

1. Rewrite the following C declarations in English. For example, the declaration “`int *x`” would be “`x` is a pointer to an integer”.

(a) `double **average;`

(b) `int *(*sqrt[5])(int *);`

(c) `int *a[10];`

(d) `char *k(int);`

(e) `int (*j)[10];`

2. Give the C declaration for each of the following:

(a) `lolla` is a pointer to a function that takes a pointer to a pointer to a character, and returns an integer

(b) `b` is a pointer to a pointer to a constant double

(c) `list_scores` is a pointer to a function that takes a pointer to a structure named `assignment` as an argument, and returns a pointer to an unsigned integer

(d) `f` is a function that takes a pointer to a function (that takes no arguments and returns an `int`) as its argument, and returns a character.

3. Suppose we make the declarations:

```
char z[10];
char *const q = z;
char *p = NULL;
```

(a) True or False: The following statement is legal.

```
*q = '2';
```

(b) True or False: The following statement is legal.

```
q=p;
```

4. What does the following program output when executed?

```
char *f0 (char *a[][3])
{
    return (**a);
}

char *f1 (char *a[][3])
{
    return (*(*(a+1)+2));
}

int main (void)
{
    char *(*funcs[2]) (char *array[][3]);
    static char *array[][3] =
        {{"0", "1", "2"},
         {"3", "4", "5"},
         {"6", "7", "8"}};

    funcs[0] = &f0;
    funcs[1] = &f1;
    printf("%s\n", (**funcs)(array));
    printf("%s\n", (**(funcs+1))(array+1));
    return (0);
}
```

5. What does the following program output when executed?

```
#include <stdio.h>

int a = 1;
int b = 2;

int f(int i)
{
    return b * i;
}

#define m(i) b * i

int main(void)
{
    int a = 3;
    int b = 4;

    printf("%d\n", a);
    printf("%d\n", b);
    printf("%d\n", f(a));
    printf("%d\n", f(b));
    printf("%d\n", m(a));
    printf("%d\n", m(b));

    return 0;
}
```

6. Consider the program given in the next column. Note that it is implemented in three files. What will the program output when executed?

sub.h

```
void f(int c);
```

sub.c

```
#include "sub.h"
#include <stdio.h>

static int a;
extern int b;

void f(int c)
{
    static int d = 0;
    a = 0;
    b = 0;
    printf("%d %d %d %d\n", a, b, c, d);
    a += 10;
    b += 10;
    c += 10;
    d += 10;
    printf("%d %d %d %d\n", a, b, c, d);
}
```

main.c

```
#include "sub.h"
#include <stdio.h>

int a, b;

int main(void)
{
    int c;
    a = 1;
    b = 2;
    c = 3;
    {
        int c;
        a = 5;
        b = 6;
        c = 7;
        printf("%d %d %d\n", a, b, c);
        f(c);
        printf("%d %d %d\n", a, b, c);
    }
    printf("%d %d %d\n", a, b, c);
    f(c);
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

7. What does the following program output when executed?

```
struct student {
    char **name;
    union u {
        double f;
        int i;
    } lucky_number;
    enum e {male, female} gender;
    unsigned short idnum;
    struct student *friend;
};
```

```
main()
{
    struct student *cs217[100];

    printf("%d\n", sizeof(struct student));
    printf("%d\n", sizeof(cs217));
}
```

8. Evaluate $01651_8 - 00735_8$ using two's complement arithmetic. Show your work, assuming a 16-bit word size. Translate the result back into octal.

9. Consider the following program:

```
main() {
    int a = 14;
    int b = 5;
    int c = 29;
    unsigned short x = 0;

    x = a << 12;
    x |= (b & 0x700) << 8;
    x |= c & 0xff;
    printf("%x\n", x);
}
```

(a) What does the program output when executed?

(b) Give the declaration for a structure `bits`, such that the following snippet of code results in the same value being stored in `x`. Your solution should be general, that is, not specific to the values of `a`, `b`, and `c` given above.

```
struct bits x;

x.field1 = a;
x.field2 = b;
x.field3 = c;
```

10. The following program reads a list of information about parts that are being shipped (one part per line), removes any duplicate entries (entries with the same part number), and prints out the resulting list of parts. Identify the three serious **runtime** errors in the program. For each error, describe the problem in enough detail to convince us that you understand what's wrong and how the program will behave. Assume functions input and output exist, are bug-free, and behave exactly as you would expect: the former reads a line from standard input and extracts the month, day, and part information, and the latter prints the corresponding information to standard out.

Note: Points will be deducted for any non-problems you identify.

Hint: the errors have nothing to do with the logic involved in traversing the linked list.

```
#define MAXLINE 128
#include <stdio.h>

struct part_info {
    int number;
    char origin[50];
};

struct ship_info {
    char *month;
    int day;
    struct part_info *part;
    struct ship_info *next;
};

typedef struct ship_info *Ship;

int main() {
    Ship p, list=NULL;
    struct part_info part;
    int month, day;
    char line[MAXLINE];

    while (fgets(line, MAXLINE, stdin)) {
        input(line, &month, &day, &part);
        insert(&list, month, day, &part);
    }
    for (p=list; p; p=p->next)
        output(p);
    return 0;
}
```

```
int insert(Ship *list,
           int month,
           int day,
           struct part_info *part)
{
    Ship new, p;
    char *months[] = {"Jan", "Feb", "Mar",
                     "Apr", "May", "Jun",
                     "Jul", "Aug", "Sep",
                     "Oct", "Nov", "Dec" };

    new = (Ship) malloc(sizeof *new);
    new->month = months[month-1];
    new->day = day;
    new->part = part;
    p = *list;
    if (!p) {
        /* list was empty */
        new->next = NULL;
        *list = new;
        return 1;
    }
    while (p->next) {
        if (part->number == p->part->number) {
            /* duplicate part: discard */
            return 0;
        }
        p = p->next;
    }
    if (part->number == p->part->number)
        return 0;
    new->next = NULL;
    p->next = new;
    return 1;
}
```