

اعداد کامل

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۶۴ مگابایت

عدد کامل یک عدد صحیح مثبت است که برابر با مجموع مقسوم‌علیه‌های سره‌ی مثبت خود (همه‌ی مقسوم‌علیه‌های مثبتش غیر از خود عدد) باشد.

تابعی بنویسید که بررسی کند یک عدد کامل است یا خیر. سپس با استفاده از این تابع، برنامه‌ای بنویسید که اعداد کامل موجود در بازه‌ی $[a, b]$ را چاپ کند.

ورودی

در خط اول ورودی استاندارد، عدد a و در خط دوم، عدد b وارد می‌شود.

$$1 \leq a \leq b \leq 10000$$

خروجی

هر عدد کامل موجود در بازه را در یک خط مجزا چاپ کنید.

ورودی نمونه ۱

1
100

خروجی نمونه ۱

6
28

ورودی نمونه ۲

300
10000

خروجی نمونه ۲

496
8128

حدس گلدباخ

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۶۴ مگابایت

یکی از قدیمی‌ترین مسائل حل‌نشده‌ی ریاضی، حدس گلدباخ است که به سال ۱۷۴۲ باز می‌گردد.

حدس گلدباخ: هر عدد زوج بزرگ‌تر از ۲ را می‌توان به صورت جمع دو عدد اول نوشت.

تابعی بنویسید که یک عدد زوج را گرفته و همه حالت‌های تشکیل آن عدد از دو عدد اول را چاپ کند.

پ.ن: دانشمندان این کار شما را تا ورودی‌های نزدیک 4×10^{18} انجام دادند، اما کماکان مثال نقضی برای این حدس نیافته‌اند.

ورودی

در تنها خط ورودی، عدد n وارد می‌شود.

$$4 \leq n \leq 10000$$

$$n = 2k$$

(یعنی n حتما زوج است.)

خروجی

در هر خط خروجی بایستی دو عدد اول با یک فاصله چاپ کنید که جمعشان n شود. خطوط بایستی به ترتیب صعودی بر اساس عدد اول چاپ شده در هر خط باشند.

ورودی نمونه ۱

24

خروجی نمونه ۱

5 19

7 17

11 13

ورودی نمونه ۲

100

خروجی نمونه ۲

3 97

11 89

17 83

29 71

41 59

47 53

اعداد خوشحال و ناراحت

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۶۴ مگابایت

عدد طبیعی n را در نظر بگیرید. در هر مرحله، جمع مربعات ارقام n را با خود n جایگزین می‌کنیم. این روند را تا زمانی ادامه می‌دهیم که به عدد ۱ یا ۴ برسیم. در صورتی که n به ۱ برسد، n عددی خوشحال بوده. در غیر این صورت، n عددی ناراحت است.

برای نمونه، به این دو عدد توجه کنید:

$$23 \rightarrow 2^2 + 3^2 = 13 \rightarrow 1^2 + 3^2 = 10 \rightarrow 1^2 + 0^2 = 1$$

پس ۲۳ یک عدد خوشحال است.

$$24 \rightarrow 2^2 + 4^2 = 20 \rightarrow 2^2 + 0^2 = 4$$

پس ۲۴ یک عدد ناراحت است.

تابعی بنویسید که نشان دهد یک عدد خوشحال است یا ناراحت! سپس، برنامه‌ای بنویسید که یک عدد طبیعی از ورودی دریافت کرده و مشخص کند خوشحال است یا ناراحت.

پ.ن: این روند به لحاظ ریاضی اثبات می‌شود که پایان دارد؛ پس نگران بی‌انتهای بودن برنامه‌تان نباشید.

ورودی

در تنها خط ورودی، عدد n وارد می‌شود.

$$1 \leq n \leq 10^4$$

خروجی

در صورتی که عدد خوشحال بود، عبارت HAPPY و در غیر این صورت، عبارت SAD را چاپ کنید.

ورودی نمونه ۱

23

خروجی نمونه ۱

HAPPY

ورودی نمونه ۲

24

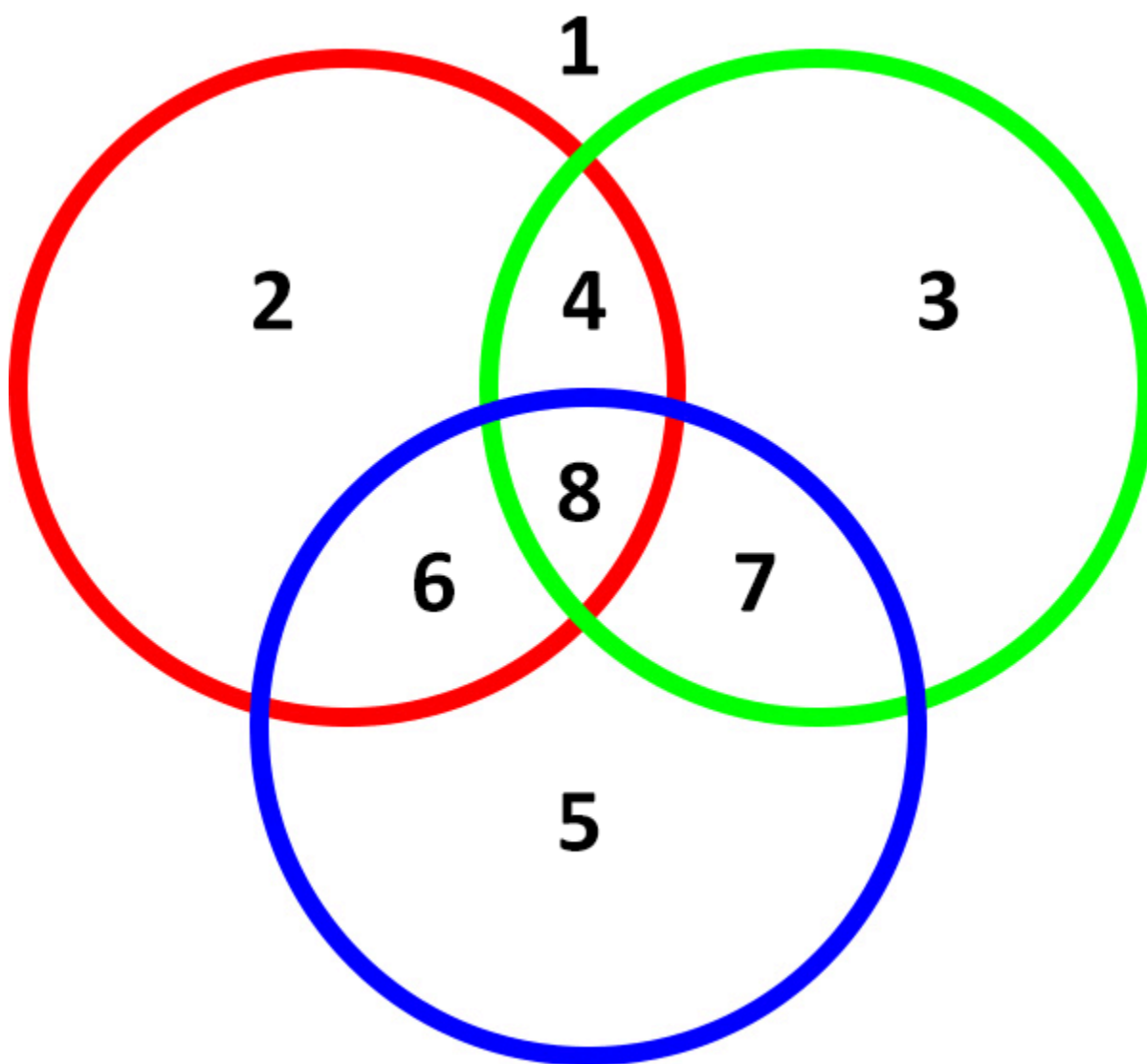
خروجی نمونه ۲

SAD

سه دایره

- محدودیت زمانی: ۱ ثانیه
- محدودیت حافظه: ۶۴ مگابایت

تابعی بنویسید که مشخصات یک دایره و یک نقطه را دریافت کرده و مشخص کند نقطه درون دایره قرار دارد یا خیر. سپس با استفاده از این تابع، برنامه‌ای بنویسید که مشخصات سه دایره و یک نقطه را دریافت کرده و با توجه به شکل زیر، مشخص کند که این نقطه در کدام ناحیه قرار دارد.



ورودی

در خط اول ورودی به ترتیب مختصات x و y مرکز دایره‌ی قرمز و در ادامه، شعاع دایره‌ی قرمز وارد می‌شود.

در خط دوم به ترتیب مختصات x و y مرکز دایره‌ی سبز و در ادامه، شعاع دایره‌ی سبز وارد می‌شود.

در خط سوم به ترتیب مختصات x و y مرکز دایره‌ی آبی و در ادامه، شعاع دایره‌ی آبی وارد می‌شود.

در خط چهارم به ترتیب مختصات x و y نقطه‌ی موردنظر وارد می‌شود.

توجه: تمامی اعداد صحیح هستند.

خروجی

در یک خط از خروجی استاندارد، شماره‌ی ناحیه‌ای که نقطه در آن قرار دارد را چاپ کنید.

ورودی نمونه ۱

```
2 2 8
-2 2 8
0 0 8
0 -1
```

خروجی نمونه ۱

```
8
```

ورودی نمونه ۲

```
3 2 2
6 7 4
8 0 6
1 4
```


خروجی نمونه ۲

1

سود سوزآور

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۶۴ مگابایت
- رئیس یک بانک از شما خواسته توابعی پیاده سازی کنید که مقدار موجودی یک حساب سودده را بعد از چند سال پیش‌بینی کند.
- هر حساب دارای موجودی اولیه، مقدار واریز، برداشت سالانه و سود سالانه است.
- تراکنش‌های برداشت و واریز در اول هر سال انجام می‌شوند ولی محاسبه و واریز سود در آخر هر سال انجام می‌شود.
- در صورتی که موجودی منفی باشد برداشت و واریز اتفاق می‌افتد ولی سود واریز نمی‌شود.
- هر حساب می‌تواند سود مرکب یا ساده داشته باشد. تفاوت این دو مورد به‌صورت زیر است:
 - در حساب مرکب، سود هر سال به موجودی اصلی اضافه شده و همراه موجودی اولیه، مقدار واریز و مقدار برداشت روی سود سال بعد تأثیر می‌گذارد.
 - در حساب ساده، سود هر سال به زیرحسابی فرعی اضافه شده و تأثیری بر سود سال بعد ندارد. سپس در پایان دوره (بعد از duration سال) به موجودی اصلی اضافه می‌شود.
- دو تابع برای محاسبه موجودی در n سال با سود ساده و مرکب پیاده سازی کنید و سپس با استفاده از آن‌ها برنامه‌ای بنویسید که مانند مثال‌های زیر رفتار کند.
- توجه: منطق برنامه‌ی شما باید خارج از تابع main قرار داشته باشد و از تابع main صرفاً برای کار با ورودی و خروجی و فراخوانی سایر توابع استفاده شود.

ورودی

- در یک خط از ورودی استاندارد، ابتدا یک کاراکتر که نشانگر نوع سوددهی حساب است (C برای مرکب و S برای ساده) وارد می‌شود. به بزرگ بودن کاراکتر ورودی دقت کنید.
- سپس، به ترتیب اعداد نشانگر موجودی اولیه، واریز سالانه، برداشت سالانه، سود سالانه (برحسب درصد) و مدت دوره برای محاسبه‌ی سود وارد می‌شود. همه‌ی این اعداد اعشاری هستند. به‌جز مدت دوره که واحد

آن سال است و صحیح نامنفی است.

$$0 \leq balance \leq 1000$$

$$0 \leq deposit \leq 1000$$

$$0 \leq withdrawal \leq 1000$$

$$0 \leq interest\% \leq 50$$

$$0 \leq duration \leq 50$$

خروجی

در یک خط از خروجی استاندارد، مدت دوره و مقدار موجودی در پایان دوره تا سه رقم اعشار به صورتی که در مثال‌ها نمایش داده شده است را چاپ کنید.

ورودی نمونه ۱

C 100 0 0 3 10

خروجی نمونه ۱

Your balance in 10 years will be: 134.392

ورودی نمونه ۲

S 20 1 3 20 23

خروجی نمونه ۲

Your balance in 23 years will be: -8.000

برای محاسبات از double استفاده کنید.

گشت تصادفی

- محدودیت زمان: ۱۰ ثانیه
- محدودیت حافظه: ۶۴ مگابایت

یکی از معروفترین مسائل مدل‌سازی، مسئله‌ی گشت تصادفی است که در آن مسیر حرکت تصادفی یک متحرک در فضای مشخص مطالعه می‌شود. شما قرار است حساب کنید که یک متحرک، در تعدادی گام مشخص، به طور میانگین چند واحد از مبدأ فاصله می‌گیرد.

قوانین گشت به صورت زیر است:

- تعداد گام‌ها در ابتدای برنامه از ورودی استاندارد گرفته می‌شود.
- در هر گام، متحرک به‌طور تصادفی در یکی از جهات محورهای مختصات دکارتی حرکت می‌کند یا در جای خود می‌ماند. برای مثال، در فضای دوبعدی متحرک می‌تواند به چپ، راست، بالا یا پایین حرکت کند یا اصلاً حرکت نکند.
- احتمال هر کدام از حالات ذکر شده با یکدیگر برابر است.
- برای محاسبه‌ی میانگین، باید تعداد n بار شبیه‌سازی را انجام دهید. خروجی هر بار شبیه‌سازی، قدر مطلق فاصله متحرک از مبدا است. سپس بین همه خروجی‌ها میانگین بگیرید.
- انتخاب n به عهده خود شماست. اگر تعداد دفعات کم باشد، میانگین دقت لازم را نخواهد داشت و اگر زیاد باشد، برنامه با محدودیت زمانی مواجه خواهد شد.

ابتدا تابعی پیاده‌سازی کنید که تعداد گام را دریافت کرده و با استفاده از تابع تولیدکننده‌ی عدد تصادفی موجود در کتابخانه استاندارد زبان، حرکت متحرک در فضای یک بعدی را شبیه‌سازی کند و مقدار جابه‌جایی متحرک بعد از تعداد گام مشخص شده را برگرداند.

امتیازی (۵۰ نمره): تابع ذکر شده را برای فضای سه‌بعدی نیز پیاده‌سازی کنید.

سپس تابعی پیاده‌سازی کنید که تعداد گام را دریافت کرده و با دقت قابل‌قبولی میانگین جابه‌جایی متحرک (در تعداد گام مشخص شده) را محاسبه کند. در این تابع باید چند دفعه از تابع قبلی استفاده شود و میانگین خروجی‌های آن در n اجرا حساب شود.

توجه:

- منطق برنامه‌ی شما باید خارج از تابع `main` قرار داشته باشد و از تابع `main` صرفاً برای کار با ورودی و خروجی و فراخوانی سایر توابع استفاده شود. همچنین ممکن است نیاز داشته باشید توابع کمکی تعریف کنید و در توابع اصلی از آن‌ها استفاده کنید.
- همه‌ی بخش‌های تمرین به‌جز تولید عدد تصادفی و محاسبه‌ی ریشه‌ی اعداد باید توسط خودتان پیاده‌سازی شود.

▼ راهنمایی

برای تولید عدد تصادفی، می‌توانید از تابع `rand()` در کتابخانه `stdlib.h` و برای محاسبه‌ی ریشه‌ی اعداد، از کتابخانه `math.h` استفاده کنید.

ورودی

در یک خط از ورودی استاندارد، عدد حسابی s که بیانگر تعداد گام‌ها است وارد می‌شود.

$$0 \leq s \leq 10000$$

خروجی

در یک خط از خروجی استاندارد، میانگین فاصله‌ی متحرک از مبدأ را با یک رقم اعشار چاپ کنید. به فرمت خروجی نهایت دقت را داشته باشید.

ورودی نمونه ۱

3

خروجی نمونه ۱

Average disposition of a one dimensional random walk in 3 steps is: 1.1 unit

ورودی نمونه ۲ (امتیازی)

210

خروجی نمونه ۲ (امتیازی)

Average disposition of a one dimensional random walk in 210 steps is: 9.4 unit
Average disposition of a three dimensional random walk in 210 steps is: 12.6 u

توجه: داوری این سوال با سوالات دیگر متفاوت است بنابراین ممکن است مقدار بیشتری طول بکشد.
نگران نباشید. همچنین، در صورتی که از کتابخانه‌ی `math.h` استفاده کرده‌اید، زبان `C++` را هنگام ارسال
کد انتخاب کنید.

توجه: برای قسمت غیرامتیازی، ۱ بُعد لازم است یعنی فقط راست و چپ برود. مثالی که برای دو بعد زده شده فقط برای
تقریب به ذهن است.