

Chapter 1

[illegible]


```

big_verse()
go_team_go()

def go_team_go():
    print("Go, team, go!")
    print("You can do it.")

def big_verse():
    go_team_go()
    print("You're the best,")
    print("In the West.")
    go_team_go()
    print()

```

11. # This program prints a pattern of starry figures.

```

def main():
    print_figure1()
    print()
    print_figure2()
    print()
    print_figure3()

def print_figure1():
    print_horizontal_bar()
    print_x()

def print_figure2():
    print_horizontal_bar()
    print_x()
    print_horizontal_bar()

def print_figure3():
    print("  *")
    print(" * ")
    print(" * ")
    print_figure1()

def print_horizontal_bar():
    print("*****")
    print("*****")

def print_x():
    print(" * *")
    print("  * ")
    print(" * *")

```

12. # This program prints a pattern of figures such as eggs and stop signs.

The code uses functions for structure and to remove redundancy.

```

def main():
    egg()
    print()
    egg()
    line()
    print()
    stop_sign()
    line()
    print()

def egg():
    egg_top()
    egg_bottom()

def egg_top():
    print("  _____")

```

```

print(" /      \\\")
print("/        \\\")

def egg_bottom():
    print("\\      /")
    print(" \\_____/")

def stop_sign():
    egg_top()
    print("|  STOP  |")
    egg_bottom()

def line():
    print("+-----+")

```

Chapter 2

```

1. number = 1
   increment = 3
   for i in range(1, 11):
       print(str(number) + " ", end="")
       number = number + increment
       increment = increment + 2
   print() # to end the line

2. n1 = 1
   n2 = 1
   print(str(n1) + " " + str(n2) + " ", end="")
   for i in range(3, 13):
       n3 = n1 + n2
       n1 = n2
       n2 = n3
       print(str(n2) + " ", end="")
   print()

3. for i in range(1, 5):
    print("*" * 5)

4. for i in range(1, 6):
    print("*" * i)

5. for i in range(1, 8):
    print(str(i) * i)

6. for i in range(1, 7):
    print("|      ", end="")
    print()
    for i in range(1, 7):
        for j in range(1, 11):
            print(j % 10, end="")
    print()

7. COUNT = 6
   INNER_COUNT = 7
   def main():
       for i in range(1, COUNT + 1):
           print("|", end="")
           print(" " * (INNER_COUNT - 1), end="")
       print()

       for i in range(1, COUNT + 1):
           for j in range(1, INNER_COUNT + 1):

```

```

        print(j % INNER_COUNT, end="")
    print()

main()

8. def print_design():
    for line in range(1, 6):
        print("-" * (-1 * line + 6), end="")
        print(str(2 * line - 1) * (2 * line - 1), end="")
        print("-" * (-1 * line + 6))

9. def main():
    for line in range(1, 7):
        print("\\" * (2 * line - 2), end="")
        print("! " * (-4 * line + 26), end="")
        print("/" * (2 * line - 2))

    main()

10. SIZE = 4
    def main():
        for line in range(1, SIZE + 1):
            print("\\" * (2 * line - 2), end="")
            print("! " * (-4 * line + (4 * SIZE + 2)), end="")
            print("/" * (2 * line - 2))

    main()

11. # Draws a resizable window figure with for loops
    # and a constant.
    COUNT = 3
    def main():
        draw_line()
        for i in range(2):
            for j in range(COUNT):
                draw_bar()
            draw_line()

    # Draws a horizontal line: +==+==+
    def draw_line():
        print("+", end="")
        print("=" * COUNT, end="")
        print("+", end="")
        print("=" * COUNT, end="")
        print("+")

    # Draws a single line of bars: |   |   |
    def draw_bar():
        print("|", end="")
        print(" " * COUNT, end="")
        print("|", end="")
        print(" " * COUNT, end="")
        print("|")

    main()

12. def main():
    for line in range(1, 6):
        print("/" * (-4 * line + 20), end="")
        print("*" * (8 * line - 8), end="")
        print("\\" * (-4 * line + 20))

    main()

```