1. چالش اولیه برنامه‌نویسی و تغییرات آن

- نوشتن برنامه‌ای که عدد صحیحی را به عنوان ورودی دریافت کرده و الگوی مثلثی راست‌چین از ستاره‌ها چاپ کند.
  - تعداد سطرهای مثلث برابر با عدد ورودی است.
  - مثلث راست‌چین است و تعداد ستاره‌ها در هر سطر از بالا به پایین کاهش می‌یابد.
- برخی برنامه‌ها نسخه معکوس مثلث را تولید می‌کنند که تعداد ستاره‌ها از بالا به پایین افزایش می‌یابد.
- خروجی درست مثلثی راست‌چین است که ارتفاع آن کاهش می‌یابد.
- تغییرات مسئله:
  - عدد ورودی به جای ورودی کاربر، به عنوان آرگومان تابع ارسال شود.
  - کاراکتر ستاره با کاراکتری که کاربر مشخص کرده، جایگزین شود.

2. توابع

- تعریف توابع با استفاده از کلمه کلیدی def، نام تابع، پرانتزها و علامت :انجام می‌شود.
  - کد داخل تابع به صورت تورفتگی نوشته می‌شود.
  - توابع می‌توانند ورودی‌هایی به نام آرگومان یا پارامتر دریافت کنند.
    - مثال: def function_name(input1, input2): که input1 و input2 پارامترها هستند.
  - بازگشت مقدار با استفاده از کلمه کلیدی returnانجام می‌شود.
    - مثال: return input1 + input2 جمع دو مقدار را برمی‌گرداند.
  - متغیرهای داخل تابع به صورت محلی تعریف می‌شوند و در خارج از تابع قابل دسترسی نیستند.
  - توابع می‌توانند چندین مقدار بازگردانند.
    - مثال: return a, b, c یک تاپل شامل a، b و c بازمی‌گرداند.
  - استفاده از return باعث خروج فوری از تابع می‌شود.
- **مقادیر پیش‌فرض پارامترها**:
  - می‌توان برای پارامترها مقدار پیش‌فرض تعریف کرد.
    - مثال: def function_name(input1, input2=10):. اگر input2 هنگام فراخوانی مقدار نداشته باشد، مقدار آن ۱۰ است.
  - پارامترهای پیش‌فرض باید در انتهای لیست پارامترها قرار گیرند.
- **آرگومان‌های موقعیتی و کلیدی**:
  - آرگومان‌های موقعیتی بر اساس ترتیبشان تخصیص داده می‌شوند.
    - مثال: function_name(1, 2, 3) که ۱ به پارامتر اول، ۲ به پارامتر دوم و ۳ به پارامتر سوم تخصیص می‌یابد.
  - آرگومان‌های کلیدی با نام پارامتر مشخص می‌شوند.
    - مثال: function_name(name3=1, name1=2, name2=3).

- آرگومان‌های موقعیتی باید قبل از آرگومان‌های کلیدی قرار گیرند.
- **پذیرش تعداد متغیر از آرگومان‌ها**:
  - با استفاده از *args و **kwargs:
    - *args آرگومان‌های موقعیتی اضافی را در یک تاپل جمع می‌کند.
    - **kwargs آرگومان‌های کلیدی اضافی را در یک دیکشنری جمع می‌کند.
    - مثال: def function_name(name1, name2, *args, **kwargs):.
    - *args باید قبل از **kwargs در لیست پارامترها بیاید.
    - هنگام فراخوانی تابع، * قبل از یک دنباله و ** قبل از دیکشنری آنها را به آرگومان‌های موقعیتی و کلیدی تبدیل می‌کند.
- توابع می‌توانند توابع دیگر را فراخوانی کنند.
- یک تابع می‌تواند به عنوان آرگومان به تابع دیگری ارسال شود.

3. مستندسازی و کامنت‌گذاری

- کامنت‌های تک‌خطی با استفاده از نماد #ایجاد می‌شوند.
  - این کامنت‌ها توسط مفسر نادیده گرفته می‌شوند.
  - برای اضافه کردن توضیحات به کد استفاده می‌شوند.
- رشته‌های چندخطی که در سه کوتیشن قرار می‌گیرند می‌توانند به عنوان کامنت استفاده شوند، اما این روش استاندارد نیست.
- **Docstrings**:
  - برای مستندسازی توابع استفاده می‌شوند.
  - در ابتدای تعریف تابع قرار می‌گیرند.
  - با استفاده از ویژگی __doc__ قابل دسترسی هستند.
  - شامل توضیحات آرگومان‌ها و مقادیر بازگشتی هستند.
    - مثال: "پارامترها: name1: نام اول"
  - استاندارد نوشتن Docstrings توسط **PEP 257**تعریف شده است.
    - **PEP** به معنی **Python Enhancement Proposal** است که نحوه تغییرات و بهبود زبان پایتون را توضیح می‌دهد.

4. ماژول‌ها

- ماژول‌ها مجموعه‌ای از توابع، متغیرها و کلاس‌ها هستند که امکانات اضافی ارائه می‌دهند.
- وارد کردن ماژول‌ها با استفاده از دستور importانجام می‌شود.
  - مثال: import datetime.

- می‌توان آیتم‌های خاصی را از ماژول وارد کرد: from module import item.
  - مثال: from math import pi.
- نمونه‌هایی از ماژول‌های داخلی:
  - **ماژول datetime**:
    - برای کار با تاریخ و زمان استفاده می‌شود.
    - می‌توان تاریخ و زمان فعلی را دریافت کرد و تفاوت بین دو تاریخ را محاسبه کرد.
  - **ماژول math**:
    - توابع ریاضی پیشرفته و ثابت‌ها را ارائه می‌دهد.
    - مثال: math.pi مقدار عدد پی را برمی‌گرداند.
  - **ماژول os**:
    - دسترسی به امکانات سیستم‌عامل مانند مدیریت مسیرها را فراهم می‌کند.
    - مثال: os.getcwd() برای دریافت مسیر کاری فعلی.
    - os.mkdir() برای ایجاد دایرکتوری‌ها استفاده می‌شود.

5. کتابخانه‌های شخص ثالث

- نصب کتابخانه‌ها با استفاده از pip یا pip3:
  - مثال: pip3 install <package_name>.
- PyPI (Python Package Index) مخزن نرم‌افزارهای پایتون است.

6. سبک کدنویسی

- رعایت سبک کدنویسی و استفاده از نام‌گذاری استاندارد اهمیت دارد.
- استفاده از منابع آنلاین مانند مستندات رسمی و وبسایت‌هایی مانند geeksforgeeks.org توصیه می‌شود.

7. اشکال‌زدایی

- بررسی پیام‌های خطا:
  - "TypeError: missing 1 required positional argument" نشان‌دهنده تعداد ناکافی آرگومان‌های تابع است.
  - "ModuleNotFoundError" نشان‌دهنده عدم دسترسی به ماژول خواسته‌شده است.