COMP500 / ENSE501: Week 11 – Exercise:

EXERCISE NAME: *Bit Flags and* **enum**

The following program source code is incomplete:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

// TODO: 1) Set powers of two for enum constants:
typedef enum eUser_Permissions
{
    VIEW,
    ADD,
    EDIT,
    DELETE
} User_Permissions;

void print_permissions(char* username, int permissions);

int main(void)
{
    // TODO: 2) Assign permissions:
    int xinyu = 0;
    int jade = 0;
    int steffan = 0;

    print_permissions("Steffan", steffan);
    print_permissions("Xinyu", xinyu);
    print_permissions("Jade", jade);

    return 0;
}

void print_permissions(char* username, int permissions)
{
    printf("%s's permissions (%d): ", username, permissions);

    // TODO: 3) Detect permissions:

    printf("\n");
}
```

At **`// TODO: 1)`** set the enumerated constant values to be powers of two, instead of the default 0, 1, 2 and 3.

At **`// TODO: 2)`** set each user permissions variable (**`xinyu`**, **`jade`** and **`steffan`**) based upon the following table:

User	*View*	*Add*	*Edit*	*Delete*
Steffan	Yes	Yes	Yes	Yes
Jade	Yes	Yes	Yes	No
Xinyu	Yes	No	No	No

At **`// TODO: 3)`** implement a sequence of **`if`** statements that detect whether a permission is set in the **`permissions`** parameter using bit masking. If a permission is set, the function should print out the equivalent text.

An example of the completed program's output is as follows:

```
Steffan's permissions (15): View Add Edit Delete
Xinyu's permissions (1): View
Jade's permissions (7): View Add Edit
```

Ensure the program output is exactly as described, and that the whitespace of your source code is well formatted.