به نام خدا

|  |  |  |  |
| --- | --- | --- | --- |
| مدرس: مظفر بگ محمدي | دانشگاه ايلام | ترم دوم سال تحصيلي 97-96 | سری دوم تمرینات برنامه‌نویسی جاوا |

سوالات ساده با رنگ سبز، سوالات معمولی با رنگ نارنجی و سوالات مشکل به رنگ قرمز مشخص شده اند.

1. **آرایه**: متدی بنویسید که یک آرایه و یک عدد را دریافت نماید و مقدار بزرگترین عنصری از آرایه را برگرداند که مقدار آن کمتر یا مساوی عدد داده شده است. اگر همه ی عناصر آرایه از عدد داده شده بزگتر باشند، خروجی -1 خواهد بود. به مثالهای زیر توجه کنید.

|  |  |  |
| --- | --- | --- |
| **آرایه** | **عدد** | **خروجی** |
| {234, 528, 235, 253, 400} | 300 | 253 |
| {98, 70, 72} | 72 | 72 |
| {900, 885, 989, 1} | 880 | 1 |
| {500, 300, 241, 99, 501} | 50 | -1 |
| {200} | 120 | 200 |

1. **رشته**: متدی به اسم compress بنویسید که یک رشته دریافت کند و تمام کاراکترهای تکراری را به جز بار اول حذف کند. دقت کنید که ترتیب کاراکترها نیز باید حفظ شوند. (20 نمره)

uniqueChar("whatisyourname?") → “whatisyournme”

uniqueChar("I love java") → “I loveja”

uniqueChar("hello Joe") → “helo Je”

1. **آرایه**: متدی بنویسید که یک آرایه را دریافت نماید و ایندکس اولین محلی از آرایه از سمت چپ را برگرداند که جمع عناصر ما قبل آن از جمع عناصر ما بعد آن بزرگتر باشد.

3, 5, 1, 7, 2 🡪 3زیرا 3+5+1+7>2

0, -2 , 1 , 3, 3, 2, 1, 1 🡪 4 زیرا (0-2+1+3+3>2+1+1)

12, 4, 1, 1 , 1, -3, 2, 0🡪 0 زیرا (12>4+1+1+1-3+2+0)

1. **رشته**: متدی به اسم binaryPart بنویسید که یک رشته را دریافت کند و معادل دهدهی بزرگترین عدد دودویی موجود در رشته را پیدا کند. به مثالهای زیر توجه کنید. (20 نمره)

binaryPart("whati101s11your0nam100e?") → 5

binaryPart ("I love java") → 0

binaryPart ("hello J10101110oe") → 174

1. **آرایه**: فرض کنید که مجاورت را به صورت عناصر مجاور آرایه با مقدار یکسان تعریف کنیم. متدی به اسم countClumps(int[] a) بنویسید که یک آرایه را دریافت کند و تعداد مجاورتهای آنرا چاپ کند. به مثالهای زیر توجه کنید:

countClumps([1, **2, 2**, 3, **4, 4**]) → 2

countClumps([**1, 1**, 2, **1, 1**]) → 2

countClumps([**1, 1, 1, 1, 1**]) → 1

1. **آرایه**: فرض کنید یک عنصر در دو جای آرایه تکرار شده است و تعدادی عنصر نیز بین آنها قرار دارند. نرخ گسترش آرایه برابر است با تعداد عناصر مابین دو عنصر تکراری و خود عناصر تکراری. متدی به اسم maxSpan(int[] a) بنویسید که یک آرایه را دریافت کند و حداکثر نرخ گسترش آنرا چاپ کند. به مثالهای زیر توجه کنید:

maxSpan([**1, 2, 1, 1**, 3]) → 4

maxSpan([1, **4, 2, 1, 4, 1, 4**]) → 6

maxSpan([1, **4, 2, 1, 4, 4, 4**]) → 6

1. **آرایه**: فرض کنید که مقادیر یک آرایه نشان دهنده ارتفاع یک کوهنورد بر حسب متر باشد و اندیس آرایه نیز نشان دهنده ی زمان بر حسب ساعت باشد. به عنوان مثال، a[0]=5 یعنی کوهنورد در زمان صفر در ارتفاع 5 متر قرار دارد. اختلاف بین a[i] و a[i-1] نیز برابر است با مسافتی که کوهنورد در فاصله ی زمانی i-1 تا i که با یک ساعت برابر است پیموده است. متدی بنویسید که برای آرایه ی داده شده، مشخص کند که کوهنورد حداکثر چند ساعت متوالی در حال صعود بوده است؟
2. **رشته**: متدی بنویسید که تعداد کلماتی که آخر آنها y یا z است را در رشته­ی ورودی پیدا کند. مثالها:

|  |  |
| --- | --- |
| ورودی | خروجی |
| fez or day | 2 |
| day fyyyz | 2 |
| Hi You | 0 |

1. **رشته**: متدی بنویسید که تعداد دفعات تکرار رشته­ی دوم را در رشته­ی ورودی پیدا کند.
2. **رشته**: متدی بنویسید که دو رشته دریافت کند و در رشته­ی اول بگردد و تمام جاهایی که رشته­ی دوم تکرار شده است را حذف کند. به مثالهای زیر توجه کنید:

|  |  |  |
| --- | --- | --- |
| رشته­ی­ اول | رشته­ی­ دوم | خروجی |
| Hello there | llo | He there |
| Hello there | e | Hllo thr |
| Hello there | x | Hello there |

1. **رشته**: متدی بنویسید که یک رشته به اسم s دریافت کند و رشته ی جدیدی تولید کند که از جایجایی کاراکترهای رشته ی قبلی به دست آمده است. برای الگوی جابجایی از نابع f(i)=(i+7)mod(s.length()) استفاده کنید. مثلاً اگر طول رشته برابر 5 باشد، کاراکتر سوم به محل 10 mod 5 یعنی 0 منتقل می گردد.
2. **رشته:** متدی بنویسید که یک رشته را دریافت کند و اگر بین حروف a و b دو کاراکتر وجود داشته باشد، جای a و b را عوض کند. به مثالهای زیر توجه کنید:
	1. “sdaxxbcs”🡪”sdbxxacs”
	2. “sdaxxbyyacs”🡪” sdbxxayybcs”