Programmer's Guide: Exponentials Simulation

- This guide aims to explain the provided code that demonstrates linear and exponential growth using Python's Matplotlib library. It breaks down the code step by step, providing explanations and context for each section.

1. **Importing the Required Libraries**:

```
import matplotlib.pyplot as plt
```

- This line imports the matplotlib.pyplot module as plt, allowing you to use its functions to create plots and visualizations.

2. Defining Growth Functions:

2.1 linear_growth Function:

```
def linear_growth(k, x):
  blocks_stacked = [i * k for i in range(1, x + 1)]
  return blocks_stacked
```

- This function models linear growth. It takes two arguments, k and x, where k represents the growth rate and x is the number of growth stages. It calculates the number of blocks stacked at each stage using the formula i * k, where i is the stage number.

2.2 exponential_growth Function:

```
def exponential_growth(x, base):
 blocks_stacked = [base ** i for i in range(x)]
 return blocks_stacked
```

- This function models exponential growth. It takes two arguments, x (number of growth stages) and base (the exponential base). It calculates the number of blocks stacked at each stage using the formula base ** i, where i is the stage number.

3. Defining the main Function:

```
def main():
    while True:
  # ... (rest of the code)
```

- The main function contains the main logic of the program. It uses a continuous loop to allow the user to repeat the process of entering growth parameters, displaying growth stages, plotting graphs, and choosing whether to run the program again.

4. Input and Display of Linear Growth:

```
stages = [
   "stage 1: block 1",
   # ... (other stages)
 "stage 10: block 10"
]
```

- This list contains strings representing the stages of linear growth. Each stage is labeled with the stage number and the number of blocks stacked.

5. Input and Display of Exponential Growth:

```
choice = input("Enter '2x', 'x2', or '3x': ")
```

- The user is prompted to enter a choice representing the type of exponential growth. It can be 2x, x2, or 3x.

6. Plotting Growth Data:

```
plt.plot(x_values, blocks_stacked_linear, label=f"Linear Growth (k = {k})")
plt.plot(x_values, blocks_stacked_exponential, label=f"Exponential Growth ({choice})")
plt.xlabel('Step')
plt.ylabel('Number of Blocks')
plt.legend()
plt.grid(True)
plt.title('Linear vs Exponential Growth')

plt.show()
```

- These lines of code create a line plot using matplotlib. It plots the number of blocks stacked against growth stages for both linear and exponential growth. The labels, legends, axes labels, grid, and title are set for the visualization.

7. Repeat Option:

```
repeat = input("Do you want to run again? (y/n): ")

if repeat.lower() != 'y':

    break
```

- After displaying the graph, the user is asked if they want to run the program again. If the user's input is not 'y', the loop breaks, and the program terminates.

8. Main Execution Block:

```
if __name__ == "__main__":

  main()
```

- This block ensures that the main function is executed only if the script is run directly, not when imported as a module into another script. It's a common practice in Python to separate the executable code from module code.