

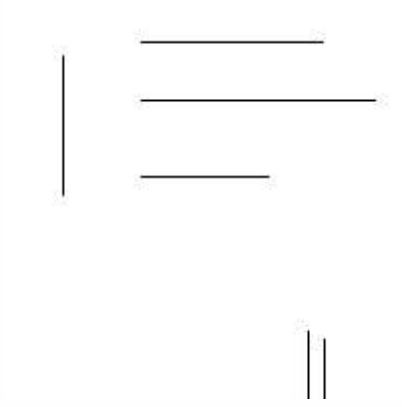
ختیاب

آقای ملکیان که به تازگی فیلم «ماتریکس ۴» را دیده است، به شدت از کیفیت فیلم و پرداخت داستانی آن ناراضی است. او تصمیم می گیرد با سیستم هایی که در اختیار دارد، خود یک دنیای ماتریکسی جدید بسازد تا اینکه بتواند دل مردم را دوباره به این داستان و ادامه اش امیدوار کند و کلید ورود به این دنیا را در اختیار همگان بگذارد. از آنجایی که سر آقای ملکیان بسیار شلوغ است و کار های بسیار زیادی دارد، نمی تواند تمام قسمت های این ماتریکس را خودش بسازد و به ناچار، به دانشجویان درس مبانی برنامه سازی برق زحمت می دهد، زیرا که می داند آنها در این درس، گرافیک می خوانند. آقای ملکیان تا حدی خودش پروژه را به جلو برده است ولیکن، فعلا نمی تواند وارد ماتریکس شود، زیرا که برای وارد شدن به اتاق ها و ساختمان ها و یا نقل و انقالات بین شهری در ماتریکس ساخته شده اول باید بداند به چه تعداد راه و جاده و راهرو و ... وجود دارد.

او تعدادی عکس ساده شده و bird's-eye view از محیط ها تهیه کرده و از شما می خواهد تعداد راه ها در هر عکس را پیدا بکنید. در واقع عکس ها بسیار ساده و با فرمت png هستند. پس زمینه عکس سفید است و روی آن خط های متعددی قرار گرفته است. شما باید کدی بنویسید تا با دادن اسم فایل عکس، تعداد خطوط موجود در آن عکس را بشمرد و آن را چاپ کند.

مثال:

ورودی یک:



خروجی یک:

6

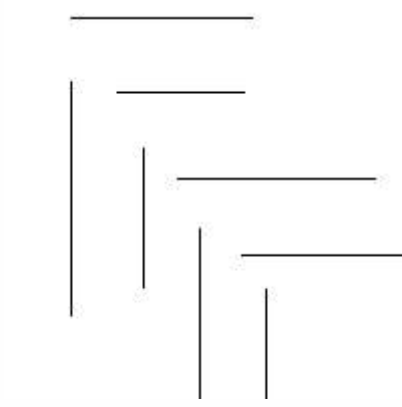
ورودی دو:



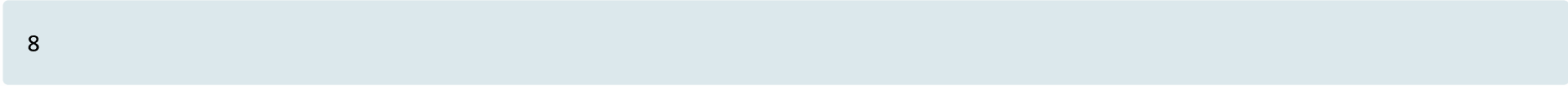
خروجی دو:

11

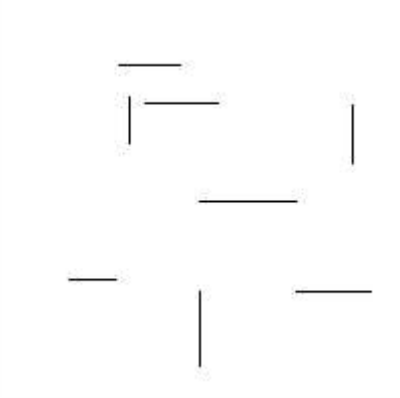
ورودی سه:



خروجی سه:



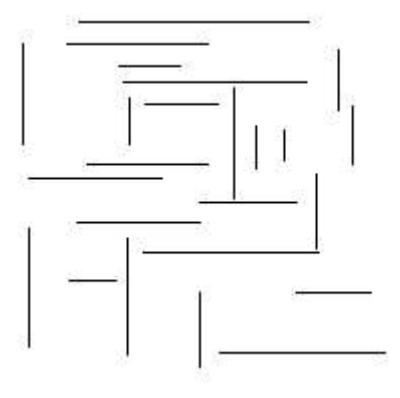
ورودی چهار:



خروجی چهار:



ورودی پنجم:



خروجی پنجم:



تعدادی عکس در لینک بالا در اختیار شما قرار داده شده اند باید برنامه نوشته شده خود را روی آنها تست کنید. برای نمره دهی از عکس هایی مشابه استفاده خواهد شد.

توضیحات عکس ها:

- خط ها همگی صاف هستند یعنی هر خط یا دقیقا عمودی است یا دقیقا افقی و شکستگی نود درجه ندارد.
- عرض و یا ضخامت هر خط دقیقا یک پیکسل است.
- هیچ دو خطی با هم بر خود ندارند.
- رنگ خط ها سیاه است. رنگ پس زمینه سفید است.

کد راهنما:

برای خواندن رنگ های یک پیکسل از یک SDL_Surface از تابع زیرمی توانید استفاده کنید:

```
1 Uint32 getpixel(SDL_Surface *surface, int x, int y)
2 {
3
4     int bpp = surface->format->BytesPerPixel;
5
6
```

```

7      /* Here p is the address to the pixel we want to retrieve */
8
9      Uint8 *p = (Uint8*) surface->pixels + y * surface->pitch + x * bpp;
10
11     switch (bpp)
12     {
13
14
15         case 1:
16
17             return *p;
18
19             break;
20
21         case 2:
22
23             return *(Uint16 *)p;
24
25             break;
26
27         case 3:
28
29             if (SDL_BYTEORDER == SDL_BIG_ENDIAN)
30
31                 return p[0] << 16 | p[1] << 8 | p[2];
32
33             else
34
35                 return p[0] | p[1] << 8 | p[2] << 16;
36
37             break;
38
39         case 4:
40
41             return *(Uint32 *)p;
42
43             break;
44
45         default:
46
47             return 0;          /* shouldn't happen, but avoids warnings */
48
49     }
50
51 }

```

با تابع زیر میتوانید یک uint32 را به b و r ترجمه کنید با توجه به sdl_pixelFormat معین شده در :surface

cpp

```

void SDL_GetRGB(Uint32 pixel,
               const SDL_PixelFormat * format,
               Uint8 * r, Uint8 * g, Uint8 * b);

```