**Parameshor Bhandari**

**Lab 5**

**COSC 3319 Data Structure**

**Dr. Burris**

**A Option**

In C option I am using array to store all data where as in B option I am using relative random access file to store the value. That is the only difference between Option B and C. This a summary of this lab where I am trying to compare the minimum, maximum and average numbers of probes that it takes to search an item when the table is 50% full and 85% full with the help of Linear and Random probe technique.

The runtime according to the notes are as follows:

1. Linear probe:
2. Random Probe: Where **a** is the load factor and **ln** is the natural log.
3. Linear Probe Collision Handling
   1. Hash Table 50% full (128/256)
      * Theoretical expected probes:  **= 1.5**
   2. Hash Table 85% full (217.6/256)
      * Theoretical expected probes:  **3.83**
4. Random Probe Collision Handling
   1. Hash Table 50% full (128/256)
      * Theoretical expected probes:  **1.38629**
   2. Hash Table 85% full (217.6/256)
      * Theoretical expected probes:

The hash algorithm derived doesn’t perform the computation close to the theoretical hash algorithm. There is always margin of error involved. The hash function has some weaknesses that affect the collision handler's performance as stated in the lecture note. According to the data set used, most of the character strings of length 16 end with spaces. The hashing algorithm provided uses the rear end of the string, meaning many of the words will have the same hash value, and thus multiple collisions.One possible quick fix to the current hashing algorithm is to use the front end of the string rather than the rear, that is, slices 1..2 and 4..5. This will get letters rather than the ending spaces for most of the data set.My conclusion is that more complicated and complex hashing function will yield less collision. The computations from my attempts on the hash functions are mentioned in the following hash table.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Original Hash Function:*** | |  |  |  | |  | |
| Hash:= (ABS(StringToInteger(fold(Key,13,14))) + ABS(StringToInteger(Fold(Key,15,16)))) mod Table\_Size;; | | | | | | |  | |
| ***Look Ups*** | ***Probes*** | ***Linear(50% Full)*** | ***Linear(85% Full)*** | | ***Random(50% Full)*** | | ***Random(85% Full)*** | |
| First 30 | Min | 1 | 1 | | 1 | | 1 | |
|  | Max | 22 | 119 | | 202 | | 114 | |
|  | Average | 9.63E+00 | 20.2 | | 98.6 | | 54.3 | |
| Last 30 | Min | 112 | 172 | | 84 | | 177 | |
|  | Max | 119 | 202 | | 114 | | 207 | |
|  | Average | 115.5 | 187.3 | | 98.6 | | 191.6 | |
|  |  |  |  | |  | |  | |
| ***My Hash:*** | Hash:= (StringToInteger(Fold(Key,1,2)) - StringToInteger(fold(Key,3,4)) + StringToInteger(fold(Key,1,3))) ;  Hash:= (ABS(Hash \* 13 + (StringToInteger(fold(Key,2,3)) - StringToInteger(fold(Key,1,4))) mod 61) mod Table\_Size); | | | | | | | |
|  | |  |  | |  | |  | |
| ***Look Ups*** | ***Probes*** | ***Linear(50% Full)*** | ***Linear(85% Full)*** | | ***Random(50% Full)*** | | ***Random(85% Full)*** | |
| First 30 | Min | 1 | 1 | | 1 | | 1 | |
|  | Max | 1 | 1 | | 1 | | 1 | |
|  | Average | 1 | 1 | | 1 | | 1 | |
| Last 30 | Min | 1 | 1 | | 1 | | 1 | |
|  | Max | 2 | 11 | | 2 | | 10 | |
|  | Average | 1 | 2.5 | | 1 | | 2.9 | |
|  |  |  |  | |  | |  | |

according to table , the hash function provided takes more probes on average where as in the my hash it takes only 1 probe on average to find an item, which clearly shows that my hash function is way better than given hash table.

The reason why the hash function given in the lab is that bad is because couple of reasons. First of all, it uses the folding technique by adding the 13 and 14 bytes with the 15 and 16 bytes of the word. This is terrible because all the data is left justified 16 byte words, making the hash function create the address with the same values, therefore, generating a lot of collisions. Consider the data value “Lisa ”, the bytes 13, 14, 15, and 16 are all empty spaces, now consider the next data value “Judy ”, the bytes 13, 14, 15, and 16 are also all empty spaces. Both of these values will generate the same hash address making a collision. Taking in count that the last 4 bytes of almost all the data are empty spaces, it makes sense that there will be a lot of collision. Secondly, once you get the sum of the last bytes of the word the given hash function does nothing else with the sum, leaving a short range of possible values.

On an empirical stand point of view, the hash function that was given in the lab had a lot of primary clustering. Referring to the 50% full and 85% full linear tables using the given hash function, you can see how the data is back to back from the first available spot until the last spot with only one part of the table empty. It is very easy to see how the primary cluster is at the top and at the bottom of each table. This happens because the hash address of almost all the values is the same. In consequence, using the linear technique you add one to the hash address until you find an empty spot, causing the values to have more collisions each time a value is searched or inserted. Consider the value “Lisa” with a hash address of 15 and 1 probe if you are look for it, then if you add the value “Judy”, it generates the same hash address of 15 but this time it will take 2 probes to find this value. Then insert the value “school” also with a hash address of 15, this time it will take 3 probes. When the table has 200

Items would not be a surprise for the maximum number of probes to be equal or close to the number of items, in this case 200.

Comparing this table with my hash we can clearly conclude that in hash the clustering is lot less and data is better distributed, and from empirical point of view it would take lot less probe to find any item from the list since there is less clustering and collision. The reason is in my hash I am folding value 2 3 byte due to which there is not that much collision as compare to given hash where folding is done at 14 , 16 byte . That concludes my reasoning.

Option A :

Choice A) Do both “C” and “B” option.

------Original Hash-------------------

Statistics for Part A (Linear Probe at 50% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 22

Avg probes: 9.63333E+00

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 112

Max probes: 119

Avg probes: 1.15500E+02

Table for Part A (Linear Probe at 50% full)

1 Perfect

2 mole

3 poison

4 brutalize

5 cap

6 cauldron

7 prepossess

8 wince

9 orthodontist

10 live

11 magnetic

12 inlet

13 constrain

14 marsupial

15 rationalize

16 scat

17 toluene

18 wet

19 sparse

20 quandary

21 dactyl

22 nosegay

23 option

24 forgetful

25 privilege

26 sponsor

27 exhilarate

28 guard

29 noggin

30 prologue

31 seal

32 seat

33 tiller

34 ichthyosaur

35 lazy

36 malfeasance

37 compass

38 diastase

39 emperor

40 history

41 keep

42 parasympathetic

43 gangway

44 labial

45 sacrament

46 taint

47 withal

48 oak

49 preordain

50 haberdasher

51 crimp

52 luculent

53 pennywort

54 prevalent

55 monolith

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 ratiocination

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 1234567890123456

154 Aguirrie90123456

155 Bhandari90123456

156 Casperri90123456

157 Danielsi90123456

158 Paschali90123456

159 Salkowski0123456

160 Qamruddin0123456

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 Ajose

194 Arauza

195 Buck

196 Clark

197 Crouch

198 Davies

199 Dugger

200 Egbe

201 Ellington

202 Farral

203 Garza

204 Gurung

205 Joseph

206 Kelly

207 Corey

208 Adam

209 Clayton

210 Dustin

211 Robert

212 Kyle

213 Scott

214 Octavio

215 Judy

216 Derek

217 Jeffrey

218 Jordon

219 Vinnela

220 Lisa

221 Todd

222 Veronica

223 Matthew

224 Michael

225 Akhila

226 John

227 Charles

228 James

229 Chris

230 Wade

231 Christopher

232 Fernando

233 Batbold

234 Joel

235 Fabulous

236 Misogamist

237 Maiden

238 Eye

239 Constriction

240 Necromancer

241 Syncopate

242 Yolk

243 Afterwards

244 Person

245 Northwest

246 Irreversible

247 Fabricate

248 Honor

249 Staple

250 Under

251 Jutty

252 Finagle

253 Cook

254 Rush

255 Wine

256 Screen

Statistics for Part B (Linear Probe at 85% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 119

Avg probes: 2.02167E+01

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 172

Max probes: 202

Avg probes: 1.87267E+02

Table for Part B (Linear Probe at 85% full)

1 Perfect

2 mole

3 poison

4 brutalize

5 cap

6 cauldron

7 prepossess

8 wince

9 orthodontist

10 live

11 magnetic

12 inlet

13 constrain

14 marsupial

15 rationalize

16 scat

17 toluene

18 wet

19 sparse

20 quandary

21 dactyl

22 nosegay

23 option

24 forgetful

25 privilege

26 sponsor

27 exhilarate

28 guard

29 noggin

30 prologue

31 seal

32 seat

33 tiller

34 ichthyosaur

35 lazy

36 malfeasance

37 compass

38 diastase

39 emperor

40 history

41 keep

42 parasympathetic

43 gangway

44 labial

45 sacrament

46 taint

47 withal

48 oak

49 preordain

50 haberdasher

51 crimp

52 luculent

53 pennywort

54 prevalent

55 monolith

56 suffrage

57 wiper

58 zygote

59 academy

60 amputate

61 brash

62 consummation

63 epicure

64 indulgent

65 minute

66 job

67 hypodermic

68 meridian

69 sinister

70 tag

71 uterus

72 perennial

73 humidity

74 gynecology

75 forego

76 interchangeable

77 needle

78 reach

79 urban

80 nonunion

81 jingle

82 feculent

83 deciduous

84 channel

85 decimate

86 alarm

87 perimeter

88 unlucky

89 wonderful

90 script

91 mousse

92 issue

93 graduation

94 neutrality

95 proxy

96 swine

97 witticism

98 parish

99 sickly

100 way

101 periodic

102 mail

103 close

104 particularity

105 bash

106 conclusion

107 drive

108 foreclose

109 habituation

110 liberty

111 pall

112 recite

113 empress

114 impetuous

115 ratiocination

116 ruminate

117 fork

118 horizon

119 prerequisite

120 romance

121 sensual

122 story

123 tuck

124 imperialist

125 great

126 excite

127 east

128 conclave

129 beast

130 calico

131 beggar

132 article

133 enemy

134 chuck

135 critical

136 ghost

137 dispense

138 figure

139 algorithm

140 bile

141 handsome

142 cat

143 ivy

144 impersonal

145 pennant

146 elephant

147

148

149

150

151

152

153 1234567890123456

154 Aguirrie90123456

155 Bhandari90123456

156 Casperri90123456

157 Danielsi90123456

158 Paschali90123456

159 Salkowski0123456

160 Qamruddin0123456

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 Ajose

194 Arauza

195 Buck

196 Clark

197 Crouch

198 Davies

199 Dugger

200 Egbe

201 Ellington

202 Farral

203 Garza

204 Gurung

205 Joseph

206 Kelly

207 Corey

208 Adam

209 Clayton

210 Dustin

211 Robert

212 Kyle

213 Scott

214 Octavio

215 Judy

216 Derek

217 Jeffrey

218 Jordon

219 Vinnela

220 Lisa

221 Todd

222 Veronica

223 Matthew

224 Michael

225 Akhila

226 John

227 Charles

228 James

229 Chris

230 Wade

231 Christopher

232 Fernando

233 Batbold

234 Joel

235 Fabulous

236 Misogamist

237 Maiden

238 Eye

239 Constriction

240 Necromancer

241 Syncopate

242 Yolk

243 Afterwards

244 Person

245 Northwest

246 Irreversible

247 Fabricate

248 Honor

249 Staple

250 Under

251 Jutty

252 Finagle

253 Cook

254 Rush

255 Wine

256 Screen

Hashing collision handler set for random probe. (Part C)

Statistics for Part C (Random Probe at 50% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 202

Avg probes: 9.86167E+01

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 84

Max probes: 114

Avg probes: 9.86333E+01

Table for Part C (Random Probe at 50% full)

1

2

3 Davies

4

5 privilege

6 Fabulous

7

8 Staple

9 Irreversible

10

11

12 Dugger

13 toluene

14

15

16 Dustin

17 Qamruddin0123456

18

19 crimp

20 guard

21 live

22 sponsor

23 Necromancer

24

25 Christopher

26 Ellington

27 Misogamist

28

29 mole

30

31 emperor

32

33

34 Michael

35

36 Kelly

37 Under

38

39

40

41 Judy

42 Fabricate

43 parasympathetic

44 malfeasance

45 Yolk

46 James

47

48 oak

49

50 marsupial

51 sparse

52

53 Casperri90123456

54

55 Clayton

56 orthodontist

57 Egbe

58 diastase

59 Joseph

60 Octavio

61 Charles

62 wet

63 wince

64 John

65

66

67

68 labial

69 Vinnela

70

71

72

73

74

75

76

77 Robert

78

79

80

81 sacrament

82

83 Person

84 Veronica

85 Gurung

86 Lisa

87

88

89

90

91

92 luculent

93 Clark

94

95 Batbold

96

97 noggin

98

99

100

101

102 magnetic

103 seal

104 forgetful

105

106

107 exhilarate

108 Wade

109

110

111

112 Syncopate

113 constrain

114 Adam

115 Scott

116 ratiocination

117

118

119

120 Garza

121

122 Fernando

123

124

125

126 Kyle

127 Farral

128

129 Rush

130 Wine

131

132 Maiden

133

134

135 Screen

136

137

138 nosegay

139 Finagle

140

141

142 poison

143 monolith

144

145 Eye

146 taint

147

148

149

150

151

152 history

153

154 1234567890123456

155

156 Northwest

157

158 haberdasher

159 Aguirrie90123456

160 Perfect

161 Matthew

162

163 lazy

164

165

166 Danielsi90123456

167 Akhila

168 Jordon

169

170

171 Todd

172

173

174

175 option

176

177 Corey

178

179

180 Cook

181

182 Jutty

183

184 Bhandari90123456

185

186

187

188

189

190

191

192

193

194 Ajose

195 brutalize

196

197

198 gangway

199 Arauza

200

201 pennywort

202 Derek

203

204 cap

205

206 Crouch

207 Honor

208 scat

209

210 Constriction

211

212

213

214

215 withal

216 Joel

217 compass

218 prepossess

219 Paschali90123456

220

221

222 Afterwards

223

224 Buck

225

226 prologue

227 Chris

228 Salkowski0123456

229

230

231

232

233 dactyl

234 prevalent

235

236

237 preordain

238 ichthyosaur

239 Jeffrey

240

241

242

243

244

245 keep

246

247 rationalize

248

249 cauldron

250

251 inlet

252 quandary

253 tiller

254

255

256 seat

Statistics for Part C (Random Probe at 85% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 114

Avg probes: 5.43000E+01

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 177

Max probes: 207

Avg probes: 1.91633E+02

Table for Part C (Random Probe at 85% full)

1 imperialist

2 great

3 Davies

4 drive

5 privilege

6 Fabulous

7 excite

8 Staple

9 Irreversible

10

11 story

12 Dugger

13 toluene

14 beast

15

16 Dustin

17 Qamruddin0123456

18

19 crimp

20 guard

21 live

22 sponsor

23 Necromancer

24

25 Christopher

26 Ellington

27 Misogamist

28 impetuous

29 mole

30

31 emperor

32 east

33 mousse

34 Michael

35

36 Kelly

37 Under

38 perennial

39 graduation

40 perimeter

41 Judy

42 Fabricate

43 parasympathetic

44 malfeasance

45 Yolk

46 James

47

48 oak

49 forego

50 marsupial

51 sparse

52 tuck

53 Casperri90123456

54 sensual

55 Clayton

56 orthodontist

57 Egbe

58 diastase

59 Joseph

60 Octavio

61 Charles

62 wet

63 wince

64 John

65 amputate

66 brash

67 calico

68 labial

69 Vinnela

70

71 consummation

72

73

74 decimate

75 academy

76 beggar

77 Robert

78 minute

79 horizon

80 cat

81 sacrament

82 foreclose

83 Person

84 Veronica

85 Gurung

86 Lisa

87

88 mail

89 interchangeable

90 enemy

91

92 luculent

93 Clark

94 recite

95 Batbold

96 epicure

97 noggin

98

99 witticism

100 figure

101

102 magnetic

103 seal

104 forgetful

105 particularity

106

107 exhilarate

108 Wade

109

110

111 alarm

112 Syncopate

113 constrain

114 Adam

115 Scott

116 ratiocination

117

118 zygote

119 handsome

120 Garza

121 article

122 Fernando

123 dispense

124 elephant

125

126 Kyle

127 Farral

128

129 Rush

130 Wine

131 job

132 Maiden

133

134 bash

135 Screen

136 prerequisite

137 ruminate

138 nosegay

139 Finagle

140 hypodermic

141 ivy

142 poison

143 monolith

144 urban

145 Eye

146 taint

147

148

149 critical

150

151 habituation

152 history

153 sickly

154 1234567890123456

155 conclusion

156 Northwest

157 conclave

158 haberdasher

159 Aguirrie90123456

160 Perfect

161 Matthew

162 issue

163 lazy

164 gynecology

165 romance

166 Danielsi90123456

167 Akhila

168 Jordon

169 channel

170 fork

171 Todd

172

173 pall

174 swine

175 option

176

177 Corey

178 bile

179 pennant

180 Cook

181 close

182 Jutty

183 reach

184 Bhandari90123456

185 meridian

186

187 humidity

188 deciduous

189 proxy

190 impersonal

191 chuck

192 neutrality

193

194 Ajose

195 brutalize

196

197 unlucky

198 gangway

199 Arauza

200 suffrage

201 pennywort

202 Derek

203

204 cap

205 nonunion

206 Crouch

207 Honor

208 scat

209

210 Constriction

211 empress

212 script

213 uterus

214 wonderful

215 withal

216 Joel

217 compass

218 prepossess

219 Paschali90123456

220

221 indulgent

222 Afterwards

223 periodic

224 Buck

225

226 prologue

227 Chris

228 Salkowski0123456

229 wiper

230 ghost

231

232

233 dactyl

234 prevalent

235

236 parish

237 preordain

238 ichthyosaur

239 Jeffrey

240 liberty

241 algorithm

242 needle

243 feculent

244

245 keep

246

247 rationalize

248 tag

249 cauldron

250 way

251 inlet

252 quandary

253 tiller

254 jingle

255 sinister

256 seat

---------------My Hash-----------------------

Statistics for Part A (Linear Probe at 50% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 1

Avg probes: 1.00000E+00

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 1

Max probes: 1

Avg probes: 1.00000E+00

Table for Part A (Linear Probe at 50% full)

1

2 Necromancer

3

4 Wade

5 Casperri90123456

6 Eye

7 pennywort

8

9

10 oak

11

12 Clark

13

14

15 noggin

16 Charles

17 magnetic

18 gangway

19

20

21 Davies

22 history

23 keep

24

25 Michael

26

27 ratiocination

28 Aguirrie90123456

29 rationalize

30 sparse

31 Arauza

32 Lisa

33 prologue

34 seal

35 Dugger

36

37 Scott

38

39

40

41 Afterwards

42

43 Screen

44 Honor

45 sacrament

46 Batbold

47 malfeasance

48 crimp

49 Jordon

50

51

52

53

54 Bhandari90123456

55 Jutty

56

57

58 lazy

59

60

61

62 Joseph

63 Qamruddin0123456

64 scat

65 seat

66 guard

67

68 Buck

69

70

71 Adam

72 wince

73

74

75

76

77

78

79

80

81 Yolk

82 mole

83 Irreversible

84 monolith

85 quandary

86

87 Rush

88 compass

89 haberdasher

90

91

92 Staple

93 forgetful

94 emperor

95

96

97 Garza

98

99 taint

100

101 parasympathetic

102 Under

103

104

105

106

107

108 1234567890123456

109 Ellington

110 Judy

111

112

113 Robert

114 Todd

115 Maiden

116 Finagle

117

118

119

120 cap

121

122

123

124 constrain

125 nosegay

126 ichthyosaur

127

128 Perfect

129

130

131 tiller

132

133

134 orthodontist

135

136

137

138

139 Gurung

140

141 prepossess

142

143

144 Jeffrey

145 Akhila

146 Joel

147 Danielsi90123456

148 Farral

149 marsupial

150 option

151

152 prevalent

153

154

155

156 Constriction

157 Fernando

158 Paschali90123456

159

160

161

162 Person

163

164

165 diastase

166

167

168

169

170

171 John

172 toluene

173

174

175

176

177

178 Crouch

179

180 Fabulous

181

182

183 Cook

184 cauldron

185 inlet

186

187 dactyl

188

189 Wine

190 Egbe

191 preordain

192

193 wet

194

195

196

197 Kelly

198

199 withal

200 Vinnela

201

202

203 Dustin

204 James

205 Fabricate

206

207

208

209

210

211

212 Ajose

213

214 Northwest

215 Kyle

216

217 privilege

218

219 Octavio

220 Veronica

221 Derek

222

223

224

225

226 brutalize

227

228

229

230 Clayton

231 Chris

232 Christopher

233 labial

234

235 luculent

236 Corey

237 sponsor

238

239

240

241

242

243

244

245 Salkowski0123456

246 poison

247 Misogamist

248 live

249 Syncopate

250

251

252

253

254

255 Matthew

256 exhilarate

Statistics for Part B (Linear Probe at 85% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 1

Avg probes: 1.00000E+00

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 1

Max probes: 11

Avg probes: 2.51000E+00

Table for Part B (Linear Probe at 85% full)

1 script

2 Necromancer

3 issue

4 Wade

5 Casperri90123456

6 Eye

7 pennywort

8 hypodermic

9 way

10 oak

11 drive

12 Clark

13 horizon

14 impersonal

15 noggin

16 Charles

17 magnetic

18 gangway

19 urban

20 pennant

21 Davies

22 history

23 keep

24

25 Michael

26

27 ratiocination

28 Aguirrie90123456

29 rationalize

30 sparse

31 Arauza

32 Lisa

33 prologue

34 seal

35 Dugger

36 gynecology

37 Scott

38 uterus

39 tuck

40 ghost

41 Afterwards

42

43 Screen

44 Honor

45 sacrament

46 Batbold

47 malfeasance

48 crimp

49 Jordon

50

51 job

52 proxy

53

54 Bhandari90123456

55 Jutty

56 mail

57 interchangeable

58 lazy

59

60

61 algorithm

62 Joseph

63 Qamruddin0123456

64 scat

65 seat

66 guard

67 indulgent

68 Buck

69 forego

70 reach

71 Adam

72 wince

73 nonunion

74 perimeter

75 parish

76 tag

77 sickly

78 wonderful

79 periodic

80 foreclose

81 Yolk

82 mole

83 Irreversible

84 monolith

85 quandary

86 perennial

87 Rush

88 compass

89 haberdasher

90 particularity

91 graduation

92 Staple

93 forgetful

94 emperor

95 habituation

96 recite

97 Garza

98 wiper

99 taint

100 minute

101 parasympathetic

102 Under

103 empress

104 fork

105 east

106 calico

107 beggar

108 1234567890123456

109 Ellington

110 Judy

111 critical

112 bile

113 Robert

114 Todd

115 Maiden

116 Finagle

117 conclusion

118 great

119 conclave

120 cap

121 ivy

122

123

124 constrain

125 nosegay

126 ichthyosaur

127 suffrage

128 Perfect

129 consummation

130 close

131 tiller

132

133

134 orthodontist

135 dispense

136

137

138

139 Gurung

140

141 prepossess

142

143

144 Jeffrey

145 Akhila

146 Joel

147 Danielsi90123456

148 Farral

149 marsupial

150 option

151 meridian

152 prevalent

153 feculent

154 unlucky

155

156 Constriction

157 Fernando

158 Paschali90123456

159

160 witticism

161

162 Person

163 swine

164

165 diastase

166 prerequisite

167 elephant

168

169

170

171 John

172 toluene

173 deciduous

174 decimate

175

176

177

178 Crouch

179

180 Fabulous

181

182

183 Cook

184 cauldron

185 inlet

186 sinister

187 dactyl

188 sensual

189 Wine

190 Egbe

191 preordain

192 zygote

193 wet

194 jingle

195 channel

196 neutrality

197 Kelly

198 mousse

199 withal

200 Vinnela

201 ruminate

202 amputate

203 Dustin

204 James

205 Fabricate

206 brash

207 excite

208 enemy

209 chuck

210 figure

211 handsome

212 Ajose

213 cat

214 Northwest

215 Kyle

216

217 privilege

218

219 Octavio

220 Veronica

221 Derek

222 bash

223 story

224 beast

225 alarm

226 brutalize

227 romance

228

229

230 Clayton

231 Chris

232 Christopher

233 labial

234

235 luculent

236 Corey

237 sponsor

238 liberty

239 pall

240 article

241

242

243 needle

244

245 Salkowski0123456

246 poison

247 Misogamist

248 live

249 Syncopate

250 academy

251 humidity

252 impetuous

253 imperialist

254 epicure

255 Matthew

256 exhilarate

Hashing collision handler set for random probe.

(Part C)

Statistics for Part C (Random Probe at 50% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 1

Avg probes: 1.00000E+00

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 1

Max probes: 1

Avg probes: 1.00000E+00

Table for Part C (Random Probe at 50% full)

1 exhilarate

2

3 Necromancer

4

5 Wade

6 Casperri90123456

7 Eye

8

9

10

11 oak

12

13 Clark

14

15

16 noggin

17 Charles

18

19

20 live

21

22 Davies

23 history

24 keep

25

26 Michael

27

28 ratiocination

29 Aguirrie90123456

30

31 prologue

32 Arauza

33 rationalize

34

35

36 Dugger

37 Lisa

38 Scott

39

40

41

42 Afterwards

43

44 Screen

45 Honor

46

47 Batbold

48 option

49

50 Jordon

51

52

53 malfeasance

54

55 Bhandari90123456

56

57 crimp

58 sparse

59 lazy

60 Jutty

61

62

63 Joseph

64 Qamruddin0123456

65 scat

66

67 guard

68 seat

69 Buck

70

71

72 Adam

73 wince

74

75 sacrament

76

77

78

79

80

81

82 Yolk

83 mole

84 Irreversible

85

86 quandary

87

88 Rush

89

90 haberdasher

91 magnetic

92

93 Staple

94 emperor

95

96

97

98 Garza

99

100 taint

101

102 parasympathetic

103 Under

104

105

106

107

108

109 1234567890123456

110 Ellington

111 Judy

112

113 monolith

114 Robert

115 Todd

116 Finagle

117

118 compass

119 Maiden

120

121 cap

122

123 forgetful

124

125 constrain

126 nosegay

127 ichthyosaur

128

129 Perfect

130 Christopher

131

132 tiller

133

134

135 orthodontist

136 gangway

137

138

139

140 Gurung

141

142 prepossess

143

144

145 Jeffrey

146 Akhila

147

148 Danielsi90123456

149 Farral

150 Joel

151

152

153 prevalent

154 marsupial

155

156

157 Constriction

158 Fernando

159 Paschali90123456

160

161 pennywort

162

163 Person

164

165

166 diastase

167

168

169

170

171

172 John

173 toluene

174

175

176

177

178

179 Crouch

180

181 Fabulous

182

183 seal

184 Cook

185

186

187

188 dactyl

189 cauldron

190 Wine

191 Egbe

192

193

194 wet

195

196 preordain

197

198 Kelly

199

200 withal

201 Vinnela

202

203

204 Dustin

205 James

206 Fabricate

207

208

209

210

211

212

213 Ajose

214 inlet

215 Northwest

216 Kyle

217

218 privilege

219

220 Octavio

221 Veronica

222 Derek

223

224

225

226

227 brutalize

228

229

230

231 Clayton

232

233 labial

234

235

236 Chris

237 Corey

238 sponsor

239

240

241 luculent

242

243

244

245

246 Salkowski0123456

247

248 Misogamist

249

250 Syncopate

251 poison

252

253

254

255

256 Matthew

Statistics for Part C (Random Probe at 85% full)

Statistics of first 30 lookups:

Min probes: 1

Max probes: 1

Avg probes: 1.00000E+00

Resetting statistics counters...

Statistics of last 30 lookups:

Min probes: 1

Max probes: 10

Avg probes: 2.90258E+00

Table for Part C (Random Probe at 85% full)

1 exhilarate

2

3 Necromancer

4 epicure

5 Wade

6 Casperri90123456

7 Eye

8 way

9 excite

10 hypodermic

11 oak

12 issue

13 Clark

14 ruminate

15 conclusion

16 noggin

17 Charles

18 drive

19 particularity

20 live

21

22 Davies

23 history

24 keep

25 humidity

26 Michael

27

28 ratiocination

29 Aguirrie90123456

30 gynecology

31 prologue

32 Arauza

33 rationalize

34

35 dispense

36 Dugger

37 Lisa

38 Scott

39 uterus

40

41

42 Afterwards

43 tuck

44 Screen

45 Honor

46

47 Batbold

48 option

49 sensual

50 Jordon

51

52 job

53 malfeasance

54 chuck

55 Bhandari90123456

56 mail

57 crimp

58 sparse

59 lazy

60 Jutty

61

62 algorithm

63 Joseph

64 Qamruddin0123456

65 scat

66

67 guard

68 seat

69 Buck

70 interchangeable

71 horizon

72 Adam

73 wince

74 forego

75 sacrament

76 parish

77 tag

78 calico

79 wonderful

80 sickly

81 foreclose

82 Yolk

83 mole

84 Irreversible

85 perennial

86 quandary

87 script

88 Rush

89 enemy

90 haberdasher

91 magnetic

92 recite

93 Staple

94 emperor

95 neutrality

96

97 reach

98 Garza

99 wiper

100 taint

101

102 parasympathetic

103 Under

104

105 minute

106 east

107 habituation

108 bile

109 1234567890123456

110 Ellington

111 Judy

112 cat

113 monolith

114 Robert

115 Todd

116 Finagle

117

118 compass

119 Maiden

120 story

121 cap

122

123 forgetful

124

125 constrain

126 nosegay

127 ichthyosaur

128 suffrage

129 Perfect

130 Christopher

131

132 tiller

133 critical

134 close

135 orthodontist

136 gangway

137 indulgent

138

139

140 Gurung

141 empress

142 prepossess

143

144 great

145 Jeffrey

146 Akhila

147 meridian

148 Danielsi90123456

149 Farral

150 Joel

151 feculent

152 ghost

153 prevalent

154 marsupial

155 consummation

156

157 Constriction

158 Fernando

159 Paschali90123456

160 perimeter

161 pennywort

162

163 Person

164 swine

165

166 diastase

167 elephant

168 fork

169 ivy

170 imperialist

171 prerequisite

172 John

173 toluene

174 deciduous

175

176 impetuous

177

178

179 Crouch

180

181 Fabulous

182 impersonal

183 seal

184 Cook

185

186 decimate

187

188 dactyl

189 cauldron

190 Wine

191 Egbe

192 zygote

193 channel

194 wet

195 jingle

196 preordain

197

198 Kelly

199 urban

200 withal

201 Vinnela

202

203 amputate

204 Dustin

205 James

206 Fabricate

207 brash

208 proxy

209

210

211

212 figure

213 Ajose

214 inlet

215 Northwest

216 Kyle

217 handsome

218 privilege

219 conclave

220 Octavio

221 Veronica

222 Derek

223 unlucky

224 nonunion

225 bash

226 alarm

227 brutalize

228 romance

229 mousse

230

231 Clayton

232 beast

233 labial

234 pennant

235 witticism

236 Chris

237 Corey

238 sponsor

239

240 graduation

241 luculent

242

243 liberty

244 needle

245 pall

246 Salkowski0123456

247

248 Misogamist

249 sinister

250 Syncopate

251 poison

252 beggar

253 article

254 periodic

255 academy

256 Matthew

--Parameshor Bhandari

--COSC3319

--Lab 5

--A Option

--04/10/2015

GENERIC

TYPE Item IS PRIVATE;

Table\_Size: Long\_Integer;

--Required for Printing the table

WITH PROCEDURE Put(X: Item);

--Required to retrieve the key from the object data.

WITH PROCEDURE Key(X: Item; Y: OUT String);

PACKAGE HashTable IS

PROCEDURE Initialize(Collisiontype: IN Integer; Nullitem: IN Item; Removeditem: IN Item);

PROCEDURE Insert(Key: IN String; Data: IN Item);

PROCEDURE Remove(Key: IN String; Data: OUT Item);

PROCEDURE Get(Key: IN String; Data: OUT Item);

PROCEDURE GetMinProbes(X: OUT Integer);

PROCEDURE GetMaxProbes(X: OUT Integer);

PROCEDURE GetAvgProbes(X: OUT Float);

PROCEDURE ResetStatistics;

PROCEDURE PrintTable;

PROCEDURE ResetTable;

PROCEDURE SetCollisionHandler(Collisiontype: IN Integer);

END HashTable;

--Parameshor Bhandari

--CS3319 Data Stucture

--Lab 5

--A Option

--Dr. Burris

WITH Ada.Text\_IO;

WITH Ada.Direct\_IO;

WITH Ada.Numerics.Generic\_Elementary\_Functions;

WITH Unchecked\_Conversion;

with Ada.Unchecked\_Conversion;

PACKAGE BODY HashTable IS

TYPE St IS NEW String(1..2);

PACKAGE Iio IS NEW Ada.Text\_IO.Integer\_IO(Integer); USE Iio;

PACKAGE Long\_IO IS NEW Ada.Text\_IO.Integer\_IO(Long\_Integer);

PACKAGE Math IS NEW Ada.Numerics.Generic\_Elementary\_Functions(Float);

FUNCTION StringToInteger IS NEW Unchecked\_Conversion(St, Long\_Integer);

FUNCTION Fold(Str: IN String; Num, Num2: Integer) RETURN St IS

Tmp: St;

BEGIN

Tmp(1) := Str(Num);

Tmp(2) := Str(Num2);

RETURN Tmp;

END Fold;

PACKAGE AHashFile IS NEW Ada.Direct\_IO(Item); USE AHashFile;

Init:Boolean:=False;

Hash\_File:AHashFile.File\_Type;

Collision\_Type:Integer:=0;

N:Integer;

Removed\_Type: Item;

Null\_Type: Item;

--Statistic vars

Minprobes:Integer:=Standard.Integer(Table\_Size);

Maxprobes:Integer:=1;

Avgprobes:Float:=1.0;

Lookups:Integer:=0;

Totalprobes:Integer:=0;

--Initialize (MUST be called prior to use)

-- In Integer CollisionType 0=Linear, !0=Random

-- In Item NullItem Used to prime the RF with empty spaces.

-- In Item RemovedItem Used to replace an item in the RF when it is removed.

PROCEDURE Initialize(Collisiontype: IN Integer; Nullitem: Item; Removeditem: IN Item) IS

BEGIN

Init:=True;

SetCollisionHandler(Collisiontype);

Removed\_Type:=Removeditem;

Null\_Type:=Nullitem;

ResetTable;

END Initialize;

PROCEDURE Insert(Key: String; Data: IN Item) IS

Hash:Long\_Integer;

Random:Integer:=1;

Offset:Integer;

TempItem:Item;

BEGIN

IF Init THEN

--Original Hash

Hash:= (ABS(StringToInteger(fold(Key,13,14))) + ABS(StringToInteger(fold(Key,15,16)))) mod Table\_Size;

--My hash

-- hash:= (StringToInteger(Fold(Key,1,2)) - StringToInteger(fold(Key,3,4)) + StringToInteger(fold(Key,1,3))) ;

-- Hash:= (ABS(Hash \* 13 + (StringToInteger(fold(Key,2,3)) - StringToInteger(fold(Key,1,4))) mod 61) mod Table\_Size);

IF Collision\_Type = 0 THEN

FOR I IN 0..Table\_Size-1 LOOP

Read(Hash\_File, TempItem, Count(((Hash+I) mod Table\_Size)+1));

IF TempItem = Null\_Type OR TempItem = Removed\_Type THEN

Write(Hash\_File, Data, To => Count(((Hash+I) mod Table\_Size)+1));

EXIT;

END IF;

END LOOP;

ELSE

WHILE 1=1 LOOP

Random:=Random \* 5;

Random:=Random mod 2\*\*(N+2);

IF Random = 1 THEN

EXIT;

END IF;

Offset:=Random / 4;

IF Long\_Integer(Offset) < Table\_Size THEN

Read(Hash\_File, TempItem, Count(((Hash+Long\_Integer(Offset)) mod Table\_Size)+1));

IF TempItem = Null\_Type OR TempItem = Removed\_Type THEN

Write(Hash\_File, Data, To => Count(((Hash+Long\_Integer(Offset)) mod Table\_Size)+1));

EXIT;

END IF;

END IF;

END LOOP;

END IF;

END IF;

END Insert;

PROCEDURE Remove(Key: IN String; Data: OUT Item) IS

Hash:Long\_Integer;

Random:Integer:=1;

Offset:Integer;

TempItem:Item;

Tempkey:String(1..16); --assumed key length of 16

BEGIN

IF Init THEN

--Original Hash

Hash:= (ABS(StringToInteger(fold(Key,13,14))) + ABS(StringToInteger(Fold(Key,15,16)))) mod Table\_Size;

--My Hash

-- hash:= (StringToInteger(Fold(Key,1,2)) - StringToInteger(fold(Key,3,4)) + StringToInteger(fold(Key,1,3))) ;

-- Hash:= (ABS(Hash \* 13 + (StringToInteger(fold(Key,2,3)) - StringToInteger(fold(Key,1,4))) mod 61) mod Table\_Size);

IF Collision\_Type = 0 THEN

FOR I IN 0..Table\_Size-1 LOOP

Read(Hash\_File, TempItem, Count(((Hash+I) mod Table\_Size)+1));

HashTable.Key(Tempitem,Tempkey);

IF Tempkey = Key THEN

Data:=TempItem;

Write(Hash\_File, Removed\_Type, To => Count(((Hash+I) mod Table\_Size)+1));

EXIT;

END IF;

END LOOP;

ELSE

WHILE 1=1 LOOP

Random:=Random \* 5;

Random:=Random mod 2\*\*(N+2);

IF Random = 1 THEN

EXIT;

END IF;

Offset:=Random / 4;

IF Long\_Integer(Offset) < Table\_Size THEN

Read(Hash\_File, TempItem, Count(((Hash+Long\_Integer(Offset)) mod Table\_Size)+1));

HashTable.Key(Tempitem,Tempkey);

IF Tempkey = Key THEN

Data:=TempItem;

Write(Hash\_File, Removed\_Type, To => Count(((Hash+Long\_Integer(Offset)) mod Table\_Size)+1));

EXIT;

END IF;

END IF;

END LOOP;

END IF;

END IF;

END Remove;

PROCEDURE Get(Key: IN String; Data: OUT Item) IS

Hash:Long\_Integer;

Random:Integer:=1;

Offset:Integer;

Probecnt:Integer:=0;

Found:Boolean;

TempItem:Item;

Tempkey:String(1..16); --assumed key length of 16

BEGIN

IF Init THEN

Lookups:=Lookups+1;

--Original Hash

Hash:= (ABS(StringToInteger(fold(Key,13,14))) + ABS(StringToInteger(fold(Key,15,16)))) mod Table\_Size;

--My Hash

-- hash:= (StringToInteger(Fold(Key,1,2)) - StringToInteger(fold(Key,3,4)) + StringToInteger(fold(Key,1,3))) ;

-- Hash:= (ABS(Hash \* 13 + (StringToInteger(fold(Key,2,3)) - StringToInteger(fold(Key,1,4))) mod 61) mod Table\_Size);

IF Collision\_Type = 0 THEN

FOR I IN 0..Table\_Size-1 LOOP

Probecnt:=Probecnt+1;

Read(Hash\_File, TempItem, Count(((Hash+I) mod Table\_Size)+1));

HashTable.Key(TempItem,Tempkey);

IF Tempkey = Key THEN

Data:=TempItem;

Found:=True;

EXIT;

END IF;

END LOOP;

ELSE

WHILE 1=1 LOOP

Random:=Random \* 5;

Random:=Random mod 2\*\*(N+2);

IF Random = 1 THEN

EXIT;

END IF;

Offset:=Random / 4;

IF Long\_Integer(Offset) < Table\_Size THEN

Probecnt:=Probecnt+1;

Read(Hash\_File, TempItem, Count(((Hash+Long\_Integer(Offset)) mod Table\_Size)+1));

HashTable.Key(Tempitem,Tempkey);

IF Tempkey = Key THEN

Data:=TempItem;

Found:=True;

EXIT;

END IF;

END IF;

END LOOP;

END IF;

--Statistics

IF Found THEN

Totalprobes:= Totalprobes + Probecnt;

Avgprobes:= Float(Totalprobes) / Float(Lookups);

IF Maxprobes < Probecnt THEN

Maxprobes:=Probecnt;

END IF;

IF Minprobes > Probecnt THEN

Minprobes:=Probecnt;

END IF;

END IF;

END IF;

END Get;

PROCEDURE GetMinProbes(X: OUT Integer) IS

BEGIN

X:=Minprobes;

END GetMinProbes;

PROCEDURE GetMaxProbes(X: OUT Integer) IS

BEGIN

X:=Maxprobes;

END GetMaxProbes;

PROCEDURE GetAvgProbes(X: OUT Float) IS

BEGIN

X:=Avgprobes;

END GetAvgProbes;

PROCEDURE GetTotalProbes(X: OUT Integer) IS

BEGIN

X:=Totalprobes;

END GetTotalProbes;

PROCEDURE ResetStatistics IS

BEGIN

Minprobes:=Standard.Integer(Table\_Size);

Maxprobes:=1;

Avgprobes:=1.0;

Lookups:=0;

Totalprobes:=0;

END ResetStatistics;

PROCEDURE PrintTable IS

Tempitem:Item;

BEGIN

IF Is\_Open(Hash\_File) THEN

FOR I IN 1..Table\_Size LOOP

Read(Hash\_File, Tempitem, Count(I));

Long\_IO.Put(I);Ada.Text\_IO.Put(" ");Put(Tempitem);Ada.Text\_IO.Put\_Line("");

END LOOP;

ELSE

Ada.Text\_IO.Put\_Line("Object needs to be initialized first.");

END IF;

END PrintTable;

PROCEDURE ResetTable IS

BEGIN

IF Is\_Open(Hash\_File) THEN

Close(Hash\_File);

END IF;

Create(Hash\_File, Mode => InOut\_File, Name => "hashtabledata.diorf" );

FOR I IN 1..Table\_Size LOOP

Write(Hash\_File, Null\_Type, To => Count(I));

END LOOP;

END ResetTable;

PROCEDURE SetCollisionHandler(Collisiontype: IN Integer) IS

Temp:Integer:=0;

BEGIN

Collision\_Type:=Collisiontype;

IF Collision\_Type /= 0 THEN

N:=Integer( Float'Ceiling(Math.Log( Float(Table\_Size) ) / Math.Log( 2.0 )));

END IF;

END SetCollisionHandler;

END HashTable;

--Parameshor Bhandari

--CS3319 Data Structure

--Lab 5

--A Option

--Dr. Burris

WITH Ada.Text\_IO; USE Ada.Text\_IO;

WITH HashTable;

PROCEDURE Program5 IS

PACKAGE Iio IS NEW Ada.Text\_IO.Integer\_IO(Integer); USE Iio;

PACKAGE Fio IS NEW Ada.Text\_IO.Float\_IO(Float); USE Fio;

TYPE String16 IS ARRAY(1..16) OF Character;

PROCEDURE String16\_Put(X: IN String16) IS

BEGIN

Ada.Text\_IO.Put(String(X));

END String16\_Put;

PROCEDURE String16\_Key(X: IN String16; Y: OUT String) IS

BEGIN

Y:=String(X);

END String16\_Key;

PACKAGE Hashtbl IS NEW HashTable(String16,256,String16\_Put,String16\_Key); USE Hashtbl;

Removed\_Item: CONSTANT String16:=" &&& ";

Null\_Item: CONSTANT String16:=" ";

WordsFile:File\_Type;

Tempkey:String(1..16);

Tempkeylen:Integer;

Min:Integer;

Max:Integer;

Avg:Float;

BEGIN

Hashtbl.Initialize(0,Null\_Item,Removed\_Item);

--50% of a table size of 256 is 128 items. insertting first 128 items from Words200D16.txt

Open(WordsFile,In\_File,"Words200D16.txt");

FOR I IN 1..128 LOOP

IF End\_Of\_File(WordsFile) THEN

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

insert(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

--Perform a lookup on the first 30 keys inserted into the table.

--Ada's file input requires me to close the file and reopen it in order to go back to line 1

Open(WordsFile,In\_File,"Words200D16.txt");

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

Put\_Line("Statistics for Part A (Linear Probe at 50% full)");

Put\_Line("Statistics of first 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Perform a lookup on the last 30 keys inserted into the table (128-30=98) (starting at line 98)

Put\_Line("Resetting statistics counters..."); resetStatistics;

Open(WordsFile,In\_File,"Words200D16.txt");

Set\_Line(WordsFile,128);

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

Put\_Line("Statistics of last 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Printing table for part A of grading option C.

Put\_Line("Table for Part A (Linear Probe at 50% full)");

printTable;

put\_line("");

--Clearing table for Part B of grading option C.

hashtbl.initialize(0,null\_item,removed\_item);

--85% of a table size of 256 is 217.6 items. insertting first 218 items from Words200D16.txt

Open(WordsFile,In\_File,"Words200D16.txt");

For i In 1..218 Loop

If End\_Of\_File(WordsFile) Then

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

insert(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

--Perform a lookup on the first 30 keys inserted into the table.

--Ada's file input requires me to close the file and reopen it in order to go back to line 1

Open(WordsFile,In\_File,"Words200D16.txt");

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

put\_line("Statistics for Part B (Linear Probe at 85% full)");

Put\_Line("Statistics of first 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Perform a lookup on the last 30 keys inserted into the table (218-30=188) (starting at line 188)

Put\_Line("Resetting statistics counters..."); resetStatistics;

Open(WordsFile,In\_File,"Words200D16.txt");

Set\_Line(WordsFile,188);

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

Put\_Line("Statistics of last 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Printing table for part B of grading option C.

Put\_Line("Table for Part B (Linear Probe at 85% full)");

printTable;

put\_line("");

--Switching to random probe collision handling

hashtbl.initialize(1,null\_item,removed\_item);

Put\_Line("Hashing collision handler set for random probe. (Part C)");

--50% of a table size of 256 is 128 items. insertting first 128 items from Words200D16.txt

Open(WordsFile,In\_File,"Words200D16.txt");

For i In 1..128 Loop

If End\_Of\_File(WordsFile) Then

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

insert(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

--Perform a lookup on the first 30 keys inserted into the table.

--Ada's file input requires me to close the file and reopen it in order to go back to line 1

Open(WordsFile,In\_File,"Words200D16.txt");

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

put\_line("Statistics for Part C (Random Probe at 50% full)");

Put\_Line("Statistics of first 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Perform a lookup on the last 30 keys inserted into the table (128-30=98) (starting at line 98)

Put\_Line("Resetting statistics counters..."); resetStatistics;

Open(WordsFile,In\_File,"Words200D16.txt");

Set\_Line(WordsFile,98);

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

Put\_Line("Statistics of last 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Printing table for part A of grading option C.

Put\_Line("Table for Part C (Random Probe at 50% full)");

printTable;

put\_line("");

--Clearing table for Part B of grading option C.

hashtbl.initialize(1,null\_item,removed\_item);

--85% of a table size of 265 is 217.6 items. insertting first 218 items from Words200D16.txt

Open(WordsFile,In\_File,"Words200D16.txt");

For i In 1..218 Loop

If End\_Of\_File(WordsFile) Then

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

insert(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

--Perform a lookup on the first 30 keys inserted into the table.

--Ada's file input requires me to close the file and reopen it in order to go back to line 1

Open(WordsFile,In\_File,"Words200D16.txt");

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Put\_Line("Words200D16.txt is shorter than expected. It may be corrupt.");

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

put\_line("Statistics for Part C (Random Probe at 85% full)");

Put\_Line("Statistics of first 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Perform a lookup on the last 30 keys inserted into the table (218-30=188) (starting at line 188)

Put\_Line("Resetting statistics counters..."); resetStatistics;

Open(WordsFile,In\_File,"Words200D16.txt");

Set\_Line(WordsFile,188);

For i In 1..30 Loop

If End\_Of\_File(WordsFile) Then

Exit;

End If;

Get\_line(WordsFile,tempkey,tempkeylen);

Skip\_Line(WordsFile);

hashtbl.get(tempkey,string16(tempkey));

End Loop;

Close(WordsFile);

Put\_Line("Statistics of last 30 lookups:");

getMinProbes(min); getMaxProbes(max); getAvgProbes(avg);

put("Min probes: ");put(min);put\_line("");

put("Max probes: ");put(max);put\_line("");

put("Avg probes: ");put(avg);put\_line("");

put\_line("");

--Printing table for part B of grading option C.

Put\_Line("Table for Part C (Random Probe at 85% full)");

printTable;

put\_line("");

End Program5;