**Q.1** **Number Conversions**

Please show the steps involved.

1. Convert binary 100101111010 to octal and to hexadecimal.

Ans: Octal: 100 = 4, 101 = 5, 111 = 7, 010 = 2, 1001011110102 = **45728**

Hexadecimal: 1001 = 9, 0111 = 7, 1010 = A and thus, 1001011110102 = **97A16**

1. Convert hexadecimal 3A7D to binary.

Ans: 3 = 0011, A = 1010, 7 = 0111, D = 1101, thus, 3A7D16 = **111010011111012**

1. Convert hexadecimal 765F to octal. [***Hint***: First convert 765F to binary, then convert that binary number to octal.]

Ans: Hexadecimal to Binary: 7 = 0111, 6 = 0110, 5 = 0101, F = 1111 => 765F16 = 1110110010111112

Binary to Octal: 111 = 7, 011 = 3, 001 = 1, 011 = 3, 111 = 7, thus, 765F16 = **731378** in Octal

1. Convert binary 1011110 to decimal.

Ans: 10111102 = 2^6 + 2^4 + 2^3 + 2^2 + 2^1 = **9410**

1. Convert octal 426 to decimal.

Ans: 4268 = 4\*8^2 + 2\*8^1 + 6\*8^0 = **27810**

1. Convert hexadecimal FFFF to decimal.

Ans: FFFF16 = 15\*16^3 + 15\*16^2 + 15\*16^1 + 15\*16^0 = **6553510**

1. Convert decimal 299 to binary, to octal and to hexadecimal.

Ans: Binary: 29910 = 256 + 32 + 8 + 2 + 1 = **1001010112**

Octal: 100 = 4, 101 = 5, 011 = 3, => **4538**

Hexadecimal: 0001 = 1, 0010 = 2, 1011 = B, => **12B16**

29910 = **1001010112** = **4538** = **12B16**

1. Convert hexadecimal FACE to binary.

Ans: F = 1111, A = 1010, C = 1100, E = 1110, FACE16 = **11111010110011102**

**Q.2** **IEEE 754**

Represent 14.75 in its IEEE 754 single precision floating point representation. Put your answer in hexadecimal.

Ans: Converting 14.75 into binary: 14 = 1110 and 0.75 = .11 => 14.75 = 1.11011 \* 2^3

sign = positive = 0, biased exponent = 127+3 = 10000010, mantissa = 11011 and fill 0 til 32bits

Therefore, the IEEE-754 single point representation of 14.75: **0 10000010 11011000000000000000000**

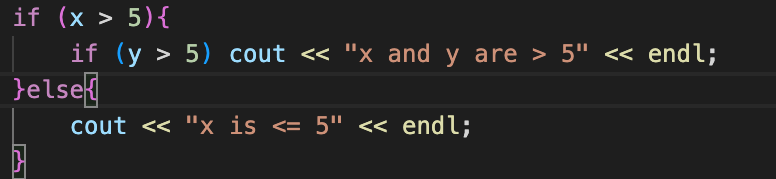
In hexadecimal: (0100=4 0001=1 0110=6 1100=C) => **416C0000**

**Q.3** **Dangling-else Problem**

Graphical user interface, text, application, email

Description automatically generated

Q3 Ans:



**Q.4** **Dangling-else Problem** (cont.)

Graphical user interface, text, application

Description automatically generated

Q4a. When x == 9 and y == 11

Output: \*\*\*\*\*

$$$$$

When x == 11 and y == 9

Output: $$$$$

b. When x == 9 and y == 11

Output: \*\*\*\*\*

When x == 11 and y == 9

Output: #####

$$$$$

**Q.5** **Dangling-else Problem** (cont.)

(Another Dangling-else Problem) Based on the dangling-else discussion in Q.3, modify the following code to produce the output shown. Use proper indentation techniques. You must not make any additional changes other than inserting braces. We eliminated the indentation from the following code to make the problem more challenging. [Note: It’s possible that no modification is necessary.]

Graphical user interface, application

Description automatically generated

Q5:

a.

一張含有 文字, 螢幕, 黑色 的圖片

自動產生的描述

b.

一張含有 文字, 螢幕 的圖片

自動產生的描述 or 一張含有 文字 的圖片

自動產生的描述

c.

一張含有 文字, 螢幕 的圖片

自動產生的描述

d.

一張含有 文字 的圖片

自動產生的描述

**Q.6**

Table

Description automatically generated

一張含有 文字, 監視器, 室內, 螢幕 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

**Q.7**

Graphical user interface, text, application

Description automatically generated

一張含有 文字 的圖片

自動產生的描述