

Computer Science 2211a Midterm Examination **Sample Solutions**

9 November 20XX

1 hour 40 minutes

Student Number: _____	
Surname: _____	Given name: _____



Instructions/Notes: The examination has **35 questions** on **9 pages**, and a total of **110 marks**. Put all answers on the question paper.

This is a closed book exam. **NO ELECTRONIC DEVICES OF ANY KIND ARE ALLOWED.**

1. [4 marks] Which of the following Unix commands/utilities are *filters*? **Correct answers are in blue.**

mkdir cd nl passwd grep cat chmod scriptfix mv

2. [1 mark] The Unix command **echo HOME** will print the contents of the environment variable whose name is **HOME**.

True False

3. [1 mark] In C, the *null character* is another name for the *null pointer*.

True False

4. [3 marks] The protection code for the file **abc.dat** is currently **-rwxr--r--**. The command

chmod a=x abc.dat

is equivalent to the command:

- a. **chmod 755 abc.dat**
- b. **chmod 711 abc.dat**
- c. **chmod 155 abc.dat**
- d. **chmod 111 abc.dat**
- e. **none of the above**

5. [3 marks] The protection code for the file **abc.dat** is currently **-rwxr--r--**. The command

chmod ug+w abc.dat

is equivalent to the command:

- a. **chmod 766 abc.dat**
- b. **chmod 764 abc.dat**
- c. **chmod 754 abc.dat**
- d. **chmod 222 abc.dat**
- e. **none of the above**

6. [3 marks] The protection code for **def.dat** is currently **dr-xr--r--** , and the protection code for **def.dat/ghi.dat** is currently **-r-xr--r--** . Give one or more **chmod** commands that will set the protections properly so that the owner of the two files will be able to delete **ghi.dat** using the command **rm def.dat/ghi.dat**

chmod u+w def.dat or chmod -r u+w def.dat

7. [1 mark] In Unix, the *relative path name* for the parent directory of the current working directory is **..** (that is, two periods).

True False

8. [1 mark] Absolute path names for files in a Unix system always begin with the character **'/'**.

True False

9. [2 marks] Suppose that the current working directory contains the regular files

a.dat a.txt b.dat

Then the command **mv *.dat** will:

- a. Copy the contents of **a.dat** into **b.dat** so that the two files are identical
 - b. Append the contents of **a.dat** to **b.dat**
 - c. Replace the contents of **b.dat** with the contents of **a.dat**, and get rid of the file **a.dat**
 - d. Fail because the **mv** command expects two files, a source and a destination
 - e. Fail because the destination file is not a directory
10. [2 marks] Describe the output produced by the command pipeline **cat * | sort -ru**
- All lines in the files in the current working directory are concatenated together and sorted in reverse ascii order; then duplicate lines are removed, and the results are displayed.
11. [1 mark] In Unix, the *root directory* is another name for your *home directory*.

True False

12. [4 marks] Suppose that the current working directory contains the files

abc def ghi

Describe exactly what happens when the command

rm * *

is executed.

First, wildcards are expanded by the shell to yield

rm abc def ghi abc def ghi

Then the **rm** command runs: the three files are deleted; then we get three error messages concerning the attempts to delete three files that are no longer found in the directory.

13. [4 marks] Give a Unix command that produces as output a file **newfile** that is the same as the file **oldfile**, except that all occurrences of the letters **b** and **e** have been replaced by the letter **k**.

tr be kk < oldfile > newfile

14. [4 marks] Give a Unix command that produces as output a file **newfile** that is the same as the file **oldfile**, except that characters that are not lower case letters have been removed.

tr -cd '[a-z]' < oldfile > outfile

15. [2 marks] The three Unix commands

grep hello *
grep 'hello' *
grep "hello" *

all produce the same output.

True False

16. [2 marks] Give a Unix command that will change the current working directory to your home directory on *gaul*, no matter where on the system you happen to be working.

cd or **cd \$HOME** or **cd ~**

17. [3 marks] Give a single Unix command that will delete the subdirectory **subdir** from the current working directory, along with all the contents of the subdirectory.

rm -r subdir or **rm -R subdir**

18. [2 marks] Assume that **abc.dat** is a regular file in the current working directory. Describe the output produced by the command

echo cat abc.dat

The output is:

cat abc.dat

19. [2 marks] Assume that **abc.dat** is a regular file in the current working directory. Describe the output produced by the command

echo `cat abc.dat`

The contents of file **abc.dat** are displayed on the screen.

20. [2 marks] Suppose that the current working directory contains the files

abc def ghi

Describe the output produced by the command

ls -l > /dev/null

is executed.

There is no output. Standard output has been redirected to the null device, and it disappears completely.

21. [3 marks] The output from the command **grep** `^[a-z]$` **file** consists of
- a. All lines in **file** that contain at least one lower-case letter
 - b. All lines in **file** that consist entirely of lower-case letters
 - c. All lines in **file** that consist of exactly one lower-case letter
 - d. All lines in **file** that contain no lower-case letters
 - e. All lines in **file** that end with a character that is not a lower-case letter
22. [3 marks] The output from the command **grep -v** `^[^a-z]$` **file** consists of
- a. All lines in **file** that contain no lower-case letters
 - b. All lines in **file** that contain at least one character that is not a lower-case letter
 - c. All non-empty lines in **file** that end with a lower-case letter
 - d. All empty lines in **file**, plus all lines that end with a lower-case letter
 - e. All lines in **file**

For questions 23 through 27, describe in your own words the output produced by the given **grep** command. (If your description is not clear, you will lose marks.)

23. [4 marks] **grep -c** `[a-z].*[a-z]` **file**

Prints the number of lines from **file** that contain at least two lower-case letters.

24. [4 marks] **grep -vi** `\([a-z]\).*\1` **file**

Prints all the lines from **file** on which no letter is repeated. Case distinctions are ignored.

25. [4 marks] **grep** `\<the$` **file**

Prints all the lines from **file** that end with the word **the**

26. [4 marks] **grep -v** `^$` **file**

Prints all the non-empty lines from **file**

27. [4 marks] **grep** `^$.*` **file**

Prints all lines from **file** that begin with a dollar sign.

28. [3 marks] Show a C variable declaration that creates a two-dimensional array of integers called **table**; **table** must have **2** rows and **3** columns, and all array locations must be initialized to **7** by the declaration.

```
int table[2][3] = { {7, 7, 7}, {7, 7, 7} };
```

For questions 29 through 31, suppose we have a C program that begins:

```
#include <stdio.h>
#include <stdlib.h>

char * fn1( int *a, int *b, int c );

int main( ) {
    int arr[6] = {2, 3, 5, 7, 11, 13};
    int j = 9, k = 26;
    int *p = &j;
    int *q = &arr[3];
    char *c;
```

29. [2 marks] `c = fn1(arr, q, arr[0]);` is a syntactically correct statement if included as the next line of the main program.

True **False**

30. [2 marks] `printf("%c\n", *(fn1(p, &k, *q)));` is a syntactically correct statement if included as the next line of the main program.

True **False**

31. [2 marks] `if (c = fn1(&arr[3], *p, k)) j++;` is a syntactically correct statement if included as the next line of the main program.

True **False**

32. [7 marks] Suppose that the function **strlen()** from **<string.h>** does not exist. Write your own version. (You may call other functions from the C libraries if you wish.)

```
size_t strlen( const char * s ) {

    /* return the number of characters found in s */

    size_t k;
    for (k = 0; s[k] != '\0'; k++); /* no body in this loop */
    return k;
}

size_t strlen( const char * s ) {

    int count = 0;
    while( *s != '\0' ) {
        s++;
        count++;
    }
    return count;
}
```

33. [7 marks] Suppose that the function **strdup()** from **<string.h>** does not exist. Write your own version. (You may call other functions from the C libraries if you wish.)

```
char * strdup( const char * s )

    /* return a dynamically allocated copy of the string referenced by s; if
    there is no memory available, return a null pointer instead */

    char * duplicate;
    duplicate = (char *) malloc ( (strlen(s) + 1) * sizeof(char) );
    return strcpy( duplicate, s);
}

char * strdup( const char * s ) {

    char * duplicate;
    int k;
    duplicate = (char *) malloc ( (strlen(s) + 1) * sizeof(char) );
    for (k = 0; k <= strlen(s); k++)
        duplicate[k] = s[k];
    return duplicate;
}
```


34. [7 marks] Write an integer-valued C function called **countDivisors()** that takes a positive integer **k** as a parameter, and that returns as its value the number of integers that divide evenly into **k**. (For example, if **k** is **12**, the function would return **6**, since **12** has **6** divisors: **1, 2, 3, 4, 6** and **12**.)

```
int countDivisors( int k ) {  
  
    int count = 0;  
    int j;  
    for ( j = 1; j <= k; j++ )  
        if ( k % j == 0 )  
            count++;  
    return count;  
}
```

35. [7 marks] Rewrite the function **countDivisors** from the previous question so that the function is of type **void**, and the result is passed back to the calling module through a second parameter. Include the function header in your answer.

```
void countDivisors( int k, int * result ) {  
  
    int j;  
    *result = 0;  
    for ( j = 1; j <= k; j++ )  
        if ( k % j == 0 )  
            *result++;  
}
```