1 KEYWORD const

2 KEYWORD int

3 IDENT a

4 EQUAL =

5 UNSIGNEDINTGER 1

6 SEMI ;

7 KEYWORD const

8 KEYWORD char

9 IDENT c

10 EQUAL =

11 CHAR 'c'

12 SEMI ;

13 KEYWORD const

14 KEYWORD int

15 IDENT f1

16 EQUAL =

17 UNSIGNEDINTGER 0

18 SEMI ;

19 KEYWORD const

20 KEYWORD int

21 IDENT f2

22 EQUAL =

23 UNSIGNEDINTGER 1

24 SEMI ;

25 KEYWORD int

26 IDENT b

27 COMMA ,

28 IDENT d

29 SEMI ;

30 KEYWORD char

31 IDENT c1

32 EQUAL =

33 CHAR 'a'

34 SEMI ;

35 KEYWORD char

36 IDENT c2

37 COMMA ,

38 IDENT c3

39 SEMI ;

40 KEYWORD int

41 IDENT num1

42 PAREN [

43 UNSIGNEDINTGER 100

44 PAREN ]

45 EQUAL =

46 BRACE {

47 IDENT a

48 BRACE }

49 SEMI ;

50 KEYWORD int

51 IDENT add

52 paren (

53 KEYWORD int

54 IDENT a

55 COMMA ,

56 KEYWORD int

57 IDENT b

58 paren )

59 BRACE {

60 KEYWORD return

61 IDENT a

62 ADDOP +

63 IDENT b

64 SEMI ;

65 BRACE }

66 KEYWORD void

67 IDENT ff

68 paren (

69 paren )

70 BRACE {

71 BRACE }

72 KEYWORD void

73 KEYWORD main

74 paren (

75 paren )

76 BRACE {

77 KEYWORD int

78 IDENT e

79 SEMI ;

80 KEYWORD int

81 IDENT num

82 SEMI ;

83 KEYWORD scanf

84 paren (

85 IDENT b

86 COMMA ,

87 IDENT d

88 paren )

89 SEMI ;

90 IDENT e

91 EQUAL =

92 IDENT a

93 ADDOP +

94 IDENT b

95 SEMI ;

96 IDENT e

97 EQUAL =

98 IDENT a

99 ADDOP -

100 IDENT b

101 SEMI ;

102 IDENT e

103 EQUAL =

104 IDENT a

105 MULTOP \*

106 IDENT b

107 SEMI ;

108 IDENT e

109 EQUAL =

110 IDENT a

111 MULTOP /

112 IDENT b

113 SEMI ;

114 IDENT e

115 EQUAL =

116 paren (

117 IDENT a

118 SMALLER <

119 IDENT b

120 paren )

121 IDENT e

122 EQUAL =

123 paren (

124 IDENT a

125 SEQUAL <=

126 IDENT b

127 paren )

128 IDENT e

129 EQUAL =

130 paren (

131 IDENT a

132 GREATER >

133 IDENT b

134 paren )

135 IDENT e

136 EQUAL =

137 paren (

138 IDENT a

139 GEQUAL >=

140 IDENT b

141 paren )

142 IDENT e

143 EQUAL =

144 paren (

145 IDENT a

146 UNEQUAL !=

147 IDENT b

148 paren )

149 IDENT e

150 EQUAL =

151 paren (

152 IDENT a

153 EQUAL ==

154 IDENT b

155 paren )

156 SEMI ;

157 IDENT c2

158 EQUAL =

159 ERROR '\_'

160 SEMI ;

161 IDENT c3

162 EQUAL =

163 CHAR 'Z'

164 SEMI ;

165 IDENT c2

166 EQUAL =

167 CHAR '+'

168 SEMI ;

169 IDENT c3

170 EQUAL =

171 CHAR '\*'

172 SEMI ;

173 IDENT c2

174 EQUAL =

175 ERROR '&'

176 SEMI ;

177 IDENT c3

178 EQUAL =

179 ERROR 'awd'

180 SEMI ;

181 IDENT c2

182 EQUAL =

183 ERROR '1'

184 SEMI ;

185 IDENT num

186 EQUAL =

187 UNSIGNEDINTGER 10023

188 SEMI ;

189 IDENT num

190 EQUAL =

191 ADDOP +

192 UNSIGNEDINTGER 158

193 SEMI ;

194 IDENT num

195 EQUAL =

196 ADDOP -

197 UNSIGNEDINTGER 1239040

198 SEMI ;

199 IDENT e

200 EQUAL =

201 ADDOP +

202 IDENT a

203 ADDOP +

204 IDENT b

205 ADDOP -

206 IDENT add

207 paren (

208 IDENT a

209 COMMA ,

210 IDENT b

211 paren )

212 SEMI ;

213 IDENT e

214 EQUAL =

215 ADDOP -

216 IDENT a

217 ADDOP +

218 IDENT b

219 MULTOP \*

220 IDENT a

221 SEMI ;

222 IDENT e

223 EQUAL =

224 IDENT num1

225 PAREN [

226 UNSIGNEDINTGER 25

227 PAREN ]

228 ADDOP +

229 IDENT a

230 MULTOP \*

231 IDENT b

232 SEMI ;

233 IDENT a

234 EQUAL =

235 UNSIGNEDINTGER 1

236 SEMI ;

237 IDENT num

238 PAREN [

239 UNSIGNEDINTGER 3

240 PAREN ]

241 EQUAL =

242 UNSIGNEDINTGER 2

243 SEMI ;

244 KEYWORD if

245 paren (

246 IDENT a

247 GREATER >

248 UNSIGNEDINTGER 0

249 paren )

250 BRACE {

251 IDENT num1

252 PAREN [

253 UNSIGNEDINTGER 2

254 PAREN ]

255 EQUAL =

256 UNSIGNEDINTGER 1

257 SEMI ;

258 BRACE }

259 KEYWORD if

260 paren (

261 IDENT a

262 ADDOP -

263 UNSIGNEDINTGER 1

264 paren )

265 BRACE {

266 IDENT num1

267 PAREN [

268 UNSIGNEDINTGER 2

269 PAREN ]

270 EQUAL =

271 UNSIGNEDINTGER 0

272 SEMI ;

273 BRACE }

274 KEYWORD while

275 paren (

276 IDENT a

277 GREATER >

278 UNSIGNEDINTGER 0

279 paren )

280 BRACE {

281 IDENT num

282 PAREN [

283 UNSIGNEDINTGER 4

284 PAREN ]

285 EQUAL =

286 IDENT num

287 PAREN [

288 UNSIGNEDINTGER 4

289 PAREN ]

290 ADDOP +

291 UNSIGNEDINTGER 1

292 SEMI ;

293 IDENT a

294 EQUAL =

295 IDENT a

296 ADDOP -

297 UNSIGNEDINTGER 1

298 SEMI ;

299 BRACE }

300 IDENT b

301 EQUAL =

302 UNSIGNEDINTGER 2

303 SEMI ;

304 KEYWORD switch

305 paren (

306 IDENT a

307 paren )

308 BRACE {

309 KEYWORD case

310 IDENT f1

311 COLON :

312 IDENT c

313 EQUAL =

314 IDENT add

315 paren (

316 IDENT a

317 COMMA ,

318 IDENT b

319 paren )

320 SEMI ;

321 KEYWORD case

322 IDENT f2

323 COLON :

324 IDENT ff

325 paren (

326 paren )

327 SEMI ;

328 KEYWORD default

329 COLON :

330 KEYWORD switch

331 paren (

332 IDENT b

333 paren )

334 BRACE {

335 KEYWORD case

336 IDENT f1

337 COLON :

338 IDENT c

339 EQUAL =

340 IDENT add

341 paren (

342 IDENT a

343 ADDOP +

344 UNSIGNEDINTGER 1

345 COMMA ,

346 IDENT b

347 paren )

348 SEMI ;

349 KEYWORD case

350 IDENT f2

351 COLON :

352 IDENT ff

353 paren (

354 paren )

355 SEMI ;

356 KEYWORD default

357 COLON :

358 IDENT d

359 EQUAL =

360 UNSIGNEDINTGER 2

361 SEMI ;

362 BRACE }

363 KEYWORD printf

364 paren (

365 IDENT a

366 ADDOP +

367 UNSIGNEDINTGER 1

368 paren )

369 SEMI ;

370 KEYWORD printf

371 paren (

372 STRING "abc! d ef g"

373 paren )

374 SEMI ;

375 KEYWORD printf

376 paren (

377 STRING "abcde"

378 COMMA ,

379 IDENT a

380 ADDOP +

381 UNSIGNEDINTGER 1

382 paren )

383 SEMI ;

384 BRACE }

结果和预测一样，但是出现了一个问题，比如当printf(“””);遇到这个符号的时候，输出结果会出现错误，但是按照题目要求，字符串不会出现ASCII码为34的”字符，因此不算作错误。