



پروژه ی دوم برنامه سازی پیشرفته

دکتر سیدابوالقاسم میرروشندل

Number Commands

طراح: امیرحسین مهدی نژاد

داستان پروژه

مشهدی غلام حسین که از کار با اعداد لذت می برد، در اوقات فراغتش تصمیم می گیرد پروژه ای به زبان جاوا بنویسد که دستورات مورد علاقه اش را روی اعداد پیاده سازی کند.

در این مسیر، وی با دو مشکل اساسی مواجه می شود:

۱. در جاوا (و بسیاری از زبان های برنامه نویسی دیگر) باید با اعداد بزرگتر از long، به عنوان رشته ی کاراکتری رفتار شود.

۲. مشهدی غلام حسین کار با رشته ها را بلد نیست!

حال از شما به عنوان یک برنامه نویس تازه کار، درخواست می شود با استفاده از لیست دستوراتی که وی مد نظر دارد، این پروژه را برایش انجام دهید.

هدف از پروژه

هدف این پروژه آشنایی بیشتر با رشته ها در جاوا، و به نوعی آشنایی با پیاده سازی یک کلاس BigInteger به صورت دستی، برای کار با اعداد بزرگ است.

در این پروژه می بایست برنامه ای بنویسید که یک خط شامل چندین فرمان را از ورودی کنسول System.in دریافت کرده و خروجی هریک را به ترتیب در خطی جداگانه نمایش دهد.

دستورات مجاز به شکل زیر می باشند:

- دستور add

این دستور، دو رشته را به عنوان ورودی گرفته، و جمع آنها را باز می گرداند. به طور مثال $\text{add}(123, 321)$ مقدار ۴۴۴ را خروجی می دهد.

- دستور sub

این دستور، دو رشته را به عنوان ورودی گرفته، و مقدار اولی منهای دومی را به عنوان خروجی باز می گرداند. به طور مثال $\text{sub}(123, 321)$ مقدار ۱۹۸- را خروجی می دهد.

- دستور pow

این دستور، دو رشته را به عنوان ورودی گرفته، و رشته ی اول را به توان رشته ی دوم می رساند. به طور مثال $\text{pow}(1, 23)$ مقدار ۱ را خروجی می دهد.

- دستور fact

این دستور یک رشته را به عنوان ورودی گرفته و مقدار فاکتوریل آن را به عنوان خروجی باز می گرداند. به طور مثال $\text{fact}(3)$ مقدار ۶ را خروجی می دهد.

- دستور nextPerm

این دستور، یک رشته را به عنوان ورودی گرفته و کوچکترین عدد بزرگتر از این عدد ورودی که از جایگشت همین ارقام به دست می آید را به عنوان خروجی، بر می گرداند و در صورتی که جواب مناسب وجود نداشته باشد ۱- برگرداند. برای مثال $\text{nextPerm}(123)$ مقدار ۱۳۲ را خروجی می دهد.

- دستور mod

این دستور، دو رشته را به عنوان ورودی گرفته و باقیمانده ی اولی بر دومی را به عنوان خروجی باز می گرداند. به طور مثال $\text{mod}(321, 123)$ مقدار ۷۵ را خروجی می دهد.

- دستور rep

این دستور، دو رشته را به عنوان ورودی گرفته و به تعداد رشته ی دوم، رشته ی اول را تکرار می کند و عدد نهایی را به عنوان خروجی بر می گرداند. برای مثال $\text{rep}(12, 3)$ مقدار ۱۲۱۲۱۲ را خروجی می دهد.

- دستور isPalindromes

این دستور دو آرگومان رشته از ورودی دریافت می کند و چک می کند آیا این دو عکس یکدیگر هستند یا خیر. (یک مقدار Boolean بر می گرداند) خروجی را با "True" یا "False" نمایش دهید.

به طور مثال ۱۲۳ و ۳۲۱ عکس یکدیگر هستند.

- دستور sumOfDigits

این دستور یک رشته به عنوان آرگومان دریافت می کند و مجموع ارقام آنرا باز می گرداند. برای مثال sumOfDigits(۱۲۳) مقدار ۶ را خروجی می دهد.

- دستور sort

همان طور که از اسم دستور مشخص است، این دستور وظیفه ی مرتب سازی ارقام عدد داده شده به آن را دارد و باید به صورت صعودی آنها را مرتب کند.

- دستور rotate

این دستور وظیفه دارد که رشته ای را به تعداد مشخصی و به جهت مشخصی دوران بدهد به طوری که جهت دوران با مثبت یا منفی بودن عدد دوران تعیین می شود:

۱. در صورت مثبت بودن عدد دوران، رشته ی ورودی به راست دوران داده می شود.

برای مثال اگر rotate(۱۲۳,۳) فراخوانی شود، رشته ی ورودی پس از ۱ دوران به ۳۱۲، پس از ۲ دوران به ۲۳۱ و پس از ۳ دوران به ۱۲۳ تبدیل می شود.

۲. در صورت منفی بودن عدد دوران، رشته ی ورودی به چپ دوران داده می شود.

برای مثال اگر rotate(۱۲۳,-۴) فراخوانی شود، رشته ی ورودی پس از ۱ دوران به ۲۳۱، پس از ۲ دوران به ۳۱۲، پس از ۳ دوران به ۱۲۳ و پس از ۴ دوران به ۲۳۱ تبدیل می شود.

- دستور indexOf

این دستور وظیفه دارد index شروع رشته ای را در رشته ی دیگر پیدا کرده و در صورت وجود، شماره ی index مورد نظر را نمایش دهد و در غیر این صورت -۱ را نمایش دهد.

- دستور subNumber

این دستور ۳ ورودی به عنوان آرگومان دریافت می کند که این ۳ آرگومان شامل یک رشته و دو عدد صحیح است که اولین عدد، شماره ی index شروع و دومین عدد شماره ی index پایانی است. وظیفه ی این دستور، پیدا کردن زیررشته ی بین شروع و پایان وارد شده در رشته و نمایش دادن آن است و اگر یکی از index های وارد شده، خارج از محدوده ی شماره index های رشته بود، باید مقدار null را نمایش دهد.

- دستور split

این دستور، دو رشته از ورودی می گیرد و سپس در رشته ی اول، قسمت هایی که با رشته ی دوم برابرند را حذف کرده و در آخر قسمت های جدا شده (قسمت هایی که پس از حذف شدن رشته های بینشان جدا شده اند) را با استفاده از یک underline (کاراکتر_) در بین آنها به یکدیگر می چسباند و رشته ی تولید شده را باز می گرداند.

ورودی

برنامه تنها شامل یک خط ورودی است که در آن دستورات با underline از هم جدا شده اند، این دستورات هریک ابتدا شامل عنوان دستور می باشند و سپس ورودی های مربوط به آن دستور در یک جفت پرانتز قرار می گیرد. هر رشته حتماً زیر ۲۰۰ کاراکتر بوده و همچنین تضمین می شود که دستورات در ورودی، قابل خروجی دادن باشند.

خروجی

به ازای هر دستور در ورودی، متناظر با آن در هر خط، خروجی آن دستور قرار می گیرد.

ورودی های نمونه

Input1

```
pow(2,3)_add(5,6)_sub(7,8)_nextPerm(9)_rep(9,8)_mod(7,6)
```

Output1

```
8
11
-1
-1
99999999
1
```

Input2

```
fact(20)_sumOfDigits(89101112131415161718)_isPalindromes(121,454)
```

Output2

```
2432902008176640000
62
False
```

نکات تکمیلی

- بدیهی است که حق استفاده از توابع پیشفرض جاوا که کارهایی مشابه با دستورات مدنظر را انجام می دهند، ندارید و الزامی است تمامی توابع را خودتان پیاده سازی کنید.
- برای کسب نمره ی کامل پروژه، از استفاده از کلاس های آماده ی جاوا مانند BigInteger برای دریافت اعداد ورودی، بپرهیزید.
- برای سادگی، هر فرمان از فرمان دیگر مستقل بوده و امکان ترکیب شدن فرمان ها، استفاده از خروجی یک فرمان برای ورودی فرمان دیگر، وجود ندارد اما پیاده سازی درست این قابلیت **نمره ی مثبت** دارد.
- اضافه کردن کلاس های دیگر به برنامه، در این پروژه الزامی نیست اما مانعی ندارد.

نحوه ی ارسال پروژه

- هیچ توجیهی در استفاده از زبان های برنامه نویسی دیگر وجود نداشته، و برابر با تحویل ندادن پروژه، **نمره ی صفر** به دانشجو تعلق می گیرد.
- پروژه تنها بصورت انفرادی قابل انجام بوده و در صورت وجود هر گونه تشابه بین دو کد یا عدم تسلط به روند برنامه در زمان تحویل، **نمره ی منفی** به دانشجویمان داده می شود.
- برای تسلط بهتر و کسب نمره خوب در این درس به انجام پروژه های اضافه بر پروژه های ترم اقدام کنید.
- زودتر از deadline پروژه از طریق quera اقدام به ارسال نموده و تأیید گرفته شده را بلافاصله چک کنید.
- رعایت فاصله ی کد از سر خط و همچنین comment نویسی مناسب **الزامی** است.