Possible marks: 33

Instructions:

1. This is an individual exam. You can use code given to you by course staff this semester or code which you wrote yourself this semester.
2. Download the answer text file from blackboard.
3. Answer Questions 1–6 in that file (using plain ascii text).
4. Question 7 will require you to write one or more C files. The names required for these files are given in the question.
5. Submit your answers via blackboard.
   - Attach all of your answer files to the same submission (Don't submit once for each file).
   - Do not change the names of the files.
   - You may submit multiple times (within the timelimit) but each submission should be complete.
6. If you wish to record assumptions you relied upon, you may do so at the end of the answer file.
   - Do not use this part as an alternative answer box.

Question1) Write shell commands to do the following: [2 marks (1 each)]

A) Show all entries in the current directory and their i-numbers

```
ls -ai
```

B) Remove group read permissions from the file called `dat7`

```
chmod g-r dat7
```

Question2) Write C to declare `foo` as . . . : [2 marks (1 each)]

A) An array of seven small positive integers.

```
unsigned short foo[7];
```

```
unsigned uint8_t foo[7];
```

B) An array of seven pointers to functions. Each function returns a string and takes no parameters.

```
char* (*foo[7])(void)
```

```
typedef char* (*fn)(void)
fn foo[7]
```

Question3) [4 marks (2 each)]

Pages on this system are 8KiB.

Part of the page table for process A is

Page	Frame
19	5
20	19
21	20
22	8

Part of the page table for process B is

Page	Frame
37	50
38	20
39	12
40	3

A) What physical address corresponds to virtual address 180279 in process A?.

B) What virtual address (if any) in process B corresponds to the virtual address 172089 in process A.

Question4) [4 marks]

Consider the following directory listing:

```
-rwxr-xr-x 2 me media 1100664 Aug 11 14:12 alt-debug
lrwxrwxrwx 1 me all         3 Aug 11 12:19 arj -> tar
lrwxrwxrwx 1 me all         3 Aug 11 14:20 compress -> arj
lrwxrwxrwx 1 me all         4 Aug 18 11:15 debug -> xxdb
-rwxr-xr-x 1 me all    138856 Aug 11 12:10 prog1
-rw-r----- 1 me staff     178 Aug 11 15:10 x.c
-rwxr-xr-x 2 me media 1100664 Aug 11 12:12 xxdb
lrwxrwxrwx 1 me all         8 Aug 11 23:18 zip -> compress
lrwxrwx--- 1 me all         8 Aug 11 12:17 tar -> zip
```

A) There are some things wrong with the above listing. What are they? [2 marks]

the permission is missing in the symb link tar->zip, should be all

there is a loop from tar->zip ... arj->zip of symb link
the system will allow us to create it, but it is wrong, we can
interact with that, if do, we will get an error

B) A system has the following ordinary users and groups (and no others): [2 marks]

User	Groups
henri	bakery staff
alphonse	staff
edward	bakery
hughes	

Consider the following directory listing:

```
-r-x---r-- 1 henri   staff  120303 Jul 22 07:45 abc
-r----x--- 1 alphose staff  120303 Jul 22 14:21 run
--w----r-- 1 edward  bakery 453201 Jun 11 07:10 tro
```

Which **users** are allowed to:

1. Read from `abc`?

   henri, edward, hughes

2. Execute `run`?

   henri

Question5) [5 marks]

A) [2 marks]

host1 10.3.9.131
host2 10.3.21.78
host3 10.3.69.57

Give the CIDR representation for the smallest possible network which includes host2 and host3 but excludes host1.

min: 10.3.21.78
max: 10.3.69.57
21: 00010101
69: 01000101
length of the subnet mask: 16+1=17
11111111.11111111.10000000.00000000
Sm: 255.255.128.0
Na = AND(255.255.128.0, 10.3.69.57) = 10.3.0.0
10.3.0.0/17

not possible

B) [3 marks]

Which of the following networks are invalid for use on the public internet and why?

Network	subnetmask
10.15.45.0	255.255.196.0
192.167.12.0	255.255.254.0
8.17.55.0	255.255.244.0
130.102.43.128	255.255.255.128

10.15.45.0 (non-routable network)
8.17.55.0
rules for subnet mask should be contiguous one
but 244: 11110100 is not.
196: 11000100 is not either

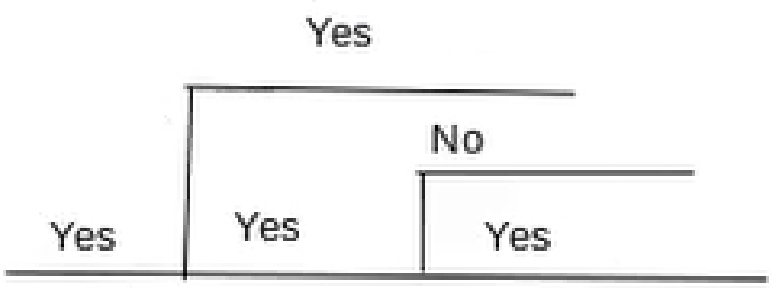
Question6) [4 marks]

A) Consider the following function:

```
void f(void) {
    int fds[2];
    pipe(fds);
    if (fork()) {
       if (fork()) {
           close(fds[0]);
           return;
       } else {
           close(fds[1]);
           return;
       }
       close(fds[0]);
    }
}
```

After a call to `f()` has completed, how many processes have the write end of the pipe open?

2 process
close(fds[0]) #close read
close(fds[1]) #close write



tes mean write still open

Question7) [12 marks]

In this question you will be required to write one or more c programs described below. There will be constraints on what the files are to be called and which functions you are not allowed to use.

You can develop the programs on your own machines if you wish. However, they must compile and run on moss for marking purposes. You must attach each file to your blackboard exam submission.

Write two files:

- `calc.c` : `gcc calc.c -std=gnu99 -pedantic -o calc`
- `check.c` : `gcc anag.c -std=gnu99 -pedantic -o check`

calc a op b

`a` and b are integers and if op is one of `+, -, *, %` then calculate the result and output it to standard out followed by a newline. If op is `%` and b is zero, ouput `"NaN"` followed by a newline. If any of the following occur:

- insufficient arguemnts are given
- `a` or `b` is not an integer
- op is not one of `+, -, *, %`

then output `"ERR"` followed by a newline.

check s1 s2

Note: String s1 is an anagram of string s2 if and only if there is a way to rearrange the characters in s1 to form s2. eg: `"ACC"` is an anagram of `"ACA"`. `"ACBB"` is not an anagram of `"BACC"`

Read lines of text from standard in until EOF. If the line (not including the newline) is:

- an anagram of both, then print `"12\n"` to standard out.
- an anagram of s1, then print `"1\n"` to standard out.
- an anagram of s2, then print `"2\n"` to standard out.
- None of the above, don't print anything.

(Use the first option in the list which matches).

You may detach this sheet.
Do not record answers on this sheet.

Example addresses:

broadcast	130.102.17.255
netmask	255.255.255.0
CIDR	130.102.17.0/24

%c	character
%d	integer
%u	unsigned integer
%lf	double (scanf)
%f	double (printf)
%p	void pointer
%ld	long integer

```
grep [-v] [ $ ^ . * ]
ls [-ladi]
ps [-ef]
sort [-r -k]
uniq [-c]
cat
head [-]
tail [-]
cut [-f -d]
wc [-l]
diff
svn
chmod
ln [-s]
rm [-rf]
mkdir
rmdir
cp [-r]
mv
vim/nano
less
```

Layers
link
application
onion
physical
network
gooey caramel
transport

KiB =	2^{10} Bytes
MiB =	2^{20} Bytes
GiB =	2^{30} Bytes

[ ]	array access
.	member selection
->	follow and select
++ --	
sizeof	
~	bitwise not
!	logical not
+ -	unary forms
&	address of
*	follow
( )	cast
* / %	
+ -	binary forms
> < <= >=	
== !=	
&	bitwise AND
^	bitwise XOR
|	bitwise OR
&&	logical AND
||	logical OR
? :	ternary operator
= += -= ...	
,	comma operator