



The following code fragment applies to questions 1 and 2:

```
int a, b, c;  
  
a = 4;  
b = 5;  
b += a--;
```

1. What is the value of a after the execution of the code fragment above?
  - (a) 3
  - (b) 4
  - (c) 8
  - (d) 9
  - (e) None of the above

2. What is the value of b after the execution of the code fragment above?
  - (a) 3
  - (b) 4
  - (c) 5
  - (d) 8
  - (e) 9

3. Consider the following C code fragment:

```
int i[] = { 2, 3, 0, 3, 2, 0, 0, 9 };  
int *p;
```

```
p = &i[4];
```

Which of the following expressions is false?

- (a) sizeof(i)
  - (b) \*(p+3) == 9
  - (c) \*p+3 == 5
  - (d) p[1]
  - (e) None of the above
4. Given the declaration: `char** f[5];`  
Which of the following is of type `char*`?
    - (a) `f[1];`
    - (b) `&(f[4]);`
    - (c) `***f;`
    - (d) `*(f[3]);`
    - (e) `(f+4);`
  5. Which of the following is the Subversion command to store changes to files in the repository?
    - (a) add
    - (b) checkin
    - (c) commit
    - (d) update
    - (e) log

The following applies to questions 6, 7, and 8:

A computer system requires natural alignment and items of type character, int, long, float, double and pointers occupy 1, 4, 4, 4, 8 and 4 bytes respectively. Consider the following declarations:

```
struct Point2 { double x,y; };
struct Point3 { double x,y,z; };
struct Line {
    struct Point2 p[2];
    char above;
};
struct Plane {
    struct Point3 p[3];
    double angle;
    char above;
};
union HalfSpace {
    struct Line two;
    struct Plane three;
};

union HalfSpace hs;
union HalfSpace* hsp = &hs;
```

6. What is the result of `sizeof(struct Point2)`?
  - (a) 8
  - (b) 16
  - (c) 24
  - (d) 32
  - (e) None of the above
  
7. What is the result of `sizeof(struct Plane)`?
  - (a) 40
  - (b) 57
  - (c) 64
  - (d) 88
  - (e) None of the above
  
8. Which of the following expressions is valid and always true?
  - (a) `hs.two.above == hs.three.above`
  - (b) `hsp->two.p[1].x == hs.three.p[1].x`
  - (c) `hsp->three.angle == hsp->two.angle`
  - (d) `hs.three.p[1] == hsp.three.p[1]`
  - (e) `hs.two.p[0].x == hsp->three.p[0]->x`

9. Consider the following C program.

```
#include <stdio.h>

int main(int argc, char* argv[]) {
    int i, n = 0, z = 0;
    for(i = 1; i < argc - 1; i++) {
        if (n == 0) {
            if ((strcmp(argv[i], "-f") == 0) ||
                (strcmp(argv[i], "-g") == 0)) {
                n = 1;
                printf("%s ", argv[i]);
                z++;
            } else if (strcmp(argv[i], "-h") == 0) {
                z++;
            }
        } else {
            printf("%s ", argv[i]);
            n = 0;
        }
    }
    printf("%d\n", z);
    return z;
}
```

If the program is compiled to an executable called `a.out` in the current directory, what will be printed as a result of running: `./a.out -g X -f y -h`

- (a) `-g -f -h 3`
  - (b) `-g X -f y 2`
  - (c) `-g X -f y 3`
  - (d) `-g X -f y -h 3`
  - (e) None of the above
10. Consider the program shown in Question 9. What is the exit status of the program as a result of running: `./a.out -h -g -f -f -f`
- (a) 0
  - (b) 2
  - (c) 3
  - (d) 4
  - (e) None of the above
11. Which of the following best describes the type of `var` in the following declaration?
- ```
char *var(char*);
```
- (a) A function taking a character pointer argument and returning a character
  - (b) A pointer to a function taking a character pointer argument and returning a character
  - (c) A function taking a character pointer argument and returning a character pointer
  - (d) A pointer to a function taking a character pointer argument and returning a character pointer
  - (e) A pointer to a function taking a character argument and returning a character pointer
12. Which of the following declarations of `var` is a function which returns an integer and has a one argument which is an array of pointers to functions which take a `char` and return a `short`?
- (a) `int var(short (*) (char) []);`
  - (b) `int (*var) (short (*) (char) []);`
  - (c) `int var(short (*) [] (char));`
  - (d) `int (*var) (short (*) [] (char));`
  - (e) `int var(short (*) [] (char));`

13. Consider the following commands and output from a shell session (the commands typed by the user are shown in **bold**):

```
> pwd
/tmp/exam/esys13
> ls -al
total 980
drwxr-xr-x 11 jfenwick audio      4096 2009-04-16 13:27 .
drwxr-x--x  3 jfenwick audio      4096 2009-04-16 13:22 ..
-rwxr-xr-x  1 jfenwick audio     3528 2009-04-16 13:22 autodocs.sh
-rwxr-xr-x  1 jfenwick audio     2676 2009-04-16 13:21 autotest-scons
drwxr-xr-x  2 jfenwick audio      4096 2009-04-16 13:21 bin
-rw-r--r--  1 jfenwick audio     2848 2009-04-16 13:21 config.log
drwxr-xr-x 10 jfenwick audio      4096 2009-04-16 13:21 doc
drwxr-xr-x  8 jfenwick audio      4096 2009-04-16 13:22 escript
drwxr-xr-x  8 jfenwick audio      4096 2009-04-16 13:21 include
drwxr-xr-x  3 jfenwick audio      4096 2009-04-16 13:21 lib
lrwxrwxrwx  1 jfenwick jfenwick    10 2009-04-16 13:27 log -> config.log
dr-xr-xr-x  6 jfenwick audio      4096 2009-04-16 13:21 modellib
drwxr-xr-x  6 jfenwick audio      4096 2009-04-16 13:21 packaging
-rw-r--r--  1 jfenwick audio       560 2009-04-16 13:21 README_LICENSE
-rw-r--r--  1 jfenwick audio     1656 2009-04-16 13:21 README_TESTS
drwxr-xr-x  3 jfenwick audio      4096 2009-04-16 13:21 release
drwxr-xr-x  3 jfenwick audio      4096 2009-04-16 13:21 scons
-rw-r--r--  1 jfenwick jfenwick 883933 2009-04-16 13:21 .sconsign.dblite
-rw-r--r--  1 jfenwick audio     34953 2009-04-16 13:22 SConstruct
-rwxr-xr-x  1 jfenwick audio     11240 2009-04-16 13:21 utest.sh
```

The only member of the 'audio' group is user 'jfenwick'.

Which of the following statements (based on current permissions) is true?

- (a) Any user may create subdirectories in /tmp/exam/esys13.
  - (b) Any user can list the names of files in /tmp/exam
  - (c) Any user can write to the file log
  - (d) jfenwick can create files in the modellib directory.
  - (e) None of the above.
14. Consider the scenario of question 13 where user 'jfenwick' (in the directory /tmp/exam/esys13) executes a shell script that has the following content:

```
#!/bin/sh
count=0
for i in r* ; do
    count=`expr $count + 1`
done
echo $count
```

What will be printed as a result of running this script?

- (a) 0
  - (b) 1
  - (c) 3
  - (d) 5
  - (e) None of the above
15. If the current directory is /tmp/exam/esys13, which of the following path names is NOT equivalent to /tmp/exam/esys13/autotest-scons?
- (a) autotest-scons
  - (b) ../esys13/autotest-scons
  - (c) autotest-scons/./autotest-scons
  - (d) ./autotest-scons
  - (e) All of the above are equivalent to /tmp/exam/esys13/autotest-scons

16. Consider the following function:

```
char* concatenate(char* str1, char* str2) {
    char* str;
    str = (char*)malloc( ... );
    sprintf(str, "%s%s\n", str1, str2);
    return str;
}
```

What expression should be placed at the point indicated by ... so the memory allocated is exactly that required?

- (a) `strlen(str1) + strlen(str2) - 1`
- (b) `strlen(str1) + strlen(str2)`
- (c) `strlen(str1) + strlen(str2) + 1`
- (d) `strlen(str1) + strlen(str2) + 2`
- (e) `strlen("%s%s\n") + 1`

For questions 17 and 18, consider a UNIX file system with a disk block size of 16kbytes which uses i-nodes with a variant on a double-indirect index. Each disk block pointer is 4 bytes long and the i-node contains 15 block pointers in total (12 direct block pointers, **two** single indirect, and **one** double indirect).

17. What's the largest file that can be stored without using the double-indirect block pointers?

- (a) 192 kbytes
- (b) 65,728 kbytes
- (c) 131,072 kbytes
- (d) 131,264 kbytes
- (e) 268,566,720 kbytes

18. For a file of maximum size, how many disk blocks are used to store index information?

(Don't include the i-node.)

- (a) 4096
- (b) 4097
- (c) 4098
- (d) 8194
- (e) 16785408

19. Consider a computer system which uses byte addressing with page frames of size 8 kbytes.

If a process' page table is as shown below, which of the following virtual addresses will result in a page fault?

| Page | Page Frame    | Valid? |
|------|---------------|--------|
| 0    | 4             | Yes    |
| 1    | 10            | Yes    |
| 2    | Not in memory | Yes    |
| 3    | 0             | Yes    |
| 4    | 3             | Yes    |
| 5    | 0             | No     |
| 6    | 0             | No     |

- (a) 0
- (b) 5000
- (c) 10000
- (d) 20000
- (e) 50000

20. Which of the following code fragments is bug-free, i.e. which code fragment does NOT have a bug which causes unallocated memory to be overwritten? (All code fragments are syntactically correct and will compile successfully.)

(a) 

```
int *p;
int ar[100];
int i =0;
p = ar;
for(i = 0; i < 100; i++) {
    *p++ = i;
}
```

(b) 

```
int* p;
int i;
p = malloc(10 * sizeof(int));
for (i = 0; i <= 10; i++) {
    p[i] = i;
}
```

(c) 

```
int *p;
int ar[100];
int i;
p = &ar[0];
for(i = 0; i < 100; i++) {
    p[i*sizeof(int)] = i;
}
```

(d) 

```
char* p;
p = malloc(12 * sizeof(char));
strcpy(p, "Apr 24, 2009");
```

(e) All of the code fragments above have memory overwrite bugs.