|  |  |  |
| --- | --- | --- |
| **تمرين سري چهارم** | **مباني برنامه‌نويسي** | **نيم‌سال اول 9۷-9۶** |

***به دو نكته توجه كنيد:***

الف)‌ زمان تحويل تا قبل از 22:00 روز شنبه 2۷ آبان ماه است.

ب) پاسخ مسائل را در سايت Quera ارسال نماييد.

1- دو عدد a و b در پيمانه m همنهشت هستند اگر باقيمانده a به m و باقيمانده b به m باهم برابر باشد. برنامه‌اي بنويسيد كه عدد x و y را از كاربر بگيرد و همه اعداد mی كه x و y در پيمانه m باهم همنهشت هستند را توليد كند براي مثال

ورودي: X = 1, Y = 25

خروجي: m= 2, 3, 4, 6, 8, 12, 24

فرمت ورودي:

<x>

<y>

فرمت خروجي:

<m> <m> <m> … <m>

۲- عدد كامل عددي است كه برابر حاصل جمع مقسوم عليه‌هاي كوچكتر از خود باشد براي مثال عدد 6 كامل است زيرا 6 = 1 + 2 +‌ 3. برنامه‌اي بنويسيد كه عدد n را از كاربر گرفته، تعداد اعداد كامل كمتر-مساوي n و نیز خود اعداد كامل را چاپ كند.

فرمت ورودي:

<n>

فرمت خروجي:

# of complete numbers: <m>

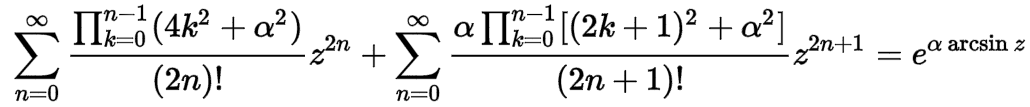
<x1>

<x2>

…

<xm>

۳- نشان داده میشود که مقدار برای از رابطه زیر قابل محاسبه است.



برنامه‌ای بنویسید که مقدار و و عدد n را از کاربر بگیرد و را با دقت n رقم بعد از اعشار محاسبه کند. در این سوال نمی‌توانید از کتابخانه‌ math استفاده کنید.

فرمت ورودي:

<alpha>

<z>

<n>

فرمت خروجي:

Result= <x>

۴- برنامه‌اي بنويسيد كه دو عدد صحيح a و b را از كاربر بگيرد و عدد a را به مبناي b ببرد كه آنرا c مي‌ناميم. اگر مجموع ارقام مكان‌هاي فرد برابر مجموع ارقام مكان‌هاي زوج c باشد True و در غير اين صورت False چاپ شود. برای مثال

a = 15 , b = 2 🡪 True

اختیاری: این برنامه بدون استفاده از آرایه.

فرمت ورودي:

<a>

<b>

فرمت خروجي:

True/False

۵- **تست سرعت و دقت تایپ**

هدف از این برنامه تست سرعت و دقت تایپ کاربر است. از این تست نمونه‌های خیلی زیادی وجود دارد برای مثال <https://www.typingtest.com>. در این تمرین قصد داریم یک نمونه از آن را ما نیز پیاده‌سازی کنیم. روال تست به شرح زیر است.

با اجرای برنامه دو عدد length و num از کاربر پرسیده می‌شود. Length تعداد کاراکترهایی است که در هر تست وجود دارد و num هم تعداد تستها است. با گرفتن این اطلاعات از کاربر، برنامه به تعداد num بار مراحل زیر تکرار می‌کند:

* یک رشته از کاراکترها فقط شامل حروف a-z و فاصله به طول length به صورت تصادفی تولید می‌شود.
* رشته تولید شده به کاربر نشان داده می‌شود
* زمان سیستم در یک متغیر ذخیره می‌شود.
* رشته‌ای که کاربر وارد کرده است خوانده می‌شود.
* زمان سیستم در یک متغیر جدید ذخیره شده و اختلاف زمانی محاسبه می‌شود.
* رشته‌ای که کار وارد کرده است با رشته‌ای که به وی نشان داده شده است کاراکتر به کاراکتر مقایسه شده و تعداد کاراکترهای غلط وارد شده محاسبه می‌شود.

بعد از تکرار این مراحل متوسط تعداد غلطها و متوسط زمان به کاربر گزارش میشود و مجددا اعداد num و length سوال می‌گردد و دوباره فرآیند تکرار می‌شود. این برنامه زمانی تمام می‌شود که کاربر num را صفر وارد کند.

**اختیاری**: علاوه بر length و num یک عدد دیگر به اسم level از کاربر سوال میشود که میزان پیچیدگی را مشخص می‌کند. این عدد به شرح زیر مشخص می‌کند که چه کاراکترهایی می‌تواند در رشته وجود داشته باشد.

|  |  |
| --- | --- |
| مقدار level | کاراکترهای مجاز |
| 1 | [a , z], ' ' |
| 2 | [a, z], [0, 9] , ' ' |
| 3 | [a, z], [0, 9], [A, Z] , ' ' |
| 4 | [a, z], [0, 9], [A, Z], {!@#$%^&\*()}, ' ' |

نکات پیاده‌سازی:

الف) برای خواندن رشته‌ای که کاربر وارد کرده است بهتر است از gets استفاده کنید.

ب) برای محاسبه اختلاف زمانی روش‌های مختلفی وجود دارد، Google بفرمایید لطفا!