Question 1:

What does the following script display as standard output?
(if you do not recall the precise format of the output of any specific
command, that's OK; just make up something with the appropriate flavor.)
(the following program runs without error)

```
rm -f cat 1 2 3 4 5 6 7 8
echo 1 2 3 4 > cat
echo 2 > 6
for cat in `cat cat`
do
        dog=`expr $cat + $cat`
        echo $dog | cat
        if test -r $dog
        then
                break
        fi
done
```

Filesystem:
- ↳ removes any files named '1', '2', '3' etc
- ↳ creates (or over-writes) file named 'cat', new contents are '1 2 3 4'
- ↳ creates a file named '6' contents are '2'

Loops over output of command 'cat cat' = 1 2 3 4. Each loop iteration variable 'cat' takes one of these values.

Iter 1: • dog=`expr 1+1` → dog=2
- • echo $dog | cat → prints 2
- • if → checks if file '2' is readable. It is not.

Iter 2: prints 4

Iter 3: prints 6 and breaks the loop because file '6' does exist.

Final Answer: 2 4 6

Question 2:

Part a) Define in 1-2 sentences the concept of "super user" (also known as root).

The super user has complete permissions to write/read/execute any file and to perform administrative tasks such as creating new users and changing file ownership.

Part b) Define in 1-2 sentences the concept of a "pipe" in Unix.

A pipe directs the output of command A to the input of command B (when run as $ A | B) using a virtual file.

Question 3:

Explain each of the following lines and what it does when they are run in order.
List the line (a,b,c,d) in your answers.
Be as precise and specific as you can be.
(if you do not recall the precise format of the output of any specific
command, that's OK; just make up something with the appropriate flavor.)

```
a) ALPHA='`date`'
b) rm *[ALPHA]*
c) yes yes | head -20 > ALPHA
d) wc ALPHA | tail -1
```

a) Single quotes cause back-ticks to be interpreted literally. ALPHA holds `date`, not a string like "Tue Oct 20..."

b) Removes any file with an A or L or P or H in its filename.

c) Creates or over-writes file named "ALPHA" with contents 20 lines all "yes"

d) wc ALPHA outputs " 20 20 80 ALPHA", tail -1 does nothing here.

Question 4:

In the C language, what value does the variable x take on?

```
main() {
    int x;
    int y;
    y = 'a' - 'b';
    x =  y++  * 2;
}
```

as int: 97
as int: 98

Values not crucial, but know that each letter is one larger than previous → y holds -1

- y is incremented AFTER being used
- X = -1 * 2 = **-2**

Question 5:

Circle and number at least 3 different bugs/errors in either logic or syntax in the
following code fragment.
Provide a written explanation for each one below.

```
File f;
char *name="test.dat";
char *name2="test2.dat";
char a=getchar(); // User enters F or S to specify file
char *buffer;
if ( a = "F" ) {
        f=fopen(name);
} else f=fopen(name2);
fgets(buffer, 1024, f);
```

① Must be FILE *f;

② Should be ==

③ Should be 'F'

④ and ⑤ Missing open code such as "r"

⑥ buffer does not point to valid memory, might seg fault

Question 6:

What does this C program print on the terminal?

```
#include <stdio.h>

void muddle( char string[] ){

        string[0] = 'M' + 1;
        string[1] = string[1];
        string[2] = string[1];

        char temp = string[4];
        string[4] = string[5];
        string[5] = temp;
}

int main()
{
        char string[10] = "goedel";
        muddle( string );
        printf("%s\n", string );
}
```

0 1 2 3 4 5
g o e d e l
N
o
o
e
l
e

- muddle can modify the contents of string because char[10] array type is equivalent to pointer here.

  (it could not change the address held by variable "string" in main(), but don't be confused by this).

- muddle changes characters one at a time to form **Noodle**