1 CONST const

2 INT int

3 IDENT one

4 EQUMARK =

5 INT 1

6 SEMICOL ;

7 CONST const

8 CHAR char

9 IDENT ch

10 EQUMARK =

11 SINGUO '

12 IDENT c

13 SINGUO '

14 SEMICOL ;

15 INT int

16 IDENT s

17 COMMA ,

18 IDENT sum

19 SEMICOL ;

20 VOID void

21 IDENT getsum

22 LPARENT (

23 INT int

24 IDENT n

25 RPARENT )

26 LBRACE {

27 IF if

28 LPARENT (

29 IDENT n

30 MOR >

31 INT 100

32 RPARENT )

33 SEMICOL ;

34 ELSE else

35 LBRACE {

36 IF if

37 LPARENT (

38 IDENT s

39 LOE <=

40 IDENT n

41 RPARENT )

42 LBRACE {

43 IDENT sum

44 EQUMARK =

45 IDENT sum

46 ADD +

47 IDENT s

48 SEMICOL ;

49 IDENT s

50 EQUMARK =

51 IDENT s

52 ADD +

53 INT 1

54 SEMICOL ;

55 IDENT getsum

56 LPARENT (

57 IDENT n

58 RPARENT )

59 SEMICOL ;

60 PBRACE }

61 ELSE else

62 PRINTF printf

63 LPARENT (

64 DOUQUO "

65 EQUMARK =

66 DOUQUO "

67 COMMA ,

68 IDENT sum

69 RPARENT )

70 SEMICOL ;

71 PBRACE }

72 PBRACE }

73 INT int

74 IDENT reverse

75 LPARENT (

76 INT int

77 IDENT n

78 RPARENT )

79 LBRACE {

80 INT int

81 IDENT j

82 SEMICOL ;

83 INT int

84 IDENT na

85 LBRACK [

86 INT 100

87 RBRACK ]

88 SEMICOL ;

89 IDENT j

90 EQUMARK =

91 INT 0

92 SEMICOL ;

93 IF if

94 LPARENT (

95 IDENT n

96 MOR >

97 INT 100

98 RPARENT )

99 RETURN return

100 SUB -

101 INT 1

102 SEMICOL ;

103 ELSE else

104 SEMICOL ;

105 WHILE while

106 LPARENT (

107 IDENT j

108 LES <

109 IDENT n

110 RPARENT )

111 LBRACE {

112 IDENT na

113 LBRACK [

114 IDENT n

115 SUB -

116 INT 1

117 SUB -

118 IDENT j

119 RBRACK ]

120 EQUMARK =

121 IDENT j

122 ADD +

123 INT 1

124 IDENT j

125 EQUMARK =

126 IDENT j

127 ADD +

128 INT 1

129 SEMICOL ;

130 PBRACE }

131 PRINTF printf

132 LPARENT (

133 DOUQUO "

134 IDENT sum

135 COLON :

136 DOUQUO "

137 RPARENT )

138 SEMICOL ;

139 PRINTF printf

140 LPARENT (

141 IDENT na

142 LBRACK [

143 INT 0

144 RBRACK ]

145 RPARENT )

146 SEMICOL ;

147 IDENT j

148 EQUMARK =

149 INT 1

150 SEMICOL ;

151 WHILE while

152 LPARENT (

153 IDENT j

154 LES <

155 IDENT n

156 RPARENT )

157 LBRACE {

158 PRINTF printf

159 LPARENT (

160 DOUQUO "

161 ADD +

162 DOUQUO "

163 COMMA ,

164 IDENT na

165 LBRACK [

166 IDENT j

167 RBRACK ]

168 RPARENT )

169 SEMICOL ;

170 IDENT j

171 EQUMARK =

172 IDENT j

173 ADD +

174 INT 1

175 SEMICOL ;

176 PBRACE }

177 IF if

178 LPARENT (

179 IDENT n

180 DIV /

181 INT 2

182 MUL \*

183 INT 2

184 EQUAL ==

185 IDENT n

186 RPARENT )

187 RETURN return

188 INT 1

189 SEMICOL ;

190 ELSE else

191 RETURN return

192 INT 0

193 SEMICOL ;

194 PBRACE }

195 VOID void

196 MAIN main

197 LPARENT (

198 RPARENT )

199 LBRACE {

200 INT int

201 IDENT x

202 COMMA ,

203 IDENT y

204 COMMA ,

205 IDENT n

206 COMMA ,

207 IDENT m

208 COMMA ,

209 IDENT i

210 COMMA ,

211 IDENT a

212 SEMICOL ;

213 CHAR char

214 IDENT c

215 COMMA ,

216 IDENT r

217 SEMICOL ;

218 IDENT s

219 EQUMARK =

220 INT 1

221 SEMICOL ;

222 IDENT sum

223 EQUMARK =

224 INT 0

225 SEMICOL ;

226 SCANF scanf

227 LPARENT (

228 IDENT x

229 COMMA ,

230 IDENT y

231 COMMA ,

232 IDENT c

233 COMMA ,

234 IDENT n

235 RPARENT )

236 SEMICOL ;

237 IDENT m

238 EQUMARK =

239 IDENT x

240 SEMICOL ;

241 IDENT a

242 EQUMARK =

243 INT 2

244 SEMICOL ;

245 IDENT i

246 EQUMARK =

247 INT 1

248 SEMICOL ;

249 IF if

250 LPARENT (

251 IDENT x

252 LES <

253 IDENT y

254 RPARENT )

255 LBRACE {

256 IDENT x

257 EQUMARK =

258 IDENT y

259 SEMICOL ;

260 IDENT y

261 EQUMARK =

262 IDENT m

263 SEMICOL ;

264 PBRACE }

265 ELSE else

266 SEMICOL ;

267 WHILE while

268 LPARENT (

269 IDENT i

270 LOE <=

271 IDENT y

272 RPARENT )

273 LBRACE {

274 IF if

275 LPARENT (

276 IDENT x

277 DIV /

278 IDENT i

279 MUL \*

280 IDENT i

281 EQUAL ==

282 IDENT x

283 RPARENT )

284 LBRACE {

285 IF if

286 LPARENT (

287 IDENT y

288 DIV /

289 IDENT i

290 MUL \*

291 IDENT i

292 EQUAL ==

293 IDENT y

294 RPARENT )

295 IDENT a

296 EQUMARK =

297 IDENT i

298 SEMICOL ;

299 ELSE else

300 SEMICOL ;

301 PBRACE }

302 ELSE else

303 SEMICOL ;

304 IDENT i

305 EQUMARK =

306 IDENT i

307 ADD +

308 INT 1

309 SEMICOL ;

310 PBRACE }

311 IDENT m

312 EQUMARK =

313 IDENT x

314 MUL \*

315 IDENT y

316 DIV /

317 IDENT a

318 SEMICOL ;

319 SWITCH switch

320 LPARENT (

321 IDENT c

322 RPARENT )

323 LBRACE {

324 CASE case

325 SINGUO '

326 IDENT g

327 SINGUO '

328 COLON :

329 PRINTF printf

330 LPARENT (

331 DOUQUO "

332 IDENT gcd

333 COLON :

334 DOUQUO "

335 COMMA ,

336 IDENT a

337 RPARENT )

338 SEMICOL ;

339 CASE case

340 SINGUO '

341 IDENT l

342 SINGUO '

343 COLON :

344 PRINTF printf

345 LPARENT (

346 DOUQUO "

347 IDENT lcm

348 COLON :

349 DOUQUO "

350 COMMA ,

351 IDENT m

352 RPARENT )

353 SEMICOL ;

354 DEFAULT default

355 COLON :

356 PRINTF printf

357 LPARENT (

358 DOUQUO "

359 IDENT error

360 DOUQUO "

361 RPARENT )

362 SEMICOL ;

363 PBRACE }

364 IDENT r

365 EQUMARK =

366 IDENT reverse

367 LPARENT (

368 IDENT n

369 RPARENT )

370 SEMICOL ;

371 IDENT getsum

372 LPARENT (

373 IDENT n

374 RPARENT )

375 SEMICOL ;

376 SWITCH switch

377 LPARENT (

378 IDENT r

379 RPARENT )

380 LBRACE {

381 CASE case

382 INT 0

383 COLON :

384 PRINTF printf

385 LPARENT (

386 DOUQUO "

387 IDENT n

388 IDENT is

389 IDENT odd

390 DOUQUO "

391 RPARENT )

392 SEMICOL ;

393 CASE case

394 INT 1

395 COLON :

396 PRINTF printf

397 LPARENT (

398 DOUQUO "

399 IDENT n

400 IDENT is

401 IDENT even

402 DOUQUO "

403 RPARENT )

404 SEMICOL ;

405 DEFAULT default

406 COLON :

407 PRINTF printf

408 LPARENT (

409 DOUQUO "

410 IDENT error

411 DOUQUO "

412 RPARENT )

413 SEMICOL ;

414 PBRACE }

415 PBRACE }

结果与预期相符。