8 bytes)
      enum e {male, female} gender;       (4 bytes)
      unsigned short idnum;               (2 bytes)
                                          (2 bytes padding)
      struct student *friend;             (4 bytes)
                                          (4 bytes padding)
   };


                                          (TOTAL:  32 bytes)
```

   (Note: Some compilers omit the first 4-byte pad and/or the second 4-byte pad, so it would be acceptable for you to omit either from your answer. It would not be acceptable to omit the 2-byte pad from your answer.)

sizeof(cs217)

   cs217 is an array containing 100 elements. Each element is of type struct student*, and thus consists of 4 bytes. Thus cs217 consists of 400 bytes.

Question 8

```
01651 (base 8) = 00000011 10101001 (base 2)
00735 (base 8) = 00000001 11011101 (base 2)

  + 01651 (base 8) = 00000011 10101001 (base 2)
+ - 00735 (base 8) = 11111110 00100011 (base 2)
_____

      714 (base 8) = 00000001 11001100 (base 2)
```

Question 9

(a) e51d

Explanation:

```
x = a << 12
x = 14 << 12
x = 0x0000000e << 12
x = 0x0000e000

x |= (b & 0x007) << 8
x |= (5 & 0x007) << 8
x |= (0x0005 & 0x0007) << 8
x |= (0x0005) << 8
x |= 0x0500
x = x | 0x0500
x = 0xe000 | 0x0500
x = 0xe500

x |= c & 0xff
x |= 29 & 0xff
x |= 0x001d & 0x00ff
x |= 0x001d
x = x | 0x001d
x = 0xe500 | 0x001d
x = 0xe51d
```

(b)

```
struct bits
{
   unsigned field1:4;
   unsigned UNUSED:1;
   unsigned field2:3;
   unsigned field3:8;
};
```

Question 10

The exam question states that there are three errors in the program. In fact, the professor made a mistake; there are only two errors in the program:

(1) When the insert function discovers a duplicate part number, it fails to free the memory that was allocated to store the new Ship. Thus there is a memory leak in the insert function.

(2) The part field of each node of the list points to the same storage. That storage is overwritten with each call to the input function. Thus, effectively, each Ship in the list has the same part information.