

## UTF-8 Encoding

From the UTF-8 encoding scheme given below, write the Emoji “Face with tears of joy” 😂 (code point 128514 = 0x1f602) in UTF-8.

### Task

Determine the UTF-8 encoding of the emoji 😂 (code point 128514 = 0x1f602) and provide your answer in hexadecimal format (e.g., 0x...).

Store your answer as a string in the variable `ans`.

### Sample submission

Declare a variable `ans` with the correct answer:

```
ans = "0xa0b1c2def" # Your answer here as a string
```

### Notes

The Reference of UTF-8 encoding is provided in the next page.

## Reference

### UTF-8 Encoding Scheme

**0000 0000 ... 0000 007F(7)** (\$FFFF FF80; 0..127) USC-4(32 bits cell, Little-endian)

31 30 29 28 27 26 25 24   23 22 21 20 19 18 17 16   15 14 13 12 11 10 9 8   7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0	0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 A7 A6 A5 A4 A3 A2 A1
0 A7 A6 A5 A4 A3 A2 A1		
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	UTF-8 Byte 1

**0000 0080 ... 0000 07FF(11)** (\$FFFF F800; 128..2 047) USC-4(32 bits cell, Little-endian)

31 30 29 28 27 26 25 24   23 22 21 20 19 18 17 16   15 14 13 12 11 10 9 8   7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0	0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   B5 B4 B3 B2 B1 A6 A5 A4 A3 A2 A1
1 1 0 B5 B4 B3 B2 B1	1 0 A6 A5 A4 A3 A2 A1	
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	UTF-8 Byte 2

**0000 0800 ... 0000 FFFF(16)** (\$FFFF 0000; 2 048..65 535) USC-4(32 bits cell, Little-endian)

31 30 29 28 27 26 25 24   23 22 21 20 19 18 17 16   15 14 13 12 11 10 9 8   7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0	0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   C4 C3 C2 C1 B6 B5 B4 B3 B2 B1 A6 A5 A4 A3 A2 A1
1 1 1 0 C4 C3 C2 C1	1 0 B6 B5 B4 B3 B2 B1	1 0 A6 A5 A4 A3 A2 A1
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	UTF-8 Byte 3

**0001 0000 ... 001F FFFF(21)** (\$FFE0 0000; 65 536..2 097 151) USC-4(32 bits cell, Little-endian)

31 30 29 28 27 26 25 24   23 22 21 20 19 18 17 16   15 14 13 12 11 10 9 8   7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0	0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   D3 D2 D1 C6 C5 C4 C3 C2 C1 B6 B5 B4 B3 B2 B1 A6 A5 A4 A3 A2 A1
1 1 1 1 0 D3 D2 D1	1 0 C6 C5 C4 C3 C2 C1	1 0 B6 B5 B4 B3 B2 B1
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	UTF-8 Byte 4

**0020 0000 ... 03FF FFFF(26)** (\$FC00 0000; 2 097 152..67 108 863) USC-4(32 bits cell, Little-endian)

31 30 29 28 27 26 25 24   23 22 21 20 19 18 17 16   15 14 13 12 11 10 9 8   7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0	0 0 0 0 0 0 0 0   E2 E1 D6 D5 D4 D3 D2 D1 C6 C5 C4 C3 C2 C1 B6 B5 B4 B3 B2 B1 A6 A5 A4 A3 A2 A1
1 1 1 1 0 E2 E1	1 0 D6 D5 D4 D3 D2 D1	1 0 C6 C5 C4 C3 C2 C1
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	UTF-8 Byte 3
		1 0 B6 B5 B4 B3 B2 B1
		1 0 A6 A5 A4 A3 A2 A1
		7 6 5 4 3 2 1 0

**0400 0000 ... 7FFF FFFF(31)** (\$8000 0000; 67 108 864..2 147 483 647) USC-4(32 bits cell, Little-endian)

31 30 29 28 27 26 25 24   23 22 21 20 19 18 17 16   15 14 13 12 11 10 9 8   7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0   7 6 5 4 3 2 1 0	0 F1 E6 E5 E4 E3 E2 E1 D6 D5 D4 D3 D2 D1 C6 C5 C4 C3 C2 C1 B6 B5 B4 B3 B2 B1 A6 A5 A4 A3 A2 A1
1 1 1 1 1 0 F1	1 0 E6 E5 E4 E3 E2 E1	1 0 D6 D5 D4 D3 D2 D1
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	UTF-8 Byte 3
		1 0 C6 C5 C4 C3 C2 C1
		1 0 B6 B5 B4 B3 B2 B1
		1 0 A6 A5 A4 A3 A2 A1
		7 6 5 4 3 2 1 0