Programming in C and C++

A C programmer makes use of the `goto` construct as follows:

```c
int test() {
    int x=0, y=0, i, j;
    int err=0;
    if ((y=init())==-1)
        goto error;
    for (i=1; i<10; i++) {
        for (j=1; j<10; j++) {
            if ((x=process(i,j))==-1) {
                err = 10*i+j;
                goto error;
            }
            y += x;
        }
    }
    return y;
}

error:
    printf("Something went wrong: %d %d\n", err/10, err%10);
    exit(1);
```

(a) Rewrite the code in C, maintaining the same functionality but avoiding the use of `goto`.  

(b) By defining a suitable C++ class to contain the error parameters i and j, rewrite the above code using C++ exceptions.

(c) Write a definition in C or C++ for a function `concat` that takes two strings `s1` and `s2` and returns a `char` pointer to heap memory containing a copy of the concatenation of `s1` and `s2`.

(d) Write a macro `CONCAT` that takes two string literals as arguments and results in them being concatenated into a single string after the preprocessor has run.

(e) Give two reasons why the following code is wrong:

```c
#define b "UoCCL"
char a[] = "UoCCL";
char i[] = CONCAT(b,a);
char j = concat(a,b);
```

and outline the key differences between `CONCAT` and `concat`.