

Princeton University  
COS 217: Introduction to Programming Systems  
Midterm 1 Answers  
Fall 1999

**Question 1**

(a)

125 (decimal) = 01111101 (binary, signed magnitude)  
-75 (decimal) = 11001011 (binary, signed magnitude)  
11010101 (binary, signed magnitude) = -85 (decimal)  
00101010 (binary, signed magnitude) = 42 (decimal)

(b)

+10 (decimal) = 00001010 (binary)  
+ +45 (decimal) = 00101101 (binary)  
-----  
+55 (decimal) = 00110111 (binary)

No overflow.

-40 (decimal) = 11011000 (binary)  
+ -20 (decimal) = 11101100 (binary)  
-----  
-60 (decimal) = 11000100 (binary)

No overflow.

-40 (decimal) = 11011000 (binary)  
+ -90 (decimal) = 10100110 (binary)  
-----  
+126(decimal) = 01111110 (binary)

Overflow.

**Question 2**

```
c = a & 0xFFFFF000  
c |= b >> 20;
```

**Question 3**

(a) 0  
6  
7

(b) 17

#### Question 4

- (a) 

```
struct day
{
    int (*date)[3];
    char **events;
};
```
- (b) 

```
struct day_element
{
    struct day today;
    struct day_element *tomorrow;
};
struct day_element *month;
```
- (c) 

```
struct day_element *year[12];
```
- (d) year is an array of 12 elements. Each element is of type struct day\_element\*\*. (year + 1) is the address of the second struct day\_element\*\* in the year array.
- (e) 

```
void (*apply)(void *x[30], void (*y)(void *xx, void **yy));
```
- (f) 

```
int (*funcs[4])(void*) = {f1, f2, f3, f4};
```
- (g) The question is unclear. Does "the third function" mean element 2 of the array, or element 3 of the array? I'll assume the former.

Reading the question literally, the answer should be:

```
(*funcs[2])(a);
```

But that's a type mismatch: the actual parameter "a" is of type int, but the formal parameter of (\*funcs[2]) is of type void\*. Maybe the expected answer is:

```
(*funcs[2])((void*)a);
```

Note that "a" is explicitly cast to be of type void\* -- a very unusual and dangerous thing to do. Or maybe the expected answer is:

```
(*funcs[2])(&a);
```

#### Question 5

The insert1 function assigns the address of the variable named one into the newly created linked list node. But the array named one ceases to exist after insert1 is finished executing. Thus the expression q->val in the main function is a dangling pointer.

### Question 6

(a)

Since `ptree` is a pointer, the expression `sizeof(ptree)` evaluates to 4. That is incorrect. Instead, the call to `malloc` should be either

```
malloc(sizeof(struct tree))
```

or

```
malloc(sizeof(*pnew))
```

(b)

2  
3  
4

### Question 7

(a) Sorry -- Can't draw a diagram using text only. Here's an outline:

```
main
  main.o
  main.c
  stat.o
  stat.c
  freq.h
  freq.o
  freq.c
  list.h
  store.o
  store.c
  table.h
  list.h
  list.o
  list.c
  list.h
  table.o
  table.c
  table.h
```

(b)

Assuming that all files are currently up to date...

```
touch main.c
make -n
  lcc -A -c main.c
  lcc -A -o main main.o stat.o freq.o store.o list.o table.o
make
  lcc -A -c main.c
  lcc -A -o main main.o stat.o freq.o store.o list.o table.o
```

```
touch freq.h
rm -f main.o
make -n
  lcc -A -c main.c
  lcc -A -c stat.c
  lcc -A -o main main.o stat.o freq.o store.o list.o table.o
rm -f list.o
make -n
  lcc -A -c main.c
  lcc -A -c stat.c
  lcc -A -c list.c
  lcc -A -o main main.o stat.o freq.o store.o list.o table.o
rm -f *.o
make -n
  lcc -A -c main.c
  lcc -A -c stat.c
  lcc -A -c freq.c
  lcc -A -c store.c
  lcc -A -c list.c
  lcc -A -c table.c
  lcc -A -o main main.o stat.o freq.o store.o list.o table.o
```

(c)

These commands (and only these commands) would need to be executed to bring the project up to date:

```
lcc -A -c list.c
lcc -A -o main main.o stat.o freq.o store.o list.o table.o
```