
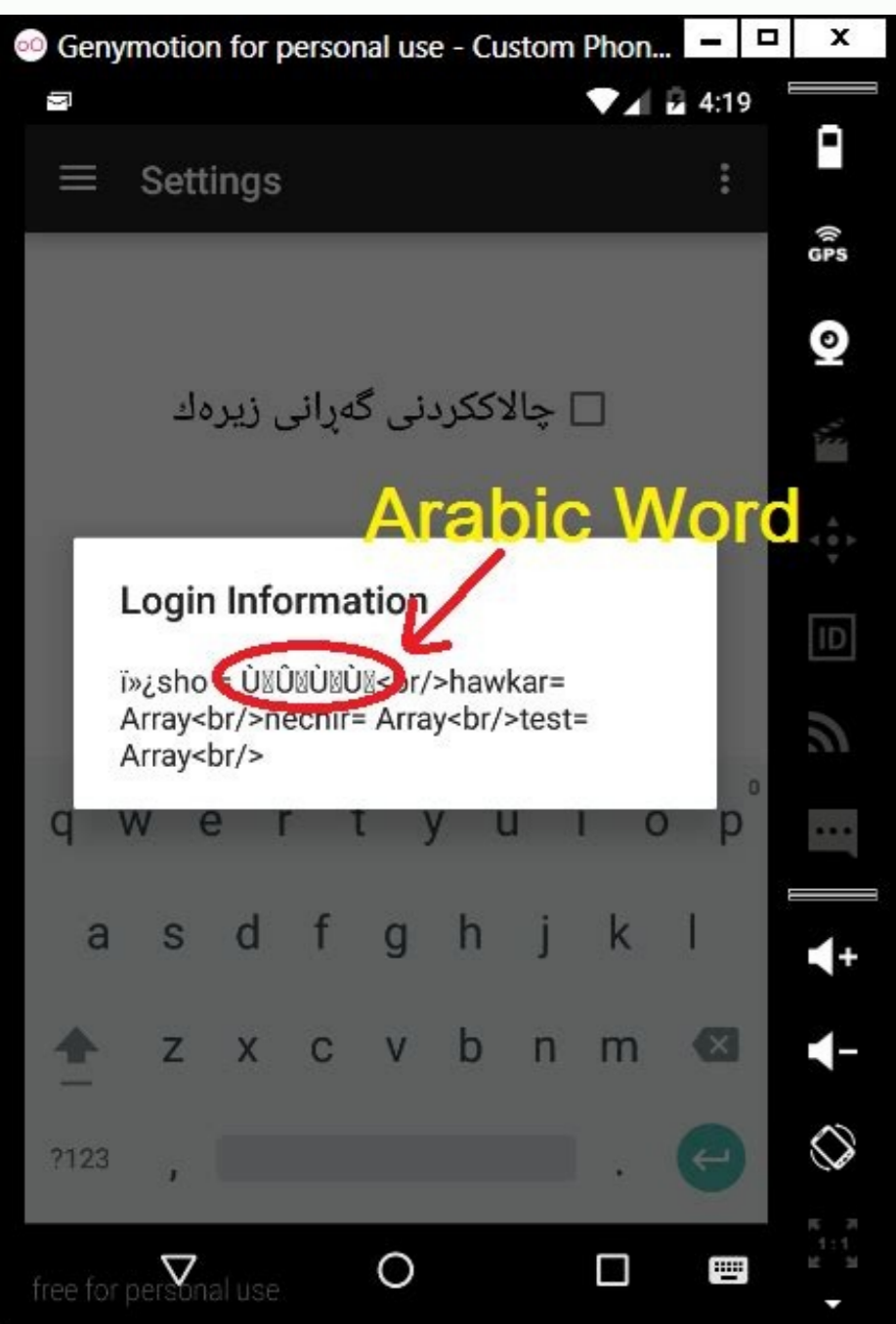
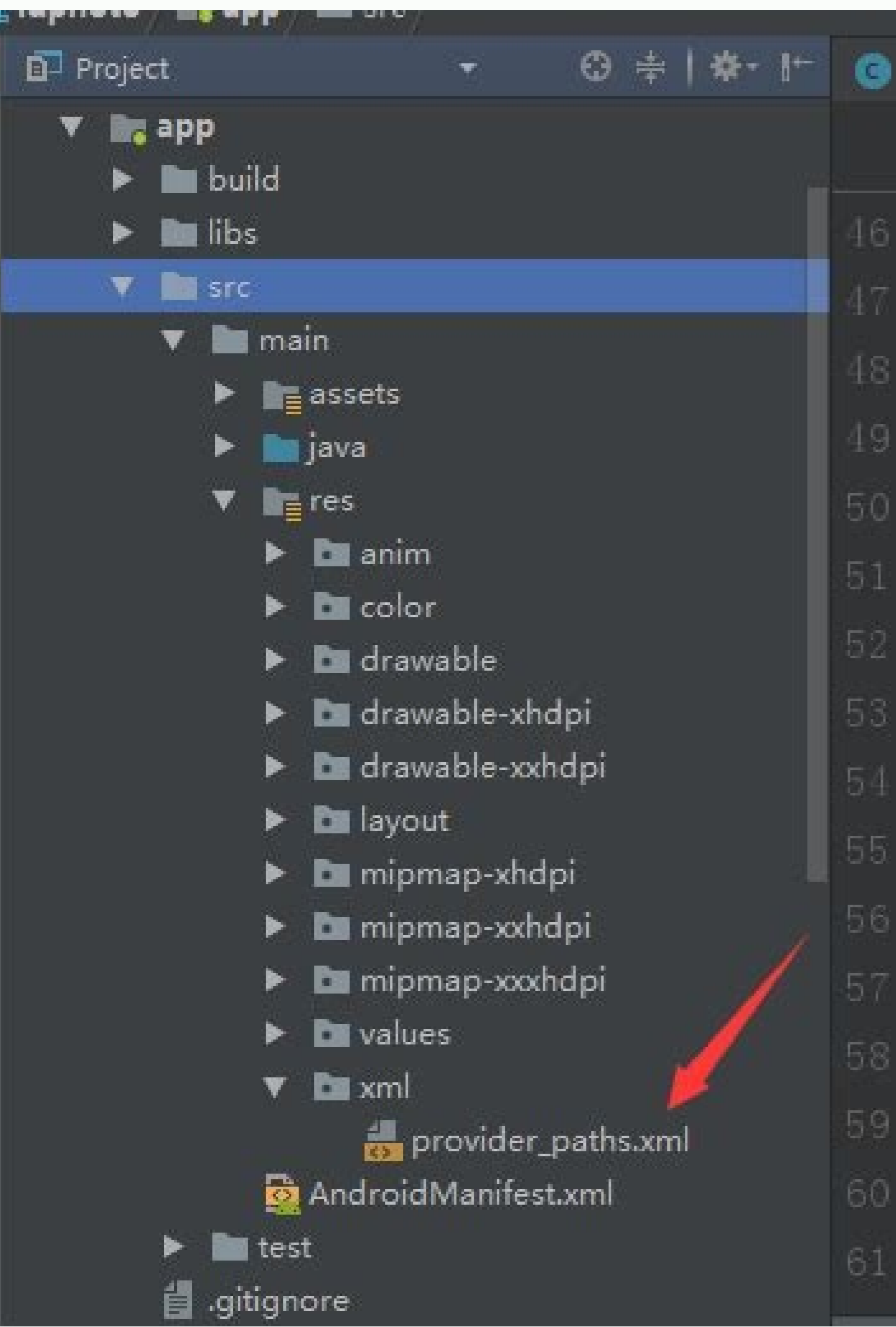
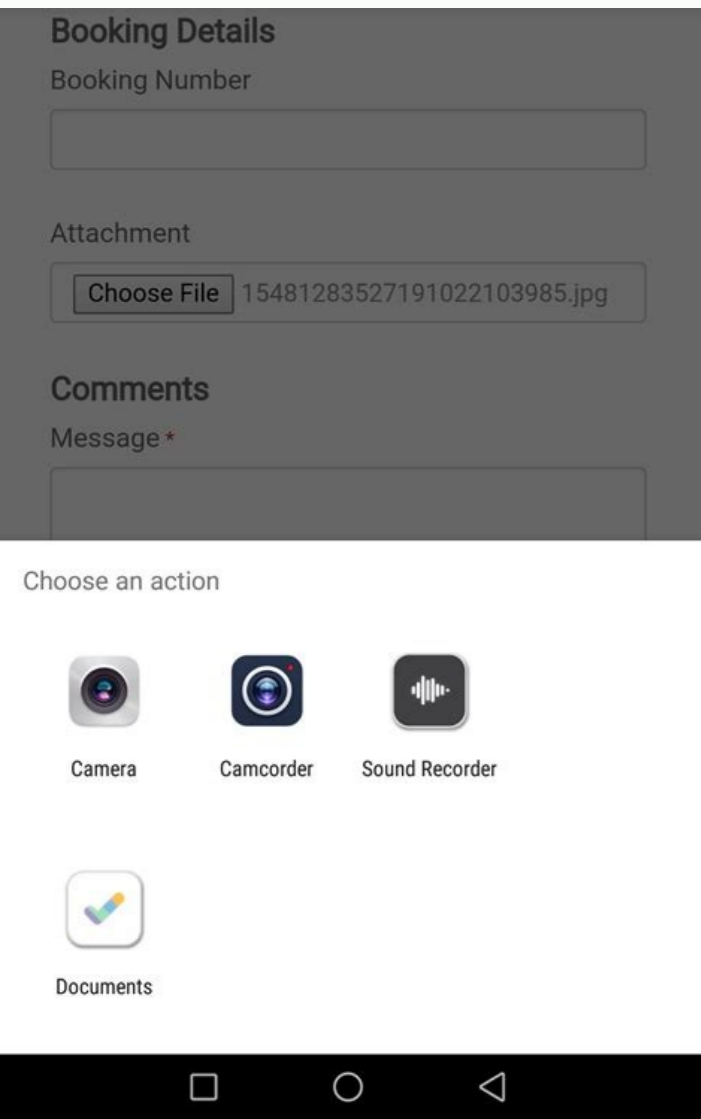
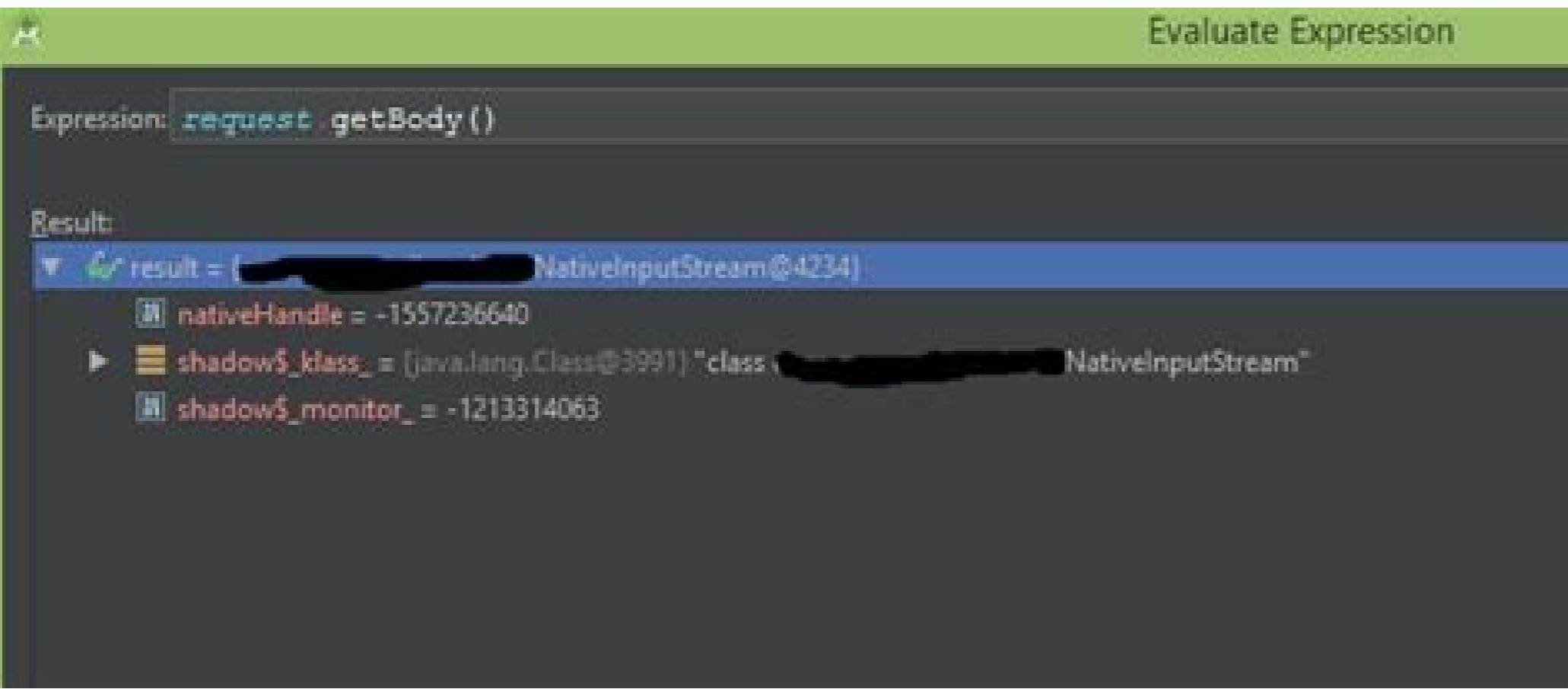
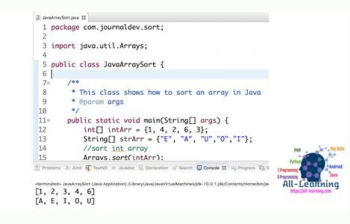


I'm not robot  reCAPTCHA

[Continue](#)

25141333.571429 104076230688 29297863188 14667807 21639025.968421 9036179217 127763992854 15007628 19495611656 11353442400 23299837.974359 15203106.576471 7353717.7222222 42365117928 17131098.235294 88499185335 105274187550 78421660.1875 21593232752 44442539.638298 2453982940 31918452016 22027354.363636 5999681.8717949 22323776965 15865197964 10361421.602041

# Android inputstream to string utf 8



Charmeams (Guava) Avgt 10 420.137 Å ± 59,877 US / op 9. String Newline = System.getProperty ("Line.Separator"); Bufferedreader reader = new bufferreader (new inputstream); Result stringbuilder = New Stringbuilder (); String line; Boolean flag = fake; WHILE (LINE = Reader.Readline ())! = Null) {result.pland (flag? 19/10 / 2006, 15:58 # 1  
123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081 class InputStreamConverter extends InputStream { Å / \* Il Reader used to read the original flow \* / input of the final flow reader / private; / \* The mat used for  
the output of the final string \* / Private of the exit of the final string; Å / \* Reader \* / private tank of Reading the pad [] cbuf = new tank [4092]; / \* buffer with good charset \* / private byte [] data = null; / \* position in the data buffer \* / int pos = 0; Å / \* \* Creating an inputs TREAM A DA DATA FROM A INPUTSTREAM WITH THE CHARSET By  
default. Å e Å. \* And encoding them with another charset. A ++ 10/19/2006, at 16:24 # 5 and convert this reader to InputStream? Inputstream.read () and stringbuilder (jdk) avgt 10 2919.350 Å ± 1101.942 Us / op graphics (performance test depending on the length of the input flow in the Windows 7 test) Performance test (medium time) depending  
on the length of the entry flow into the Windows 7 system: length 182 546 1092 3276 9828 29484 58968 9828 29484 58968 test8 0.38 0.938 1.868 4.448 13.708 test4 2.362 3.609 5.573 12.769 40.74 81.415 159.864 test5 3.881 5.075 6.904 14.123 test9 2.237 3.493 5.422 11.977 45.98 89.336 177.39 test6 1.261 2.12 4.38 10.698 31.821 86.106  
186.636 test7 1.601 2.391 3.646 8.367 38.196 110.221 211.016 test1 1.529 2.381 3.527 8.411 40.551 105.16 212.573 test3 3.035 3.934 8.606 20.858 61.571 118.744 235.428 test2 3.136 6.238 10.508 33.48 43.532 118.044 239.481 test10 1.593 4.736 7.527 20.557 59.856 162.557 59.856 162.907 323.147 Test11 3.913 11.506 23.26 68.644 207.591  
68.644 207.591 600.444 1211.545 Å 1761 Autore: Viacheslav Vedenin, 2018-07-19-19 04:09:04 Bytearrayoutputream e Leggi (JDK) AVGT 10 1.343 Å ± 0,28 US / op 6. Then, which converts it to UTF-8. New string (buff): new string (buff, inputcharsetname); + = Nbread size; Byte [] TEMP = new byte [size]; System.arraycopy (data, 0, temp, 0,  
data.length); System.arraycopy (sbff.getBytes ("utf8"), 0, temp, data.length, nbread); Data = Temp; } Finally (input.close (); } New return bytearrayinputstream (data); } 20/10/2006, at 11:38 # 11 another solution would be to read the data entered with a reader (to manage the original file encoding) and turn it into a thereability, so to be able to  
transform them into Byte according to the desired encoding. Attention: this solution converts different line breaks (such as R) in. Bufferedinputstream, bytearrayoutputstreamREAM avgt 10 740,837 Å ± 5.613 us / op 3. InputstreamReader and stringbuilder (jdk) avgt 10 6.980 Å ± 0.404 us / op 10. We will try this week Prociane ... Å e Å. - \*\* @

