C Error Handling Answers

Sayak Haldar
IIEST, Shibpur

1) Answer is d) All of the mentioned

2) Answer is b) They check for output errors

3) Answer is b) stdout

Explanation: stderr is not exactly same as stdout, but similar in the sense that both puts the output or error in the output screen (terminal in case of linux based operating system)

4) Answer is a) The dianositc ouput is directly displayed in the output

Explanation: In case of stderr, the diagnostic output is directly displayed in the output

Whereas, in case of stdout, the diagnostic output iss pipelined to the output file

In case of stdout, a buffer is maintained. The buffer content only is printed when a '\n' character is found at the end of the current instance of buffer.

5) Answer is b) exit(expr)

6) Answer is b) 0.

Explanation: Since there is no error in the stream fp.

7) Answer is b)
Explanation: exit(expr) (Since, within main)

8) Answer is d) None of the mentioned

Explanation: First, it prints all the content of newfile1.txt completely to the screen then it will print some garbage characters additionally
More Explanation:

feof function tests the end-of-file indicator for the given stream.

Now, when is the EOF flag set?

The EOF flag is set when the user attempts to read past the EOF. Thus, a read of the last character in the file does not turn the flag on. A subsequent read attempt reaches the EOF.

```
#include <stdio.h>
int main()
{
    FILE *fp;
    char c;
    int n = 0;
    fp = fopen("newfile1.txt", "r");
    while (!feof(fp))
    {
        c = getc(fp);
        putc(c, stdout);
    }
    return 0;
}
```

Now, chek this program. Suppose, the last character is read by getc. Now as it is mentioned, a read of the last character in the file does not turn the flag on. So, if in next iteration of while loop, !feof(fp) will return 1(or true). So, some garbage character will be printed. And then EOF flag is turned on. So, while loop will no more iterate.

Whereas, the following program will run fine:

```
#include <stdio.h>
int main()
{
FILE *fp;
char c;
int n = 0;
fp = fopen("newfile1.txt", "r");
while (1)
{
c =getc(fp);
if(feof(fp))
{
        break;
}
putc(c,stdout);
}
return 0;
}
```

9) Answer is b) hello

10) Answer is c) Whatever user says

References:

1) http://www.sanfoundry.com/c-interview-questions-answers/

2) http://stackoverflow.com/