1. Divide by zero error in C

Most common error that we face in C is divide by zero error and many times we stuck into this while programming.

C does not have any special function or class to deal with such type of error but we can simply overcome it by checking the divisor before the division.Let us understand by following example. Let's see when does divide by zero error comes.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
        int y=10/0;
        printf("result is: %.2f", y);

}
```

2. No such file or directory

This error comes if we try to access or open a file that does not exist in the required location and following is code to display suitable error for that puropse.

```
#include <stdio.h>
#include <errno.h>
#include <string.h>

int main ()
{
   FILE *fp;
   fp = fopen("a.txt", "r");
    //perform operation on fp
   //we are opening a file that does not exist

   return 0;
}
```

3. Semantic Error

This error occurs when the statements written in the program are not adding meaning to the compiler.

```
#include<stdio.h>
#include<string.h>
int main()
{
  int a, b, c;
  a + b = c; //semantic error
```

```
        getchar();
        return 0;
}
```

4. Syntax error

Errors that occur when you do not follow the rules of writing C syntax is known as syntax errors. This is a compiler error which indicates something that must be fixed before the code can be compiled. All these errors are detected by compiler and thus are known as compile-time errors.Following is the example:-

```
#include<stdio.h>
#include<string.h>
int main()
{
   int x = 10;
   int y = 15;

   printf("%d", (x, y)) // semicolon missed
   getch();
   return 0;
}
```

5. Use of EXIT_SUCCESS ,EXIT_FAILURE to deal with errors

This error handling is used in C as it is considered a good approach to use exit() in function and show it's status ,i.e. the it is a failure or success by EXIT_SUCCESS and EXIT_FAILURE .EXIT_SUCCESS is a macro defined by 0 and EXIT_FAILURE is defined by -1 .Following is the example of the it.

```
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>

int main ()
{
        FILE * fp;
        fp = fopen ("filedoesnotexist.txt", "rb");

        if (fp == NULL)
        {
                printf("Value of errno: %d\n", errno);
                printf("Error opening the file: %s\n",
                                                    strerror(errno));
                perror("Error printed by perror");
        exit(EXIT_FAILURE);
        printf("I will not be printed\n"); //line below exit() is not executed
    }

    else
    {
        fclose (fp);
        exit(EXIT_SUCCESS);
        printf("I will not be printed\n");
    }
    return 0;
}
```