

سوال ۱: تابعی بنویسید که رشته ای را به عنوان آرگومان دریافت کند و اولین عدد صحیح رشته از سمت چپ را برگرداند.

Example:

leftDigit("TrAdE2W1n95!") → 2

leftDigit("V3r1ta\$") → 3

leftDigit("UDertHe1nflu3nC3") → 1

سوال ۲: تابعی ایجاد کنید که تعداد بلوک های دو یا چند 1 مجاور را در یک آرایه بشمارد. به عنوان مثال برای آرایه [1, 0, 0, 1], 1, 0, 1, 1, 1, 0, 1 دو مورد داریم: [1, 1] (در اواسط) و [1, 1, 1] (در انتها)

Example:

countOnes([1, 0, 0, 1, 1, 0, 1, 1, 1]) → 2

countOnes([1, 0, 1, 0, 1, 0, 1, 0]) → 0

countOnes([1, 1, 1, 1, 0, 0, 0, 0]) → 1

سوال ۳: یک آرایه و دو عدد داریم. این دو عدد، ایندکس شروع و پایان یک آرایه داخلی را نشان می دهند. تابعی بنویسید که آرایه داخلی موجود در این محدوده را معکوس کند. برای درک بهتر به مثال زیر توجه کنید:

rangedReversal([1, 2, 3, 4, 5, 6], 1, 3) → [1, 4, 3, 2, 5, 6]

از ایندکس 1 تا 3 یعنی [2, 3, 4] باید معکوس شوند. اعداد 1 و 5 و 6 در جای خود باقی می مانند.

Example:

rangedReversal([1, 2, 3, 4, 5, 6], 1, 3) → [1, 4, 3, 2, 5, 6]

rangedReversal([1, 2, 3, 4, 5, 6], 0, 4) → [5, 4, 3, 2, 1, 6]

rangedReversal([9, 8, 7, 4], 0, 0) → [9, 8, 7, 4]

سوال ۴: تابعی ایجاد کنید که بررسی کند آیا یک عدد صحیح داده شده فاکتوریل یک عدد صحیح دیگر است یا خیر.

Example:

isFactorial(2) → True

isFactorial(27) → False

isFactorial(24) → True

سوال ۵: در این تمرین قصد داریم آرایه ای از اعداد اعشاری بسازیم. آرایه ای از مقادیر بین 0 و یک نقطه پایانی n با فواصل زمانی i ایجاد کنید.

Example:

genValues(2, 0.25) → [0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0]

genValues(1, 0.1) → [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]

genValues(20, 2) → [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

سوال ۶: تابعی بنویسید که یک عدد صحیح n را بگیرد، نمایش باینری آن عدد صحیح را معکوس کند و باینری معکوس شده را دوباره به عدد صحیح برگرداند.

Example:

`reversedBinaryInteger(10) → 5`

`reversedBinaryInteger(12) → 3`

`reversedBinaryInteger(25) → 19`

سوال ۷: تابعی ایجاد کنید که آرایه‌ای از اعداد را بگیرد و بزرگترین مقسوم علیه مشترک آن اعداد را برمی گرداند.

Example:

`gcd([84, 70, 42, 56]) → 14`

`gcd([19, 38, 76, 133]) → 19`

`gcd([120, 300, 95, 425, 625]) → 5`

سوال ۸: شما در یک کارخانه کار می کنید و وظیفه شما این است که اقلام را از یک تسمه نقاله بردارید و آنها را در جعبه ها بسته بندی کنید. هر جعبه می تواند حداکثر 10 کیلوگرم را در خود جای دهد. با داشتن آرایه‌ای حاوی وزن (به کیلوگرم) هر کالا در ورودی، برای بسته بندی همه اقلام به چند جعبه نیاز دارید؟ (برای درک بهتر به مثال زیر توجه کنید)

Example:

`boxes([2, 1, 2, 5, 4, 3, 6, 1, 1, 9, 3, 2]) → 5`

Box 1 = [2, 1, 2, 5] (10kg)

Box 2 = [4, 3] (7kg)

Box 3 = [6, 1, 1] (8kg)

Box 4 = [9] (9kg)

Box 5 = [3, 2] (5kg)

سوال ۹: در این تمرین آرایه‌ای از رشته های متشکل از حروف a - z به شما داده می شود. این آرایه، آرایه‌ای از اعداد اول متوالی است که به یک رمز جایگزین ساده تبدیل شده اند. هر یک از اعداد (0-9) با یکی از حروف a - z جایگزین شده و رمزنگاری شده اند. در هر تست، طرح جایگزینی برای همه اعضای آرایه یکسان است.

وظیفه شما ایجاد تابعی است که آرایه اعداد اول را از آرایه رشته های رمز شده بازیابی کند.

Example:

`cryptoprimes(["bb", "bi", "bg", "bc"]) → [11, 13, 17, 19]`

`cryptoprimes(["b", "c", "a", "i"]) → [2, 3, 5, 7]`

`cryptoprimes(["fgf", "fgb", "fgi", "fgd", "ffb"]) → [101, 103, 107, 109, 113]`

سوال ۱۰: تابعی بنویسید که رشته ای از اعداد را در ورودی بگیرد و طولانی ترین رشته ای که دارای ارقام فرد/ زوج یا زوج/فرد یکی در میان است را در خروجی برگرداند. اگر دو یا چند زیررشته دارای طول یکسانی باشند، زیر رشته را که اول پیدا شده است برگردانید.

در زیر، زیر رشته های پیدا شده در هر رشته مشخص شده است و طولانی ترین آنها در خروجی برگردانده شده است.

Example:

`longestSubstring("225424272163254474441338664823")` → "272163254"

`longestSubstring("594127169973391692147228678476")` → "16921472"

`longestSubstring("721449827599186159274227324466")` → "7214"