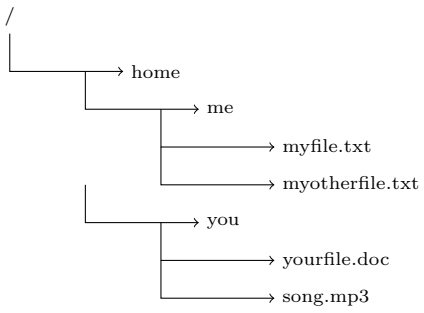
Consider the following code.

```
class Foo {
    Foo(File file) {
        this.file = file;
    }

    String doSomething(boolean c) throws IOException {
        String str = file.getName();
        if(c && file.isDirectory()) return file.getCanonicalPath();
        return str;
    }

    File file;

    public static void main(String [] args) throws FileNotFoundException, IOException {
        File file = new File(System.getProperty("user.dir"));    // => File("/home")
        boolean canon = false;

        for(int i = 0; i < args.length; i++) {
            switch(args[i]) {
                case "-c":
                    canon = true;
                    break;
                default:
                    file = new File(args[i]);
                    if(!file.exists())
                        throw new FileNotFoundException();
            }
        }

        Foo foo = new Foo(file);
        String str = foo.doSomething(canon);
        System.out.println(str);
    }
}
```

Assume that code above is being executed from `/home`, given the following directory structure.



Write the (one-line) output of each of the following calls. If an exception is thrown out of `main()` simply write the type of that exception. **Note:** `getName()` only gets the *last element in the path*.

```
java Foo -c -c   =>   /home
```

```
java Foo -cc   =>   FileNotFoundException()
```

```
java Foo me/myfile.txt   =>   myfile.txt
```

```
java Foo you/../you -c   =>   /home/you
```

```
java Foo -c you/../you/song.mp3   =>   song.mp3
```