

Computer Fundamentals & Programming – Assignment 01

Answers

Topic: Number Systems, Conversions, and Algorithm Design

A1. Convert from Base-2 to Base-10

$$\begin{aligned} 1. \quad 1011_2 &\rightarrow ?_{10} \\ &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 8 + 0 + 2 + 1 \\ &= 11_{10} \end{aligned}$$

$$\begin{aligned} 2. \quad 100101.01_2 &\rightarrow ?_{10} \\ \text{Integer part:} \\ 100101_2 &= 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 32 + 4 + 1 = 37 \end{aligned}$$

$$\begin{aligned} \text{Fraction part:} \\ 0.01_2 &= 0 \times 2^{-1} + 1 \times 2^{-2} = 0.25 \end{aligned}$$

$$\begin{aligned} \text{Final answer:} \\ 37.25_{10} \end{aligned}$$

$$\begin{aligned} 3. \quad 101000.011_2 &\rightarrow ?_{10} \\ \text{Integer part:} \\ 101000_2 &= 1 \times 2^5 + 1 \times 2^3 = 32 + 8 = 40 \end{aligned}$$

$$\begin{aligned} \text{Fraction part:} \\ 0.011_2 &= 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ &= 0.25 + 0.125 = 0.375 \end{aligned}$$

$$\begin{aligned} \text{Final answer:} \\ 40.375_{10} \end{aligned}$$

A2. Convert from Base-10 to Base-2

$$\begin{aligned} 1. \quad 18.25_{10} &\rightarrow ?_2 \\ 18_{10} &\rightarrow 10010_2 \\ 0.25_{10} &\rightarrow 0.01_2 \end{aligned}$$

$$\begin{aligned} \text{Final answer:} \\ 10010.01_2 \end{aligned}$$

$$2. \quad 13.5_{10} \rightarrow ?_2$$

$$13_{10} \rightarrow 1101_2$$

$$0.5_{10} \rightarrow 0.1_2$$

Final answer:

$$1101.1_2$$

$$3. \quad 40.375_{10} \rightarrow ?_2$$

$$40_{10} \rightarrow 101000_2$$

$$0.375_{10} \rightarrow 0.011_2$$

Final answer:

$$101000.011_2$$

A3. Non-Decimal to Non-Decimal Conversion

(All conversions go through base-10)

$$1. \quad 24.13_5 \rightarrow ?_3$$

Step 1: Base-5 to Base-10

$$24.13_5 = 2 \times 5^1 + 4 \times 5^0 + 1 \times 5^{-1} + 3 \times 5^{-2}$$

$$= 10 + 4 + 0.2 + 0.12$$

$$= 14.32_{10}$$

Step 2: Base-10 to Base-3

$$14_{10} = 112_3$$

0.32_{10} is repeating in base-3

Final answer:

$$112.(0221)_3$$

$$2. \quad 1011.101_2 \rightarrow ?_8$$

Step 1: Base-2 to Base-10

$$1011_2 = 11_{10}$$

$$0.101_2 = 0.5 + 0.125 = 0.625$$

$$\text{Total} = 11.625_{10}$$

Step 2: Base-10 to Base-8

$$11_{10} \rightarrow 13_8$$

$$0.625_{10} \rightarrow 0.5_8$$

Final answer:

$$13.5_8$$

3. $A.C_{16} \rightarrow ?_8$

Step 1: Base-16 to Base-10

$$A_{16} = 10$$

$$C_{16} = 12$$

$$A.C_{16} = 10 + 12 \times 16^{-1}$$

$$= 10 + 0.75$$

$$= 10.75_{10}$$

Step 2: Base-10 to Base-8

$$10_{10} \rightarrow 12_8$$

$$0.75_{10} \rightarrow 0.6_8$$

Final answer:

$$12.6_8$$

B1. Binary Addition

$$1011_2 + 1101_2$$

$$1011$$

$$+ 1101$$

$$11000_2$$

Explanation:

- $1 + 1 = 0$, carry 1
- $1 + 0 + \text{carry} = 0$, carry 1
- $0 + 1 + \text{carry} = 0$, carry 1
- $1 + 1 + \text{carry} = 1$, carry 1
- Final carry = 1

Final Result:

$$11000_2$$

B2. Binary Subtraction

$$11010_2 - 1011_2$$

11010

- 01011

1111₂

Explanation:

- Subtraction is done using borrowing in base-2
- Required borrows are applied from left to right
- All digits resolve correctly after borrowing

Final Result:

1111₂

B3. Octal Addition

$$157_8 + 24_8$$

157

+ 24

203₈

Explanation:

- $7 + 4 = 11_8 \rightarrow$ write 3, carry 1
- $5 + 2 + \text{carry} = 10_8 \rightarrow$ write 0, carry 1
- $1 + \text{carry} = 2$

Final Result:

203₈

B4. Base-9 Subtraction

$$725_9 - 348_9$$

725

- 348

366₉

Explanation:

- $5 - 8 \rightarrow$ borrow 1 (equal to 9), $14 - 8 = 6$
- $1 - 4 \rightarrow$ borrow 1, $10 - 4 = 6$
- $6 - 3 = 3$

Final Result:

366₉

B5. Hexadecimal Addition

$$A3_{16} + 1C_{16}$$

A3

+ 1C

BF₁₆

Explanation:

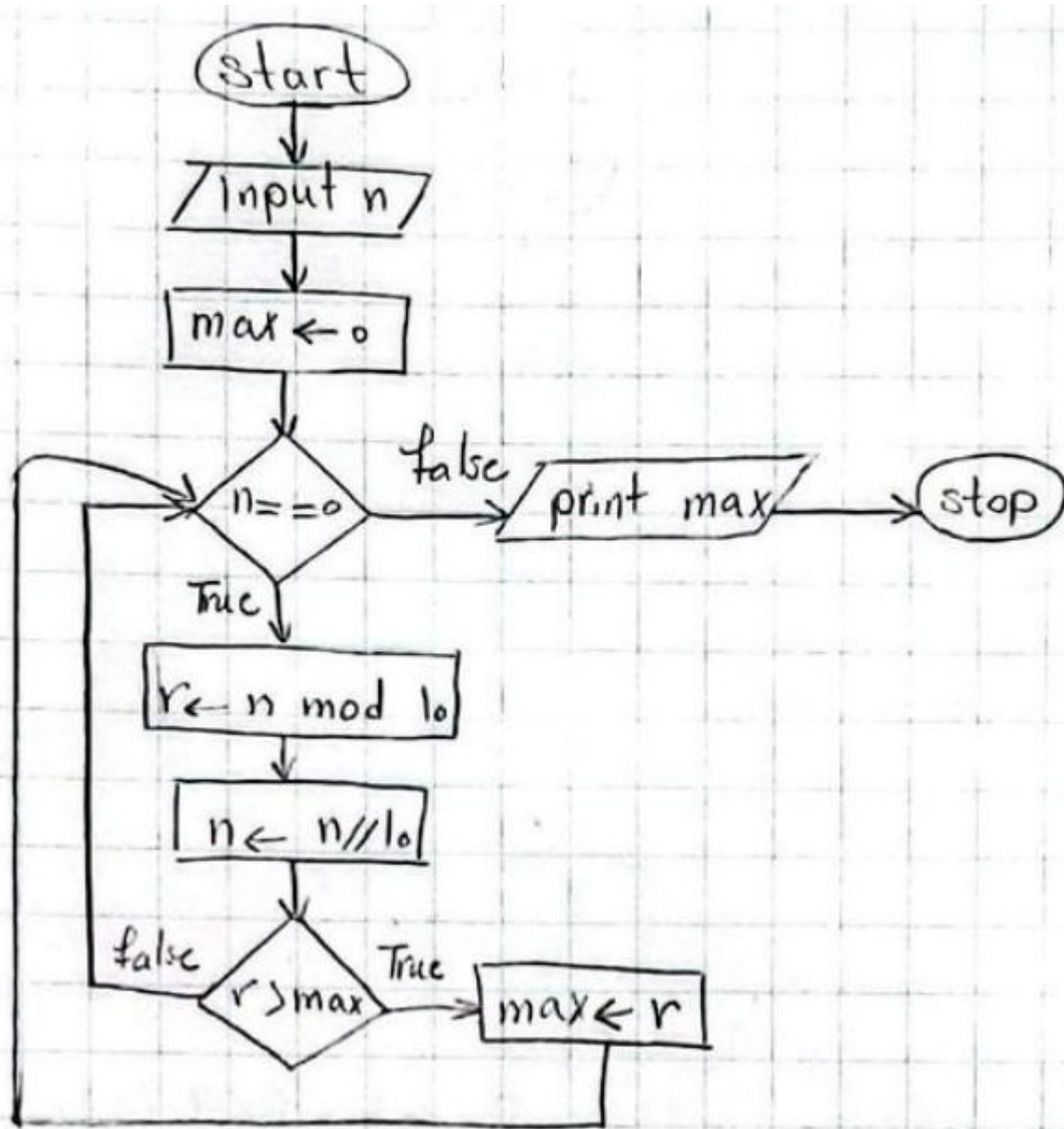
- $3 + C(12) = F(15)$, no carry
- $A(10) + 1 = B(11)$

Final Result:

BF₁₆

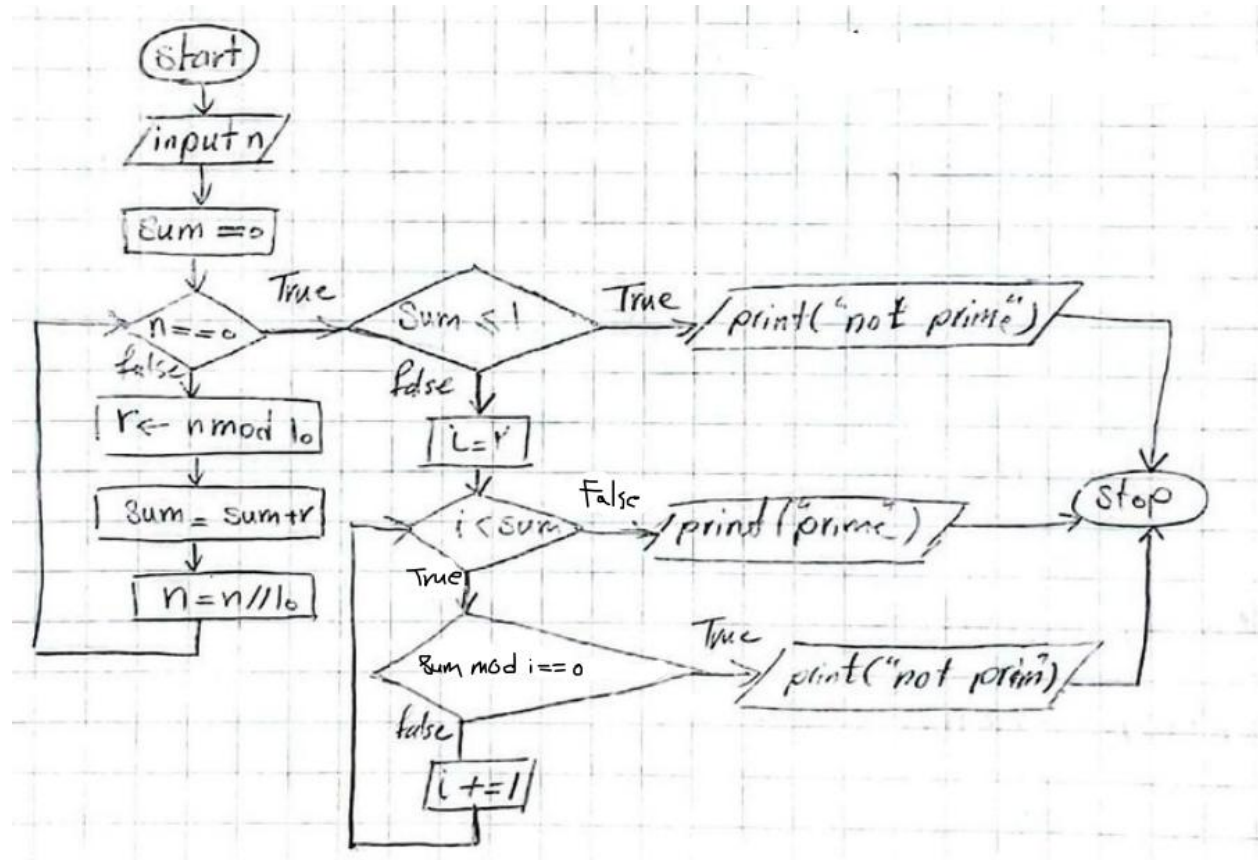
C1. Largest Digit of a Number

Design an algorithm and draw its flowchart to read an integer from the user and print its largest digit.



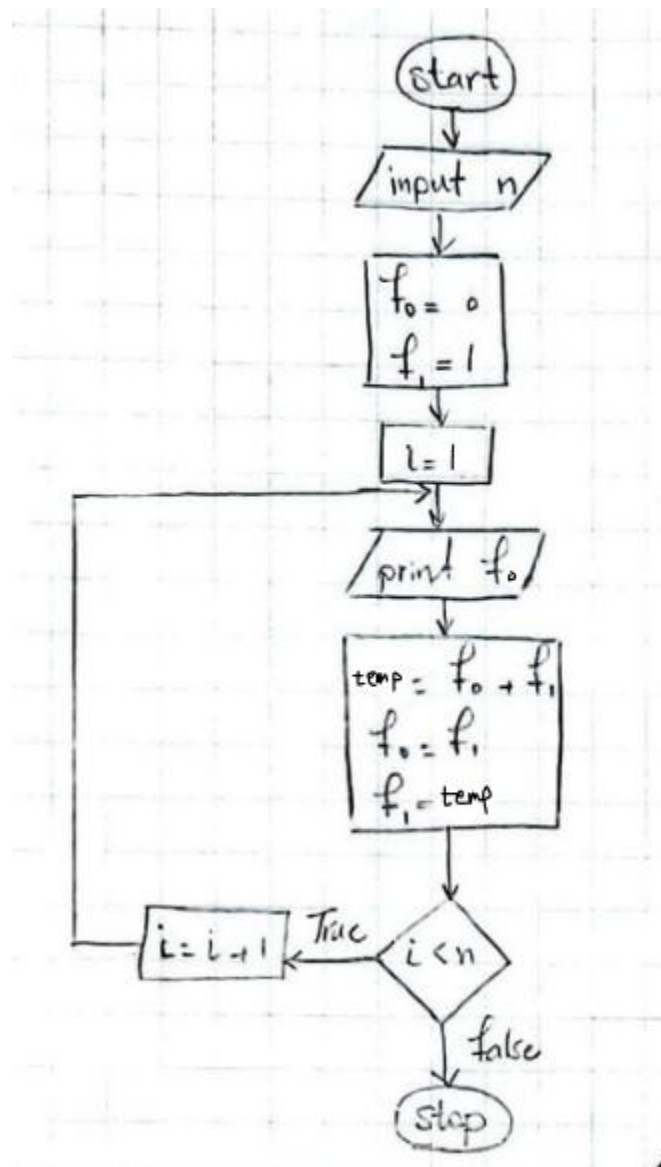
C2. Checking Whether the Sum of Digits Is Prime or Composite

Design an algorithm and draw its flowchart to read a number, compute the sum of its digits, and determine whether this sum is prime or composite.



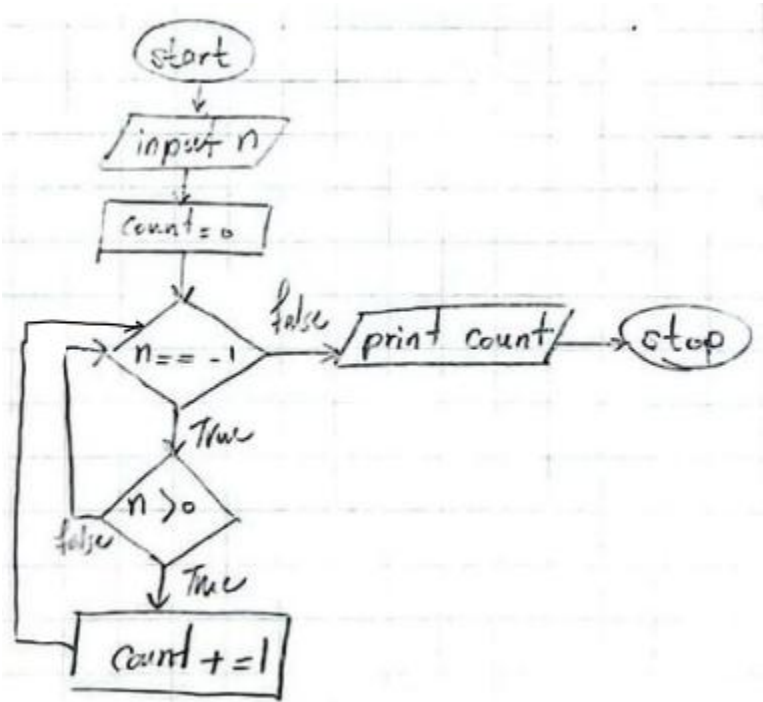
C3. Fibonacci Sequence up to the n-th Term

Design an algorithm and draw its flowchart to read an integer n and print the Fibonacci sequence up to the n-th term.



C4. Counting Positive Numbers Until the User Enters -1

Design an algorithm and draw its flowchart that repeatedly reads integers until -1 is entered, and prints how many of the entered numbers were positive.



C5. Sum of All Prime Numbers Up to n

Design an algorithm and draw its flowchart to compute the sum of all prime numbers from 1 to n.

