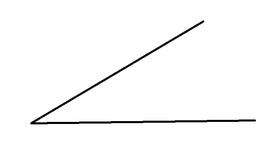
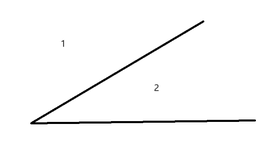
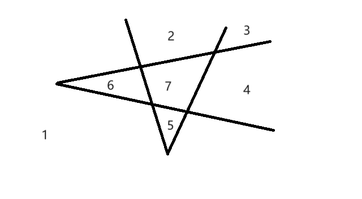
**李华的折线 （时间限制：1秒 占用内存：256 M）**

李华有n条折线，每一条折线都是这样的一个角：  


李华想知道，这n条折线最多能将平面分为多少个区域？

下面是1条折线和2条折线界定区域最多的方法图：

格式

输入格式：

输入一个整数n。

输出格式：

输出一个整数，表示你的答案。

样例

输入：

2

输出：

7

1≤n≤1000。

思路：

// 每增加一个折线，首先折角要产生一个封闭区域

// 然后两条直线把原有的区域三分，也就是增加原有的两倍

代码：

#include<bits/stdc++.h>

using namespace std;

int n, a;

int main() {

cin >> n;

if (n == 1) {

cout << 2;

return 0;

}

a = 2;

for (int i = 2; i <= n; ++i) {

a += (i - 1) \* 2 \* 2 + 1;

}

cout << a;

return 0;

}

判别用例

输入 1

输出 2

输入 43

输出 3656

输入 453

输出 409966

输入 888

输出 1576201

输入 999

输出 1995004

**李华的幸运数 （时间限制：1秒 占用内存：256 M）**

李华是计算机辅助设计图形学与可视化研究所的一名新生，他立志于在计算机图形学领域大展身手。

今天，李华在课上学习了四元数。众所周知，复数是由实数加上虚数单位 i 组成，其中 i^2 = -1 。相似地，四元数都是由实数加上三个虚数单位 i, j 和 k 组成。每个四元数都是 1, i, j 和 k 的线性组合，即四元数一般可表示为 a + bi + cj + dk，其中 a, b, c, d 是整数（注意，此处四元数的定义要求 a, b, c, d 均是整数而不是实数）。我们又称 D = sqrt{a^2 + b^2 + c^2 + d^2} 为该四元数的二范数。

李华认为 n 是他的幸运数字，所以他想知道一共有多少四元数 ( a + bi + cj + dk ) 满足它的二范数的平方为 n 。

输入格式：

第一行一个数 T （1 <=T <= 100 ），表示有 T 组测试样例；

接下来 T 行输入，每行一个数 n （1 <= n <=10^5 ），表示李华的幸运数字。

输出格式：

T行输出，每行一个数，表示满足条件的四元数数量。

输入：

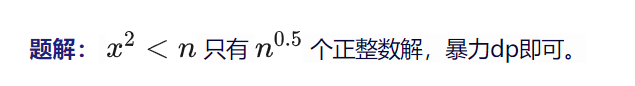
1

1

输出：

8

说明：满足的四元组分别为(1,0,0,0) (-1,0,0,0) (0,1,0,0) …… (0,0,0,-1) 共8个



#include <bits/stdc++.h>

#define LL long long

using namespace std;

// 计算函数

LL cnt(LL x) {

if (x == 0)

return 1;

LL ans = 0;

for (LL i = 1; i \* i <= x; i++) {

if (x % i == 0) {

// 如果因数 i 满足条件，累加 i

if (i & 3)

ans += i;

// 如果因数 x / i 满足条件，累加 x / i（避免平方根重复累加）

if (i \* i != x && ((x / i) & 3))

ans += x / i;

}

}

return ans << 3; // 左移 3 位相当于乘以 8

}

int T; // 测试用例数量

LL x;

signed main() {

scanf("%d", &T); // 输入测试用例数量

while (T--) {

scanf("%lld", &x); // 输入 x

printf("%lld\n", cnt(x)); // 输出结果

}

return 0;

}

**测试1：**

39

52683

1079

64272

76123

82738

48919

63195

47932

1623

70971

61217

78568

33445

33896

96051

75039

17378

22155

2599

35108

49800

34648

26920

403

55072

9305

83799

46977

54905

80832

40077

12602

96303

78168

36996

82795

80097

91763

96692

输出

595584

9408

139776

608992

1018080

435456

884736

300672

17344

776832

560448

285696

321120

107520

1037952

800448

208560

325632

21888

215424

249984

107136

97056

3584

41328

89376

968448

572928

537600

40512

477152

151248

1050624

312768

296064

823680

854400

839040

605952

**测试2：**

93

5304

44531

88165

23525

75518

9105

10059

24473

6772

36883

79193

36593

76139

83900

60933

52882

17649

4267

46048

91857

91231

84558

57989

16891

10353

42633

83037

35877

17802

99036

63133

99815

1533

11111

6839

22538

6383

43723

94204

80763

46905

93317

67113

58592

7134

82925

16922

40752

11720

17391

988

23399

20942

6309

23784

7161

46360

77798

70552

24617

87832

32495

73438

9639

4287

7408

81326

75373

87649

33550

58247

95644

86766

85112

29540

2016

45409

12283

8636

15781

9200

29117

54775

59573

80839

83875

99862

77883

34412

27177

5022

7550

33668

输出：

24192

356256

1059840

233616

922560

116736

122880

195792

40656

368640

633552

321024

710400

624960

684800

642528

213408

36288

34560

996608

834176

1434240

469248

163840

138240

505600

898560

382720

329472

1013760

599040

958272

18944

91392

62592

276480

55104

365184

616896

861504

622080

853248

775632

43968

120960

**测试3：**

74

90937

79342

7643

57146

14200

76333

92933

94605

77399

45513

35052

13861

99319

71013

24151

65694

52394

200

53059

80073

16511

20435

1530

421

35632

32562

93796

29516

15704

85463

92383

92225

12127

70647

75402

97377

68248

1851

43112

36881

9740

78550

54014

26496

73190

46925

24363

54552

15269

22963

61611

50588

87466

56091

49381

40172

88116

36618

26191

62226

38701

23381

32023

34440

95540

28660

77807

6682

53391

57022

77080

10007

33504

28496

输出

907776

952128

61152

685776

53568

610672

748800

1492992

707712

567840

294912

112896

866880

757504

193216

1051200

705024

744

429632

1118208

153600

202368

33696

3376

57024

594048

570240

182016

51072

810240

739072

1142784

99008

753600

1347840

1187328

216000

19776

137376

317856

70272

1169568

656640

7488

1137024

**测试4**

41

31067

23760

84444

28486

91439

29446

27859

40300

6969

23253

7417

58734

24700

91920

11102

62145

50664

48458

87771

81815

27908

97144

98968

53269

20334

64281

5294

71003

88465

931

78704

757

12413

99074

19181

67673

36780

63936

96215

39311

32040

输出

254208

69120

700416

341856

744000

353376

240128

333312

78336

259584

59344

1100736

208320

221184

166656

862368

202752

581520

991872

785472

167472

291456

302400

426160

325440

783872

63552

620160

915264

9120

118080

6064

99312

1188912

153456

583680

353664

36480

1056000

331200

168480

**测试5**

99

78530

47460

60689

34538

603

26659

75548

26495

12941

93984

43947

14851

18147

49114

50346

41588

15855

23347

98294

93255

74328

43925

37188

86274

87770

27545

79906

30033

61490

32719

74604

35485

49141

29185

59906

8936

58367

80700

75045

61256

72774

75417

5704

42508

60056

83629

66169

17432

4738

47301

6770

17896

91050

83740

82059

84160

99211

61389

68540

16014

48123

65630

12860

26692

97529

6913

79741

50457

617

20314

47871

92819

30828

98411

27257

12843

16884

41953

10014

24059

42974

9727

86040

8495

62307

57886

86966

90092

23482

96960

65945

55638

40846

57559

5074

82918

86530

26443

17534

输出

1130976

525312

485520

473856

7072

217728

528768

291072

103536

103680

536640

118816

202752

635040

872976

257184

233472

192128

1477440

1193856

314880

499968

322608

1495728

1292544

302592

958896

359424

1064448

261760

596928

350208

396896

302400

898560

26832

466944

803520

960768

215040

1362816

840192

18432

255072

180192