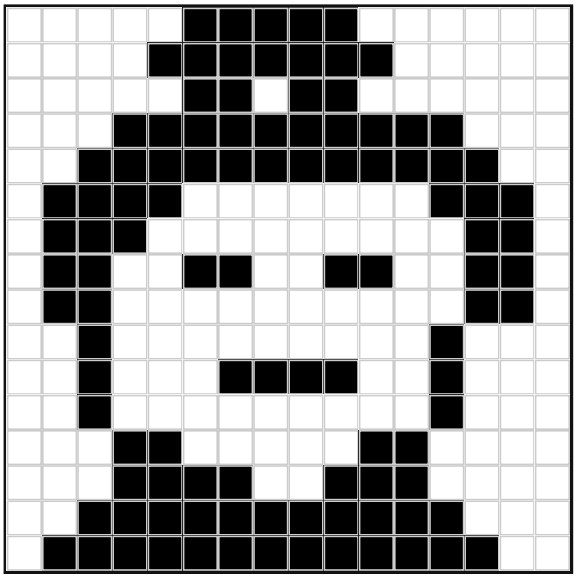
🖶 PRINT / SAVE AS PDF ⬇ DOWNLOAD PIXEL ART PNG

BINARY IMAGE REPRESENTATION — MAFALDA 16×16

Computer Science · Data Representation · Each black pixel = **1** · Each white pixel = **0**

PIXEL IMAGE



BINARY REPRESENTATION

0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0
0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

🔑 KEY

Black pixel = **1**

White pixel = **0**

Grid size: **16 × 16 = 256 pixels**

Bits per pixel: **1 bit**

Total: **256 bits = 32 bytes**

📖 HOW IT WORKS

- Each pixel is stored as a single **bit**
- A **1** means the pixel is switched *on* (black)
- A **0** means the pixel is switched *off* (white)
- The computer reads the grid **row by row**, left to right
- The image **metadata** (e.g. width, height) is also stored

🧮 CALCULATE

- Image size = width × height × bits per pixel
- 16 × 16 × 1 bit = **256 bits**
- ÷ 8 = **32 bytes**
- If 2 colours → **1 bit/pixel**
- If 4 colours → **2 bits/pixel**
- If 256 colours → **8 bits/pixel**

✏ TASKS

1. How many pixels are in this image?

2. How many *bits* are needed to store it?

3. Convert that to *bytes*.

4. Write out the binary for Row 4 (the hair row).

5. If we added grey (3 colours), how many bits per pixel would we need?

6. If the image was 32×32 (same 1-bit colour), how many bytes?

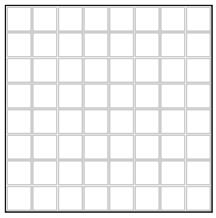
7. What is the binary value of the decimal number 10?

★ Extension: Explain what *metadata* a bitmap image file might store alongside the pixel data.

✏ STUDENT ACTIVITY — DESIGN YOUR OWN 8×8 SPRITE

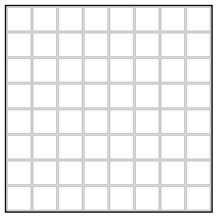
Shade squares in the left grid to create your sprite. Then fill in the binary grid on the right. Remember: black = 1, white = 0.

My Pixel Art:



→

Binary:



Calculate:
Width × Height × Bits/pixel
= 8 × 8 × 1 = **64 bits**
= **8 bytes**