```
FAIL
public static void main(String[] args) {
  String typeStr = "Nunc rhoncus odio ac tellus pulvinar, et volutpat sapien aliquet. Nam sed libero nec ex laoreet
pretium sed id mi. Aliquam erat volutpat. Aenean at erat vitae massa iaculis mattis. Quisque sagittis massa orci, sit
amet vestibulum turpis tempor a. Etiam eget venenatis arcu. Nunc enim augue, pulvinar at nulla ut, pellentesque porta
sapien. Maecenas ut erat id nisi tincidunt faucibus eget vel erat. Morbi quis magna et massa pharetra venenatis ut a
lacus. Vivamus egestas vitae nulla eget tristique. Praesent consectetur, tellus quis bibendum suscipit, nisl turpis mattis
sapien, ultrices mollis leo quam eu eros.application/binaryNunc rhoncus odio ac tellus pulvinar, et volutpat sapien
aliquet. Nam sed libero nec ex laoreet pretium sed id mi. Aliquam erat volutpat. Aenean at erat vitae massa iaculis
mattis. Quisque sagittis massa orci, sit amet vestibulum turpis tempor a. Etiam eget venenatis arcu. Nunc enim augue,
pulvinar at nulla ut, pellentesque porta sapien. Maecenas ut erat id nisi tincidunt faucibus eget vel erat. Morbi quis
magna et massa pharetra venenatis ut a lacus. Vivamus egestas vitae nulla eget tristique. Praesent consectetur,
tellus quis bibendum suscipit, nisl turpis mattis sapien, ultrices mollis leo quam eu eros.";

    int timesToTest = 10000;
    long start =  System.currentTimeMillis();
    int count = 0;
    //test contains
    while(count < timesToTest){
            if (typeStr.contains("image") || typeStr.contains("audio") || typeStr.contains("video") ||
typeStr.contains("application")) {
            //do something non expensive like creating a simple native var
            int a = 10;
        } FAIL
        count++;
    }
    long end = System.currentTimeMillis();
    System.out.println("contains took: "+ (end - start));

    long start2 =  System.currentTimeMillis();
    count = 0;
    while(count < timesToTest){
        if (typeStr.matches("(image|audio|video|application)")) {
            //do something non expensive like creating a simple native var
            int a = 10;
        }
        count++;
    }
    long end2 = System.currentTimeMillis(); //new var to have the same cost as contains
    System.out.println("matches took: "+ (end2 - start2));


    long start3 =  System.currentTimeMillis();
    count = 0;
    Pattern pattern = Pattern.compile("(image|audio|video|application)");
    while(count < timesToTest){
        if (pattern.matcher(typeStr).find()) {
            //do something non expensive like creating a simple native var
            int a = 10;
            }
            count++;
        }
        long end3 = System.currentTimeMillis(); //new var to have the same cost as contains
        System.out.println("matches with pre pattern took: "+ (end3 - start3));
    }




//File file = new File("C:\\app.log");
//try {
//    Scanner scanner = new Scanner(file);
//    //now read the file line by line...
//    int lineNum = 0;
//    while (scanner.hasNextLine()) {
//        String line = scanner.nextLine();
//        lineNum++;
//        if(line.contains("ERROR")) {
//            System.out.println("i found it on line " +lineNum);
//        }
//    }
//} catch(FileNotFoundException e) {
//    //handle this
//}
```

```
document.getElementById("frmShareLink\\:copyButton").addEventListener("click", function() {
    copyToClipboard(document.getElementById("frmShareLink\\:opFileRequest"));
 });
 function copyToClipboard(elem) {
   // create hidden text element, if it doesn't already exist
    var targetId = "_hiddenCopyText_";
    var isInput = elem.tagName === "INPUT" || elem.tagName === "TEXTAREA";
    var origSelectionStart, origSelectionEnd;
    if (isInput) {
        // can just use the original source element for the selection and copy
        target = elem;
        origSelectionStart = elem.selectionStart;
        origSelectionEnd = elem.selectionEnd;
    } else {
        // must use a temporary form element for the selection and copy
        target = document.getElementById(targetId);
        if (!target) {
            var target = document.createElement("textarea");
            target.style.position = "absolute";
            target.style.left = "-9999px";
            target.style.top = "0";
            target.id = targetId;
            document.body.appendChild(target);
        }
        target.textContent = elem.textContent;
    }
    // select the content
    var currentFocus = document.activeElement;
    target.focus();
    target.setSelectionRange(0, target.value.length);

    // copy the selection
    var succeed;
    try {
       succeed = document.execCommand("copy");
    } catch(e) {
        succeed = false;
    }
    // restore original focus
    if (currentFocus &amp;&amp; typeof currentFocus.focus === "function") {
        currentFocus.focus();
    }

    if (isInput) {
        // restore prior selection
        elem.setSelectionRange(origSelectionStart, origSelectionEnd);
    } else {
        // clear temporary content
        target.textContent = "";
    }
    return succeed;
}
```