انتگرال گاوسی

برنامه‌ای بنویسید که مقدار انتگرال معین تابع n متغیره زیر را به ازای ماتریس مربعی متقارن A و بازه‌های داده شده برای متغیرهای آن محاسبه کرده و نتیجه را تا دو رقم اعشار گرد کرده و نمایش دهد.

$$\int_{\hat{a}}^{\hat{b}} \exp(-\hat{x}^t \cdot A \cdot \hat{x}) d\hat{x} = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} \exp(-\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}x_i x_j) dx_n \dots dx_1$$

نکته: برای حل عددی این انتگرال از روش ذوزنقه‌ای با طول گام های برابر با 0.01 برای همه متغیرهای آن استفاده شود.

راهنمایی: در حل این سوال بهتر است برای سرعت عملکرد بالاتر کد هنگام کار کردن با بردارها و ماتریس‌ها، از کتابخانه numpy استفاده گردد و در کنار آن برای اجتناب از حلقه‌های تو در تو می‌توان از مفهوم توابع بازگشتی استفاده شود بدین صورت که به کمک تقسیم‌بندی بلوکی ماتریس A در ضرب ماتریسی موجود و انجام انتگرال روی یک متغیر از تعداد متغیرها در هر مرحله یکی کاست به طوری که در هر مرحله حاصل انتگرال با k متغیر را وابسته به حاصل انتگرالی با k-1 متغیر کرد و بدین ترتیب به در نهایت به یک انتگرال یگانه رسید که با بدست آوردن مقدار آن و جایگذاری در نتایج مراحل قبلی در نهایت حاصل کل انتگرال را بدست آورد. البته لازم به ذکر است که این روش در صورت افزایش تعداد متغیرها به علت ساختار بازگشتی به شدت کند خواهد بود لذا می توان از توابع خانواده meshgrid کتابخانه numpy استفاده کرد که می‌تواند کل دامنه مسئله را مشبندی کند و با برداری‌سازی محاسبات دیگر نیازی به استفاده از حلقه‌های for که سرعت اجرا کد را کاهش می‌دهند نباشد اما این روش نیز مشکل با مشکل کمبود حافظه در تعداد متغیر بالا و بازه های انتگرال‌گیری بزرگ مواجه می‌شود. لذا بهترین رویکرد تلفیق مناسب این دو رویکرد است تا به سرعت مناسب اجرای کد به همراه مصرف بهینه حافظه دست یافت.

ورودی

ورودی کد می بایست در سه خط باشد که در خط اول، می‌بایست ماتریس A تعیین شود به این صورت که هر سطر از ماتریس به ترتیب از سطر اول تا آخر پشت سر هم وارد شده و با یک کاما و فاصله (,) از یکدیگر تفکیک می‌شوند و درایه های موجود در هر سطر نیز با یک فاصله از همدیگر جدا می‌شوند. در خط

دوم، مقادیر کران پایین (a) همه متغیرها متناظر با ترتیب متغیرها و با یک فاصله از همدیگر وارد می‌شوند و در خط بعدی نیز مقادیر کران بالای (b) متناظر با هر متغیر به ترتیب و با یک فاصله از یکدیگر تعیین می‌شود.

خروجی

خروجی برنامه نیز مقدار محاسبه شده برای انتگرال مذکور خواهد بود که تا دقت دو رقم اعشار گرد شده‌است و در خروجی با همان **فرمت دو رقم اعشار** نمایش داده می‌شود.

مثال

ورودی نمونه ۱

```
1
-1
2.5
```

خروجی نمونه ۱

```
1.63
```

ورودی نمونه ۲

```
1 0 0 0, 0 1 0 0, 0 0 1 0, 0 0 0 1
-2 0 -11.2 0.7
2 1 15.2 1.3
```

خروجی نمونه ۲

```
0.53
```

ورودی نمونه ۳

```
1 0.5, 0.5 1
-2 -1
3 5.5
```

خروجی نمونه ۲

```
3.20
```