|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LEARNING PROFILE FOR Chapter04Exercise02.hexValue(char) | | | | | |
| *Name* | *:* | *Tyler Lucas* | *Submission Date* | *:* | *N/A* |
| *Student ID* | *:* | *3305203* | *Date Created* | *:* | *2017-05-19* |

# Problem Statement

Chapter 4, Exercise 2

The hexadecimal digits are the ordinary, base-10 digits ’0’ through ’9’ plus the letters ’A’ through ’F’. In the hexadecimal system, these digits represent the values 0 through 15, respectively. Write a function named hexValue that uses a switch statement to ﬁnd the hexadecimal value of a given character. The character is a parameter to the function, and its hexadecimal value is the return value of the function. You should count lower case letters ’a’ through ’f’ as having the same value as the corresponding upper case letters. If the parameter is not one of the legal hexadecimal digits, return -1 as the value of the function. A hexadecimal integer is a sequence of hexadecimal digits, such as 34A7, ﬀ8, 174204, or FADE. If str is a string containing a hexadecimal integer, then the corresponding base-10 integer can be computed as follows:

int value = 0;

for ( int i = 0; i < str.length(); i++ )

value += value\*16 + hexValue( str.charAt(i) );

Of course, this is not valid if str contains any characters that are not hexadecimal digits. Write a program that reads a string from the user. If all the characters in the string are hexadecimal digits, print out the corresponding base-10 value. If not, print out an error message.

# Description of the Code

Tests overloaded method parseHex() that converts character sequences representing hexadecimal values into those (decimal) values. Tests include converting hex data file (hexdata.txt to hexoutput.txt), converting number sequence to hex and back again, and converting user input. Each instance of parseHex() uses the method int hexValue(char) to convert individual characters, as requested by the exercise.

# Errors and Warnings

No compiler errors encountered during development save those from missing semicolons or misplaced brackets. There were several data type and text parsing errors, as follows.

# Sample Input and Output

## Abridged Sample

### Input file “hexdata.txt”

0x000001

0x000002

0x000003

0x000004

0x000005

0x000006

0x000007

0x000008

0x000009

0x00000A

0x00000B

0x00000C

0x00000D

0x00000E

0x00000F

0x000010

0x000011

0x000012

0x000013

0x000014

0x000015

0x000016

0x000017

0x000018

0x000019

0x00001A

0x00001B

0x00001C

0x00001D

0x00001E

0x00001F

0x000020

0x000021

0x000022

0x000023

0x000024

0x000025

0x000026

0x000027

0x000028

0x000029

0x00002A

0x00002B

0x00002C

0x00002D

0x00002E

0x00002F

0x000030

0x000031

0x000032

0x000033

0x000034

0x000035

0x000036

0x000037

0x000038

0x000039

0x00003A

0x00003B

0x00003C

0x00003D

0x00003E

0x00003F

0x000040

0x000041

0x000042

0x000043

0x000044

0x000045

0x000046

0x000047

0x000048

0x000049

0x00004A

0x00004B

0x00004C

0x00004D

0x00004E

0x00004F

0x000050

0x000051

0x000052

0x000053

0x000054

0x000055

0x000056

0x000057

0x000058

0x000059

0x00005A

0x00005B

0x00005C

0x00005D

0x00005E

0x00005F

0x000060

0x000061

0x000062

0x000063

0x000064

0x000065

0x000066

0x000067

0x000068

0x000069

0x00006A

0x00006B

0x00006C

0x00006D

0x00006E

0x00006F

0x000070

0x000071

0x000072

0x000073

0x000074

0x000075

0x000076

0x000077

0x000078

0x000079

0x00007A

0x00007B

0x00007C

0x00007D

0x00007E

0x00007F

0x000080

0x000081

0x000082

0x000083

0x000084

0x000085

0x000086

0x000087

0x000088

0x000089

0x00008A

0x00008B

0x00008C

0x00008D

0x00008E

0x00008F

0x000090

0x000091

0x000092

0x000093

0x000094

0x000095

0x000096

0x000097

0x000098

0x000099

0x00009A

0x00009B

0x00009C

0x00009D

0x00009E

0x00009F

0x0000A0

0x0000A1

0x0000A2

0x0000A3

0x0000A4

0x0000A5

0x0000A6

0x0000A7

0x0000A8

0x0000A9

0x0000AA

0x0000AB

0x0000AC

0x0000AD

0x0000AE

0x0000AF

0x0000B0

0x0000B1

0x0000B2

0x0000B3

0x0000B4

0x0000B5

0x0000B6

0x0000B7

0x0000B8

0x0000B9

0x0000BA

0x0000BB

0x0000BC

0x0000BD

0x0000BE

0x0000BF

0x0000C0

0x0000C1

0x0000C2

0x0000C3

0x0000C4

0x0000C5

0x0000C6

0x0000C7

0x0000C8

0x0000C9

0x0000CA

0x0000CB

0x0000CC

0x0000CD

0x0000CE

0x0000CF

0x0000D0

0x0000D1

0x0000D2

0x0000D3

0x0000D4

0x0000D5

0x0000D6

0x0000D7

0x0000D8

0x0000D9

0x0000DA

0x0000DB

0x0000DC

0x0000DD

0x0000DE

0x0000DF

0x0000E0

0x0000E1

0x0000E2

0x0000E3

0x0000E4

0x0000E5

0x0000E6

0x0000E7

0x0000E8

0x0000E9

0x0000EA

0x0000EB

0x0000EC

0x0000ED

0x0000EE

0x0000EF

0x0000F0

0x0000F1

0x0000F2

0x0000F3

0x0000F4

0x0000F5

0x0000F6

0x0000F7

0x0000F8

0x0000F9

0x0000FA

0x0000FB

0x0000FC

0x0000FD

0x0000FE

0x0000FF

0x00FFFF

0x010000

0x010001

0x7FFFFFFF

0x80000000

0x80000001

0xFFFFFFFF

0x100000000

0x100000001

#0001UR

FFFRC6C1

OXFSARA9

0XEEA2AD

#GDMC9Q

OX8B5F65

OXLFCICL

0xFFB5C5

OXEEA9B8

OxYDHE9E

8BDB6UKC

0xDP7DN3

#FF82AB

EE799F

CD6889

0x8RS75D

#UFM0NK

#SJEFEN

OXCVKIC5

#8HK3OR

0xFF3E96

0xEUMA8C

0xCD3278

8B2252

OXFF69B4

0xFF6EB4

OxEERAT7

OXCA60H0

8B3A62

87BG2LIH

FF1493

#ER1OAO

0XCI107V

OXSK0A5T

OXFY3UBX

OxEE30A7

OXCD2990

0x8B1Y6F

OXC71585

0XD02090

#DA70D6

0xGF83AA

EEEE7RE9

#CD69C9

0X8B4789

D8BFD8

FFE1FF

OxEED2EE

OxASW5CY

0x8Q7B8E

0XFFBBFF

OxEEAEEE

OxCD96CD

#8B66JB

0XDDA0DD

EEKEEIEE

0xFR00FF

0xEE00EE

#CD00CD

0XNSG0GN

0xJV00WC

OxBUB5DA

0xE06IFA

OXO15ZEE

B4I45ECL

0X7AAI8F

0XG400PJ

OX9VFHTC

BFBX3EFS

0xB2VJEU

OX9A3GCQ

OX68D2PY

0xNBAMD2

0XXR2WGL

0X9BM0FN

Ox912CEE

7D26CD

#XZFAWP

Ox9370DB

0xAB82FF

Ox9F79GE

#CR68GD

OX5D478B

Ox483D8B

0x8470FF

7B7XHSPE

OX6ADACY

OXGA6FFU

7A67EE

#6959CD

Ox4O3CYB

OXFBF8IY

E6E6FA

000NIAFK

Ox0000EE

OXSL0ZCD

0x00008B

000EAREX

Ox191970

OXQJ59AB

0xZ16YAY

Ox4876FF

0x436EEE

0X3EGFZD

27408B

64QQ95HD

OXB0C4DE

0xDBEFFW

#BCFMEE

0XA2B5CD

Ox6E7B8B

777G8K9B

70708090

0xC6E2FF

0xB9DOGE

9FB6CD

0xDEFDIT

0X1E90FF

1C86EE

0X1874CD

0x10OWGN

0XT0FVCF

OxLCUHQP

63B8FF

Ox5CACEE

0X4FH4CD

#36648B

OX87CEFA

OXJVE2GF

OxIYZOEH

0xVTDTLD

6060MBCE

0X87CEFF

7E7EDNEK

0X6CA6CD

0x4J70XB

#8LYWXQ

OxDJMFAF

OX00B2EE

0xT0WAGX

00688B

33A1C9

0xADD8E6

0xBFEFFF

OxB2DFEE

OX9AC0CD

OX688A8W

OxB0E0U6

OX98F5FF

0x8EE5EE

7AC5CD

#HY8SZB

Ox00F5FF

#E0TXEE

OX00C5CD

#00868B

Ox5FZMHK

OX0BCEDD

0xF0FFFF

0xD0EGEE

OXC1WDQX

OX83WBUB

E0XEAOFF

0XD1EEEE

B4U4CDDD

0X7AMW8U

OXBBFFFF

#VEEEEE

0X96CDCD

OXT6YBJI

OXYF4NPF

#9PFFKF

0xVBEEIE

0xI9EGBA

52528RDF

00D0FUFF

OXDPEBVZ

0XV0CDCD

0000GO8B

008080

Ox48D1CC

OXUVN2AQ

030UY8TE

Ox40E0D0

0X808A87

Ox00C78C

7F7WFXNJ

76EEC6

#ICCDAA

458B74

00FA9A

0XP5BZFA

0XM0FHWE

#NREEID

OxMACDW6

008B45

0x3CB371

OxLZFFKF

#4EEE94

43MCCCOP

0X2E8B57

Ox00C957

0XBDFCC9

OxADBV40

0xF0FFF0

0xE0EAEZ

OXC1CDC1

OXAY8XNX

Ox8FBC8F

0xFPFFB1

#BKETXL

#9BCD9B

0xD98BDD

0x98FB98

9AFF9A

0x9OXY90

7C7LCD7M

0X54SB54

#32CD32

Ox22ZB22

0x00DF00

0xP0EZLP

0000CC0M

#00ZBSZ

0X008000

000IDFDC

OxY0PD14

0X7JXKW0

Ox7ONFMW

0x7UGE00

#66ADU0

0X458B00

ADADJJGU

OXCAFF70

OXBCEE68

A2CD5A

#6E8B3D

0XNHQDER

#6B8E23

OxC0FF3E

B3EE3A

Ox9ACD32

0XJ9OG2N

OxFFFFF0

OxELFCZW

#KONDCO

OxKBNBG3

#UOF5DU

OXYFMFE0

#EEIKN1

#CDCDBK

0xOB8JLA

0xFAFXHQ

0xFFFF00

OXEEEE00

0XCDCDQB

8B8B00

Ox8080XW

0x808000

BDBEST6G

#FFFR8E

0xOHR68J

OxCDC6QX

Ox8B864E

OXF0E68C

0xEEE8AA

0xFFFACD

OxEEE9BF

#CDCCA5

8B8VQ97J

0XFFEC8B

0XEEDC82

0XCDBE70

OXMQ814Z

#E3CL5V

#ZJU700

OXEEC900

OXCDAD00

8B8BJDKD

OxBYFHDZ

#EGECND

0xCYZ8BY

#JB88SN

OxDAA520

0xFAC125

OXEEH42Y

0XCD9R1D

#8B6914

#UO86GP

#JXB90E

OXEEAD0E

OxCD950C

0X8Y6E0B

0XFFA500

0XEE9V00

#CD8P00

0x8B5A00

OxFFFAF0

OXFDF5E6

OXF5DEBL

0XFFE7BA

#EED8AE

OXMDEZK6

Ox8B7E66

0xCVE4R5

#FFEFD5

0XFFEBCD

0xBFVAAK

OXEYCFA1

0xCDB38B

8BAM795E

0XDCETCS

OxDMBL8T

#9C661F

0XFF9912

0XOPRBR7

0XFGPPDP

EEEEWKCL

0xDDV0D0

#8B8378

OXTSNJ87

OxFFZZ9B

0xMMWW9G

OXCDAA7D

OX8BHC5D

0XFFE4C4

#EED5B7

#CARG9I

Ox8B7D6B

OXENU8UI

0XRD912W

0XFFWV0K

OXFF7F00

EEEE7600

0xCD6600

OX8UXZDS

FFFE800G

OxFEM5DW

0xEE9A49

0xCD853F

0XKP5AZB

FAF0E6

0xFSKAB9

OxEECKUR

CDAF95

OX8VW7YS

0xFFF5EE

0XEER5DE

0XCDC5BF

#RIG68C

OXF4ALVC

C76114

D2691E

0XCB7X2R

0XLXH6EF

0XCD661D

0x8B4513

0XH924ZZ

OxFF7D40

0xFF6103

8A360F

OXA0522D

OxFT824D

0XEE7942

#XDP8CM

0X8B47SN

OxFFA07A

OxEE9572

0xCUB16R

#8EQ74O

#RFYSIK

OxFU4500

0XEE4000

0xCDMZ00

8B8T2CA0

0X5E2612

OXE9967A

FFFL8FRT

OxEE8262

OXCD7054

0x8G4C39

OXFF7256

OxEE6A50

0XCD5B45

0xSK3E2F

0x8A3324

0xFF6347

0xEE5C42

#DHEN39

0xXG3K2D

FA8072

0xFFE4E1

OxUEU5U2

0XCDB7B5

0XKNZE7B

0XFFFAFA

EEE9E9

OXCDC9C9

#8B8989

0xBC8F8F

FFFFCPS1

OxEEB4B4

#CD9B9B

0XUY6969

OXJM80S0

0XCD5C5C

FF6A6A

0xGJR363

Ox8B3A3A

CD5555

OXA52A2A

FFFPYMHR

0xEE3B3B

0xCD3333

#8BL3Y3

OxB22W2A

OxFF3030

OXEE2C2C

CDCD26O6

0x8B1A1A

0XFF0000

OxOJFZM0

OxCQ0G00

0x8B0000

0x800000

#8E388E

OXPE71II

7D9EC0

0X388E8E

Ox71C671

0x8E8E38

OxC5R1AA

OXC67171

OxAKM5T5

1E1E1E1E

Ox28LR28

Ox5O515L

OX5B5B5B

0XG48EKK

#8E8E8E

OXAAAAAA

B7B7B7L7

0XV1OMCF

0XEAEAEA

OXF4K4FY

0XFRFFFN

F5F5F5

0XDGDCDC

D3Y3K3D3

C0VDD0C0

#A9A9A9

Ox8A8080

Ox696969

0X000000

FCSCFGQV

0xOWXCFA

0XP7K7F7

0XF5W5FL

F2JHF2FK

F0K0FET0

#IPGDIB

0xEBEBEB

OXL8E8EM

#R5EYG5

OXTATEY3

OxS0EKIT

0xDEDEDE

OxDBDBDB

OxD9D9D9

0XZ6D6RJ

0XO4BMDY

#D1Q1D1

OxCFCFCB

OXCPCCCL

#C9C9C9

OXC7C7C7

OXCLC4CS

#ZAC2RO

BFBFBF

#BGBOBD

#BABABA

OXZSSUGP

0XB5B5B5

#B3B3B3

#VKBFBK

0XADANAL

OXZDWHMJ

#D8A8YZ

OxAOAZSK

OXA3A3A3

#A1A1A1

Ox9E9E9E

#9C9C9C

OXZYV99I

0xG69RTP

0xNGX49R

#919191

0x8F8F8F

OX8CDCPA

8A8A8A

0x878VOZ

Ox858585

0X8S828E

Ox7F7F7F

OX7D7D7D

OX7A7A7N

0X787878

757G7Q7L

#WRE3NQ

#7J7C70

0X6E6E6E

OX6BJW6B

Ox696969

0x666666

OxJ3WWZN

Ox616161

OXIE5E5E

OX5C5C5C

#59R95W

0X575757

54B45JA4

0xNP52ZG

OxNF4M4I

0xVM4LBB

4A4A4A

OX474747

45E54JC5

0xB2424J

OxLE40I0

0xHDWT3V

OX3BEBIV

0XI8C83O

0X363636

OXL33J3O

0X3MR030

OX2E2E2L

2B2B2B

0X292929

Ox262626

OX242424

Ox212121

1F1ACL1F

#1FYHDC

0x1A1AMJ

Ox171717

0X141414

0X12DW12

#RFAF0D

0x0D0D0D

0X0A0OVW

080808

0XV5050C

0303030U

### Output file “hexoutput.txt”

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

[invalid hex data]

65535

65536

65537

[invalid hex data]

2147483647

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

0

1

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

15639213

[invalid hex data]

[invalid hex data]

[invalid hex data]

16758213

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

16745131

15628703

13461641

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

16727702

[invalid hex data]

13447800

9118290

[invalid hex data]

16740020

[invalid hex data]

[invalid hex data]

9124450

[invalid hex data]

16716947

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

13639824

14315734

[invalid hex data]

[invalid hex data]

13461961

9127817

14204888

16769535

[invalid hex data]

[invalid hex data]

[invalid hex data]

16759807

[invalid hex data]

[invalid hex data]

[invalid hex data]

14524637

[invalid hex data]

[invalid hex data]

15597806

13435085

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

8201933

[invalid hex data]

[invalid hex data]

11240191

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

8679679

[invalid hex data]

[invalid hex data]

[invalid hex data]

8021998

6904269

[invalid hex data]

[invalid hex data]

15132410

[invalid hex data]

[invalid hex data]

[invalid hex data]

139

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

4419310

[invalid hex data]

2572427

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

10663373

[invalid hex data]

[invalid hex data]

1886421136

13034239

[invalid hex data]

10467021

[invalid hex data]

2003199

1869550

1602765

[invalid hex data]

[invalid hex data]

[invalid hex data]

6535423

[invalid hex data]

[invalid hex data]

3564683

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

8900351

[invalid hex data]

7120589

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

26763

3383753

11393254

12578815

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

9364974

8046029

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

34443

[invalid hex data]

[invalid hex data]

15794175

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

13758190

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

9883085

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

32896

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

8424071

[invalid hex data]

[invalid hex data]

7794374

[invalid hex data]

4557684

64154

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

35653

3978097

[invalid hex data]

5172884

[invalid hex data]

3050327

[invalid hex data]

12451017

[invalid hex data]

15794160

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

10210715

14257117

10025880

10157978

[invalid hex data]

[invalid hex data]

[invalid hex data]

3329330

[invalid hex data]

57088

[invalid hex data]

[invalid hex data]

[invalid hex data]

32768

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

4557568

[invalid hex data]

[invalid hex data]

[invalid hex data]

10669402

7244605

[invalid hex data]

7048739

[invalid hex data]

11791930

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

16776960

[invalid hex data]

[invalid hex data]

9145088

[invalid hex data]

8421376

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

15657130

16775885

[invalid hex data]

13487269

[invalid hex data]

16772235

15654018

13483632

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

16433445

[invalid hex data]

[invalid hex data]

9136404

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

16753920

[invalid hex data]

[invalid hex data]

9132544

[invalid hex data]

[invalid hex data]

[invalid hex data]

16771002

15653038

[invalid hex data]

[invalid hex data]

[invalid hex data]

16773077

16772045

[invalid hex data]

[invalid hex data]

13480843

[invalid hex data]

[invalid hex data]

[invalid hex data]

10249759

16750866

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

9143160

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

16770244

15652279

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

13460992

[invalid hex data]

[invalid hex data]

[invalid hex data]

15637065

13468991

[invalid hex data]

16445670

[invalid hex data]

[invalid hex data]

13479829

[invalid hex data]

16774638

[invalid hex data]

13485503

[invalid hex data]

[invalid hex data]

13066516

13789470

[invalid hex data]

[invalid hex data]

13461021

9127187

[invalid hex data]

[invalid hex data]

16736515

9057807

[invalid hex data]

[invalid hex data]

15628610

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

15613952

[invalid hex data]

[invalid hex data]

6170130

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

13458245

[invalid hex data]

9057060

16737095

15621186

[invalid hex data]

[invalid hex data]

16416882

16770273

[invalid hex data]

13481909

[invalid hex data]

16775930

15657449

[invalid hex data]

9144713

12357519

[invalid hex data]

[invalid hex data]

13474715

[invalid hex data]

[invalid hex data]

13458524

16738922

[invalid hex data]

[invalid hex data]

13456725

[invalid hex data]

[invalid hex data]

15612731

13447987

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

9116186

16711680

[invalid hex data]

[invalid hex data]

9109504

8388608

9320590

[invalid hex data]

8232640

3706510

[invalid hex data]

9342520

[invalid hex data]

[invalid hex data]

[invalid hex data]

505290270

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

9342606

[invalid hex data]

[invalid hex data]

[invalid hex data]

15395562

[invalid hex data]

[invalid hex data]

16119285

[invalid hex data]

[invalid hex data]

[invalid hex data]

11119017

[invalid hex data]

[invalid hex data]

0

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

15461355

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

14606046

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

13224393

[invalid hex data]

[invalid hex data]

[invalid hex data]

12566463

[invalid hex data]

12237498

[invalid hex data]

11908533

11776947

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

10592673

[invalid hex data]

10263708

[invalid hex data]

[invalid hex data]

[invalid hex data]

9539985

9408399

[invalid hex data]

9079434

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

7895160

[invalid hex data]

[invalid hex data]

[invalid hex data]

7237230

[invalid hex data]

[invalid hex data]

6710886

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

5723991

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

4868682

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

3552822

[invalid hex data]

[invalid hex data]

[invalid hex data]

2829099

2697513

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

[invalid hex data]

1315860

[invalid hex data]

[invalid hex data]

855309

[invalid hex data]

526344

[invalid hex data]

[invalid hex data]

### Standard Output (Abridged)

Converting hexadecimal data in "hexdata.txt" to decimal and writing to "hexoutput.txt".

Line 001: 0x000001 read as 000001 and converted to (written as) 1

Line 002: 0x000002 read as 000002 and converted to (written as) 2

Line 003: 0x000003 read as 000003 and converted to (written as) 3

Line 004: 0x000004 read as 000004 and converted to (written as) 4

Line 005: 0x000005 read as 000005 and converted to (written as) 5

...

Line 252: 0x0000FC read as 0000FC and converted to (written as) 252

Line 253: 0x0000FD read as 0000FD and converted to (written as) 253

Line 254: 0x0000FE read as 0000FE and converted to (written as) 254

Line 255: 0x0000FF read as 0000FF and converted to (written as) 255

Line 256: read as and converted to (written as) [invalid hex data]

Line 257: 0x00FFFF read as 00FFFF and converted to (written as) 65535

Line 258: 0x010000 read as 010000 and converted to (written as) 65536

Line 259: 0x010001 read as 010001 and converted to (written as) 65537

Line 260: read as and converted to (written as) [invalid hex data]

Line 261: 0x7FFFFFFF read as 7FFFFFFF and converted to (written as) 2147483647

Line 262: 0x80000000 read as 80000000 and converted to (written as) [invalid hex data]

Line 263: 0x80000001 read as 80000001 and converted to (written as) [invalid hex data]

Line 264: read as and converted to (written as) [invalid hex data]

Line 265: 0xFFFFFFFF read as FFFFFFFF and converted to (written as) [invalid hex data]

Line 266: 0x100000000 read as 100000000 and converted to (written as) 0

Line 267: 0x100000001 read as 100000001 and converted to (written as) 1

Line 268: read as and converted to (written as) [invalid hex data]

Line 269: #0001UR read as 0001UR and converted to (written as) [invalid hex data]

Line 270: FFFRC6C1 read as FFFRC6C1 and converted to (written as) [invalid hex data]

Line 271: OXFSARA9 read as OXFSARA9 and converted to (written as) [invalid hex data]

Line 272: 0XEEA2AD read as EEA2AD and converted to (written as) 15639213

Line 273: #GDMC9Q read as GDMC9Q and converted to (written as) [invalid hex data]

...

Line 817: 0x0D0D0D read as 0D0D0D and converted to (written as) 855309

Line 818: 0X0A0OVW read as 0A0OVW and converted to (written as) [invalid hex data]

Line 819: 080808 read as 080808 and converted to (written as) 526344

Line 820: 0XV5050C read as V5050C and converted to (written as) [invalid hex data]

Line 821: 0303030U read as 0303030U and converted to (written as) [invalid hex data]

428 of 821 values (lines) translated from hexadecimal to decimal integers. (393 values were invalid.)

Fibonacci Sequence Conversion

n : [input num.] -> 0x [hex] -> [output num.]

00 : 0 -> 0x00000000 -> 0

01 : 1 -> 0x00000001 -> 1

02 : 1 -> 0x00000001 -> 1

03 : 2 -> 0x00000002 -> 2

04 : 3 -> 0x00000003 -> 3

05 : 5 -> 0x00000005 -> 5

06 : 8 -> 0x00000008 -> 8

07 : 13 -> 0x0000000d -> 13

08 : 21 -> 0x00000015 -> 21

09 : 34 -> 0x00000022 -> 34

10 : 55 -> 0x00000037 -> 55

11 : 89 -> 0x00000059 -> 89

12 : 144 -> 0x00000090 -> 144

13 : 233 -> 0x000000e9 -> 233

14 : 377 -> 0x00000179 -> 377

15 : 610 -> 0x00000262 -> 610

16 : 987 -> 0x000003db -> 987

17 : 1,597 -> 0x0000063d -> 1,597

18 : 2,584 -> 0x00000a18 -> 2,584

19 : 4,181 -> 0x00001055 -> 4,181

20 : 6,765 -> 0x00001a6d -> 6,765

21 : 10,946 -> 0x00002ac2 -> 10,946

22 : 17,711 -> 0x0000452f -> 17,711

23 : 28,657 -> 0x00006ff1 -> 28,657

24 : 46,368 -> 0x0000b520 -> 46,368

25 : 75,025 -> 0x00012511 -> 75,025

26 : 121,393 -> 0x0001da31 -> 121,393

27 : 196,418 -> 0x0002ff42 -> 196,418

28 : 317,811 -> 0x0004d973 -> 317,811

29 : 514,229 -> 0x0007d8b5 -> 514,229

30 : 832,040 -> 0x000cb228 -> 832,040

31 : 1,346,269 -> 0x00148add -> 1,346,269

32 : 2,178,309 -> 0x00213d05 -> 2,178,309

33 : 3,524,578 -> 0x0035c7e2 -> 3,524,578

34 : 5,702,887 -> 0x005704e7 -> 5,702,887

35 : 9,227,465 -> 0x008cccc9 -> 9,227,465

36 : 14,930,352 -> 0x00e3d1b0 -> 14,930,352

37 : 24,157,817 -> 0x01709e79 -> 24,157,817

38 : 39,088,169 -> 0x02547029 -> 39,088,169

39 : 63,245,986 -> 0x03c50ea2 -> 63,245,986

40 : 102,334,155 -> 0x06197ecb -> 102,334,155

41 : 165,580,141 -> 0x09de8d6d -> 165,580,141

42 : 267,914,296 -> 0x0ff80c38 -> 267,914,296

43 : 433,494,437 -> 0x19d699a5 -> 433,494,437

44 : 701,408,733 -> 0x29cea5dd -> 701,408,733

45 : 1,134,903,170 -> 0x43a53f82 -> 1,134,903,170

46 : 1,836,311,903 -> 0x6d73e55f -> 1,836,311,903

User Input Conversion

Hexadecimal input: 0xaa

Decimal conversion: 170

### More User Input Conversion Variations (Attempts to Break the Program)

#### 100 zeroes followed by a 1

Hexadecimal input: 0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001

Decimal conversion: 1

#### Non-standard characters and those treated differently by the Java standard input stream

Note that typed input is after the “0x”; i.e. “%n” in the first example.

Hexadecimal input: 0x%n

Decimal conversion: [invalid hex data]

Hexadecimal input: 0x\n

Decimal conversion: [invalid hex data]

Other inputs attempted with the same result: “'\n'”, “'%n'”, “ 1”, “g”, “0g”, “1g1”, “g1”, “0x1”, “1 “, “ f”, “f “, “f f”, ”ff “, “\t”, “’\t’”, “#ffffff”, and “0xffffff”.

#### Large value inputs, near and over the maximum value for Java int

##### 2^31-1 (Integer.MAX\_VALUE)

Hexadecimal input: 0x7FFFFFFF

Decimal conversion: 2,147,483,647

##### 2^31

Hexadecimal input: 0x80000000

Decimal conversion: [invalid hex data]

##### 2^31+1

Hexadecimal input: 0x80000001

Decimal conversion: [invalid hex data]

##### 2^32-1

Hexadecimal input: 0xFFFFFFFF

Decimal conversion: [invalid hex data]

##### 2^32

Hexadecimal input: 0x100000000

Decimal conversion: 0

##### 2^32+1

Hexadecimal input: 0x100000001

Decimal conversion: 1

##### 2^32-1+255

Hexadecimal input: 0x1000000FF

Decimal conversion: 255

##### 2^32 + Integer.MAX\_VALUE

Hexadecimal input: 0x17FFFFFFF

Decimal conversion: 2,147,483,647

##### 2^32 + Integer.MAX\_VALUE + 1

Hexadecimal input: 0x180000000

Decimal conversion: [invalid hex data]

##### 2^33-1

Hexadecimal input: 0x1FFFFFFFF

Decimal conversion: [invalid hex data]

##### 2^33

Hexadecimal input: 0x200000000

Decimal conversion: 0

##### 2^33+1

Hexadecimal input: 0x200000001

Decimal conversion: 1

##### 2^33+255

Hexadecimal input: 0x2000000FF

Decimal conversion: 255

##### 2^33 + Integer.MAX\_VALUE

Hexadecimal input: 0x27FFFFFFF

Decimal conversion: 2,147,483,647

##### 2^33 + Integer.MAX\_VALUE + 1

Hexadecimal input: 0x280000000

Decimal conversion: [invalid hex data]

# Discussion

Values that are multiples of 232 but have a modulus less than Integer.MAX\_VALUE (231-1) fooled the algorithm into thinking the input was less than Integer.MAX\_VALUE. I had attempted to avoid this by implementing the methods addIntsNoOVF(int,int) and multiplyIntsNoOVF(int,int), which check to make sure that the int data type does not overflow attempting to store the result of adding or multiplying. They didn’t work! They were implemented by storing the result in a larger data type (long) and checking to see if it was larger than Integer.MAX\_VALUE. The problem with that is that even long will overflow. Going to a larger-yet data type runs into the same issue. I replaced my overflow-checking methods with the standard methods Math.multiplyExact(int,int) and Math.addExact(int,int). Their documentation summary states:

The platform uses signed two's complement integer arithmetic with int and long primitive types. The developer should choose the primitive type to ensure that arithmetic operations consistently produce correct results, which in some cases means the operations will not overflow the range of values of the computation. The best practice is to choose the primitive type and algorithm to avoid overflow. In cases where the size is int or long and overflow errors need to be detected, the methods addExact, subtractExact, multiplyExact, and toIntExact throw an ArithmeticException when the results overflow. For other arithmetic operations such as divide, absolute value, increment, decrement, and negation overflow occurs only with a specific minimum or maximum value and should be checked against the minimum or maximum as appropriate. (Oracle, 2017)[[1]](#footnote-1)

I had also missed checking for an int multiplication overflow on one of the terms. It works, now. For each but the first of the following inputs, the output was “[invalid hex data]” (as they should have been under Large value inputs, near and over the maximum value for Java int): 7FFFFFFF, 80000000, 80000001, FFFFFFFF, 100000000, 100000001, 1000000FF, 17FFFFFFF, 180000000, 1FFFFFFFF, 200000000, 200000001, 2000000FF, 27FFFFFFF, 280000000.

1. Oracle. (2017, 05 22). *Math (Java Platform SE 8 )*. Retrieved from Oracle Help Center: https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html [↑](#footnote-ref-1)