9 Security I (MGK)

(*a*) Briefly explain *return-oriented programming*: what kind of software vulnerability and countermeasure does this class of attacks target, how does it work, and under what conditions is it applicable? [6 marks]

(*b*) Identify and fix a potential vulnerability in the following C function: [2 marks]

```
#include <stdlib.h>
void *bitmalloc(size_t bits) {
  return malloc((bits + 7)/8);
}
```

(*c*) On a Linux file server, you find this file:

```
$ ls -l
-rw----r-- 1 frank students  13593 May 31 14:55 question.tex
```

User `frank` is a member of group `students`.

(*i*) Based on the POSIX access-control settings shown, illustrate how the server's operating system will authorize access (if-statement pseudo code). [3 marks]

(*ii*) What does an equivalent Windows NTFS access-control list look like? [3 marks]

(*iii*) Does the Windows GUI for manipulating NTFS access-control lists allow users to enter this configuration? [2 marks]

(*d*) Give an example of how POSIX file-system access control can be used to provide the equivalent of password protection for parts of the file space. In particular, show how user `alice` can set up a directory `papers` such that only those members of group `committee` (which includes `alice`) who know the secret string "`SEL-4sB3`" can read a file `restricted.pdf`. Show the setup either as a sequence of shell commands that `alice` can use to create it, or in the form of the metadata of the files and directories involved (as `ls -l` would output it). [4 marks]