

```

# --- GENERATOR SCRIPT (RUN THIS ONCE) ---
# This script uses libraries to WRITE the code for you.
# The OUTPUT file (minor2_project.py) will have NO libraries.

from PIL import Image
import sys

def generate_project_file(image_filename, output_filename="minor2_project.py"):
    try:
        # 1. Open the image directly (JPG/PNG works fine here)
        img = Image.open(image_filename)

        # 2. Resize it nicely for the terminal (Width 50-60 is best)
        # We maintain aspect ratio roughly
        base_width = 55
        w_percent = (base_width / float(img.size[0]))
        h_size = int((float(img.size[1]) * float(w_percent)) * 0.55) # 0.55 corrects vertical stretch
        img = img.resize((base_width, h_size), Image.Resampling.LANCZOS)
        img = img.convert("RGB") # Ensure it's RGB

        width, height = img.size
        pixels = img.load() # Get pixel data

        print(f"Generating code for {width}x{height} image...")

        # 3. Write the "Handmade" Code to a file
        with open(output_filename, "w") as out:
            out.write("# Minor II Project - ASCII Art Generator\n")
            out.write("# Logic: Conditional Mapping of Coordinates\n\n")

            out.write("def generate_ascii_art():\n")
            out.write(f"    # Dimensions: {width}x{height}\n")
            out.write(f"    height = {height}\n")
            out.write(f"    width = {width}\n\n")
            out.write("    print(\"Printing ASCII Art for Kareena Kapoor...\\n\")\n\n")

            out.write("    for row in range(height):\n")
            out.write("        line = \"\\n\"\n")
            out.write("        for col in range(width):\n")
            out.write("            char = \" \"\n")

            # --- THE LOGIC GENERATOR ---
            chars = ["@", "%", "#", "*", "+", "=", "-", ":", ".", " "]

            for y in range(height):
                out.write(f"                if row == {y}:\n")

                # Optimization: Track if we've written an 'if' for this row yet
                first_match = True

                for x in range(width):
                    r, g, b = pixels[x, y]
                    brightness = int((r + g + b) / 3)
                    char_idx = (brightness * (len(chars) - 1)) // 255
                    ascii_char = chars[char_idx]

                    # Only write code if it's NOT a space (saves file size)
                    if ascii_char.strip() != "":
                        if first_match:
                            out.write(f"                    if col == {x}:\n")
                            first_match = False
                        else:
                            out.write(f"                    elif col == {x}:\n")

                        out.write(f"                        char = \"{ascii_char}\\n\")

                        if first_match:
                            out.write("                        pass\\n")
                            first_match = False

                        out.write("                    line += char\\n")
                        out.write("                    print(line)\\n\\n")

                out.write("# Execute the function\n")
                out.write("if __name__ == \"__main__\":\n")
                out.write("    generate_ascii_art()\n\n")

            print(f"SUCCESS! Download '{output_filename}' and submit that file.")

    except Exception as e:
        print(f"Error: {e}")

```

```
# --- SETUP ---
# 1. Upload your photo as 'kareena.jpg' (or png) to Colab files
# 2. Run this block
if __name__ == "__main__":
    # CHANGE THIS to your actual image filename!
    generate_project_file("kareena.ppm")
    # If your file is jpg, change to "kareena.jpg"
```

```
Generating code for 55x37 image...
SUCCESS! Download 'minor2_project.py' and submit that file.
```