

Week 5 Exercise Sheet – Worked Solutions

Identifying Variables · IPO Charts · Flowcharts · Pseudocode

For each question the solution is presented in four parts: identified variables, an IPO chart, a flowchart, and numbered pseudocode. Questions 1–15.

Pseudocode Conventions

BEGIN / END - marks the start and end of the algorithm.

INPUT - reads a value from the user.

DISPLAY - outputs a value to the user.

IF ... THEN ... ELSE ... ENDIF - conditional branch.

WHILE ... ENDWHILE - pre-checked loop.

LOOP ... EXIT LOOP ... ENDLLOOP - mid-exit loop.

MOD - remainder after division ($7 \text{ MOD } 2 = 1$).

RANDOM(a,b) - random integer between a and b inclusive.

Question 1 – Leap Year

Find if the user-input year is a leap year or not.

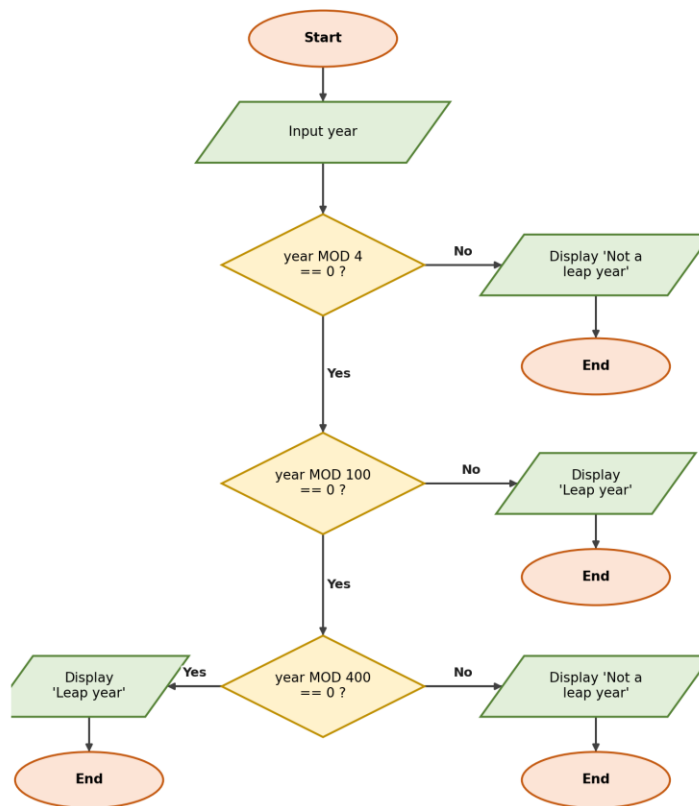
Identified Variables

Variable	Type	Description
year	<i>Integer</i>	The year entered by the user

IPO Chart

Input	Process	Output
year	1. Check $\text{year} \bmod 4 == 0$ 2. If yes, check $\text{year} \bmod 100 == 0$ 3. If yes, check $\text{year} \bmod 400 == 0$	"Leap year" or "Not a leap year"

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT year
3    IF year MOD 4 == 0 THEN
4      IF year MOD 100 == 0 THEN
5        IF year MOD 400 == 0 THEN
6          DISPLAY "Leap year"
7        ELSE
8          DISPLAY "Not a leap year"
9        ENDIF
10     ELSE
11       DISPLAY "Leap year"
12     ENDIF
13   ELSE
14     DISPLAY "Not a leap year"
15   ENDIF

```


Question 2 – Largest of Three with Equality Check

Find and display the largest of three input numbers A, B, and C. If any two numbers are equal, display "Two numbers are equal"; if all three are equal, display "All inputs are equal".

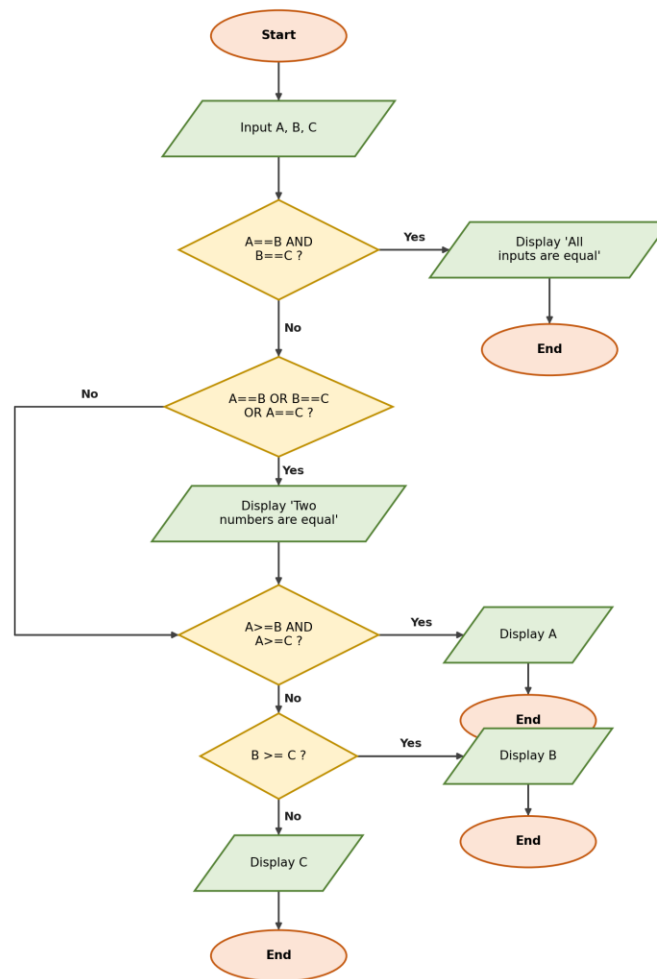
Identified Variables

Variable	Type	Description
A, B, C	<i>Number</i>	The three input numbers

IPO Chart

Input	Process	Output
A, B, C	1. Check if all three are equal 2. Check if any two are equal 3. Find the largest of A, B, C	Equality message (if any); largest value

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT A, B, C
3    IF A == B AND B == C THEN
4      DISPLAY "All inputs are equal"
5    ELSE
6      IF A == B OR B == C OR A == C THEN
7        DISPLAY "Two numbers are equal"
8      ENDIF
9      IF A >= B AND A >= C THEN
10       DISPLAY A
11     ELSE
12       IF B >= C THEN
13         DISPLAY B
14       ELSE
15         DISPLAY C
16       ENDIF

```

17	ENDIF
18	ENDIF
19	END

Question 3 – Area of a Right-Angled Triangle

Calculate the area of a right-angled triangle using base and height.

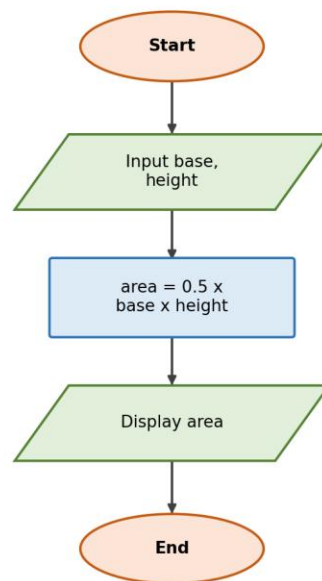
Identified Variables

Variable	Type	Description
base	Number	The base length of the triangle
height	Number	The perpendicular height of the triangle
area	Number	Calculated area = $0.5 \times \text{base} \times \text{height}$

IPO Chart

Input	Process	Output
base, height	$\text{area} = 0.5 \times \text{base} \times \text{height}$	area

Flowchart



Pseudocode

```

1  BEGIN
2      INPUT base, height
3      area = 0.5 * base * height
4      DISPLAY area
  
```


Question 4 – Yearly Tax (User Provides Rate Name)

Calculate yearly tax for an employee. The user inputs their yearly income and the name of their tax rate band; the program outputs the yearly tax payment.

Bands: Personal Allowance $\leq$ £12,570 (0%); Basic Rate £12,571–£50,270 (20%); Higher Rate £50,271–£125,140 (40%); Additional Rate $>$ £125,140 (45%).

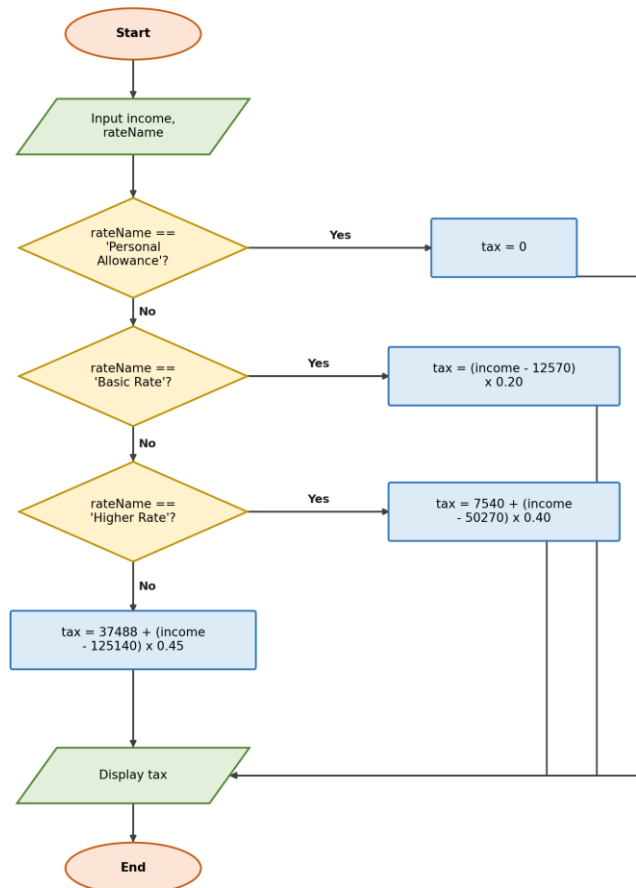
Identified Variables

Variable	Type	Description
income	Number	Gross yearly income entered by the user
rateName	String	Name of the tax band (e.g. "Basic Rate")
tax	Number	Calculated yearly tax payment

IPO Chart

Input	Process	Output
income, rateName	Apply the formula for the named band: • Personal Allowance: $\text{tax} = 0$ • Basic Rate: $\text{tax} = (\text{income} - 12570) \times 0.20$ • Higher Rate: $\text{tax} = 7,540 + (\text{income} - 50,270) \times 0.40$ • Additional Rate: $\text{tax} = 37,488 + (\text{income} - 125,140) \times 0.45$	tax (yearly)

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT income, rateName
3    IF rateName == "Personal Allowance" THEN
4      tax = 0
5    ELSE
6      IF rateName == "Basic Rate" THEN
7        tax = (income - 12570) * 0.20
8      ELSE
9        IF rateName == "Higher Rate" THEN
10         tax = 7540 + (income - 50270) * 0.40
11       ELSE
12         tax = 37488 + (income - 125140) * 0.45
13       ENDIF
14     ENDIF
15   ENDIF
16   DISPLAY tax

```


Question 5 – Yearly Tax (Auto Band Detection)

Same as Question 4, but the user only inputs their yearly income. The program automatically determines the tax band and outputs yearly, monthly, and weekly tax payments.

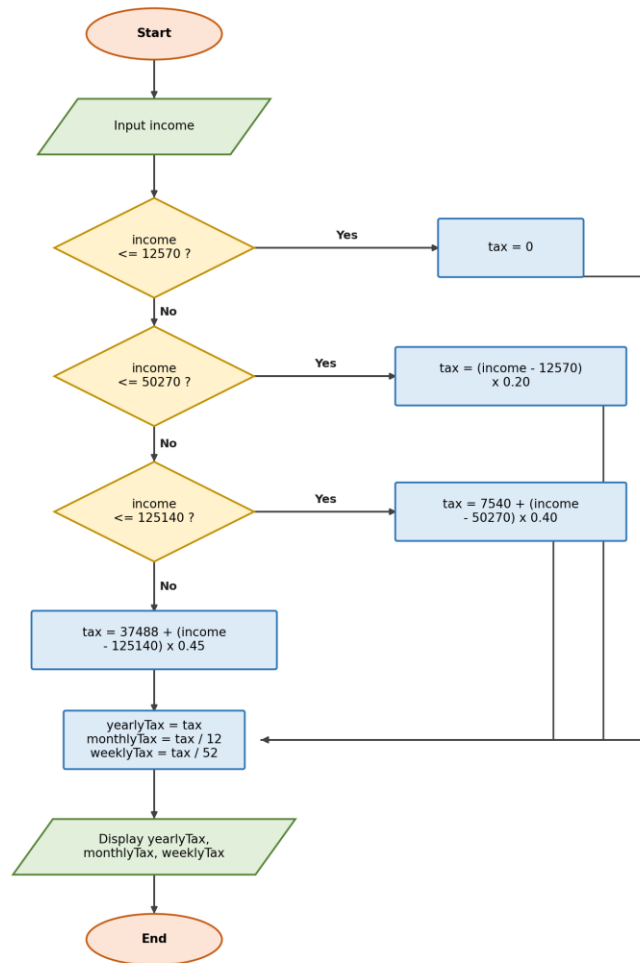
Identified Variables

Variable	Type	Description
income	<i>Number</i>	Gross yearly income entered by the user
tax	<i>Number</i>	Base yearly tax calculated from income
yearlyTax	<i>Number</i>	Same as tax
monthlyTax	<i>Number</i>	$\text{tax} \div 12$
weeklyTax	<i>Number</i>	$\text{tax} \div 52$

IPO Chart

Input	Process	Output
income	Determine band from income; calculate progressive tax; $\text{yearlyTax} = \text{tax}$ $\text{monthlyTax} = \text{tax} / 12$ $\text{weeklyTax} = \text{tax} / 52$	yearlyTax, monthlyTax, weeklyTax

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT income
3    IF income <= 12570 THEN
4      tax = 0
5    ELSE
6      IF income <= 50270 THEN
7        tax = (income - 12570) * 0.20
8      ELSE
9        IF income <= 125140 THEN
10       tax = 7540 + (income - 50270) * 0.40
11     ELSE
12       tax = 37488 + (income - 125140) * 0.45
13     ENDIF
14   ENDIF
15 ENDIF
16 yearlyTax = tax

```

```
17 monthlyTax = tax / 12
18 weeklyTax = tax / 52
19 DISPLAY yearlyTax, monthlyTax, weeklyTax
20 END
```

Question 6 – Positive, Negative, or Zero

Find and display whether a given number is positive, negative, or zero.

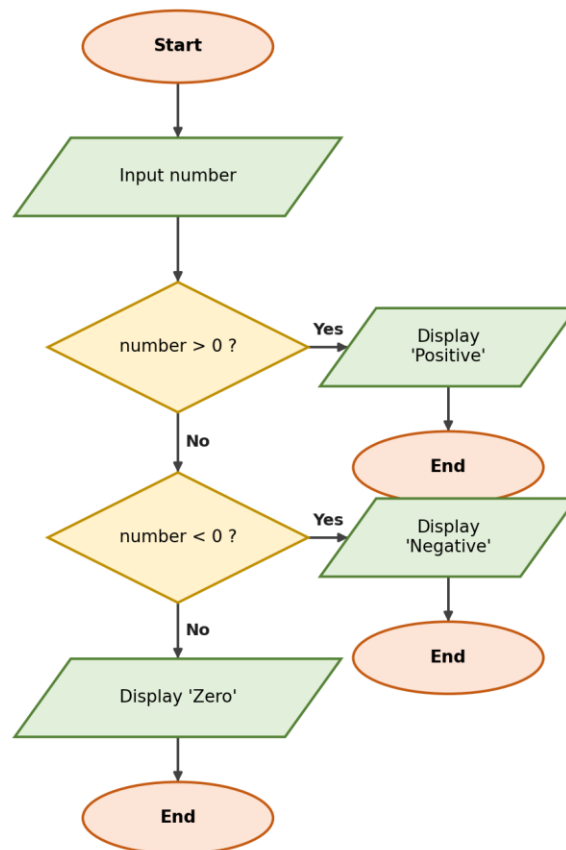
Identified Variables

Variable	Type	Description
number	Number	The number entered by the user

IPO Chart

Input	Process	Output
number	Compare number with 0	"Positive", "Negative", or "Zero"

Flowchart



Pseudocode

```
1 BEGIN
2   INPUT number
3   IF number > 0 THEN
4     DISPLAY "Positive"
5   ELSE
6     IF number < 0 THEN
7       DISPLAY "Negative"
8     ELSE
9       DISPLAY "Zero"
10    ENDIF
11  ENDIF
12 END
```

Question 7 – Nth Fibonacci Number

A Fibonacci series generates each term by adding the two preceding terms (1, 1, 2, 3, 5, 8, ...). The program takes a user input N and outputs the Nth Fibonacci number.

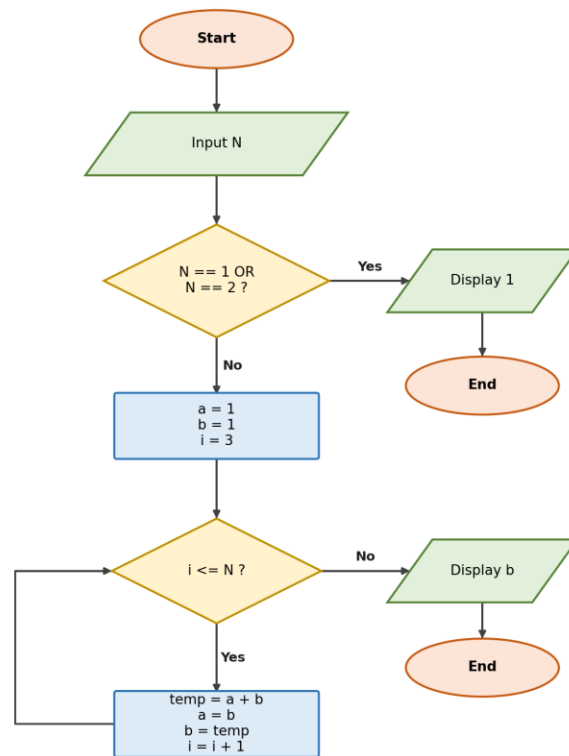
Identified Variables

Variable	Type	Description
N	<i>Integer</i>	Which Fibonacci term to compute
a, b	<i>Integer</i>	The two most recent Fibonacci values
temp	<i>Integer</i>	Temporary store when shifting a and b along
i	<i>Integer</i>	Loop counter (starts at 3)

IPO Chart

Input	Process	Output
N (position in sequence)	If $N \leq 2$: result = 1 Otherwise: iterate from $i=3$ to N, shifting a and b forward each step	Nth Fibonacci number

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT N
3    IF N == 1 OR N == 2 THEN
4      DISPLAY 1
5    ELSE
6      a = 1
7      b = 1
8      i = 3
9      WHILE i <= N
10       temp = a + b
11       a = b
12       b = temp
13       i = i + 1
14     ENDWHILE
15     DISPLAY b
16   ENDIF

```


Question 8 – Profit or Loss

Find whether a transaction gives a profit, a loss, or neither, using cost price (CP) and selling price (SP).

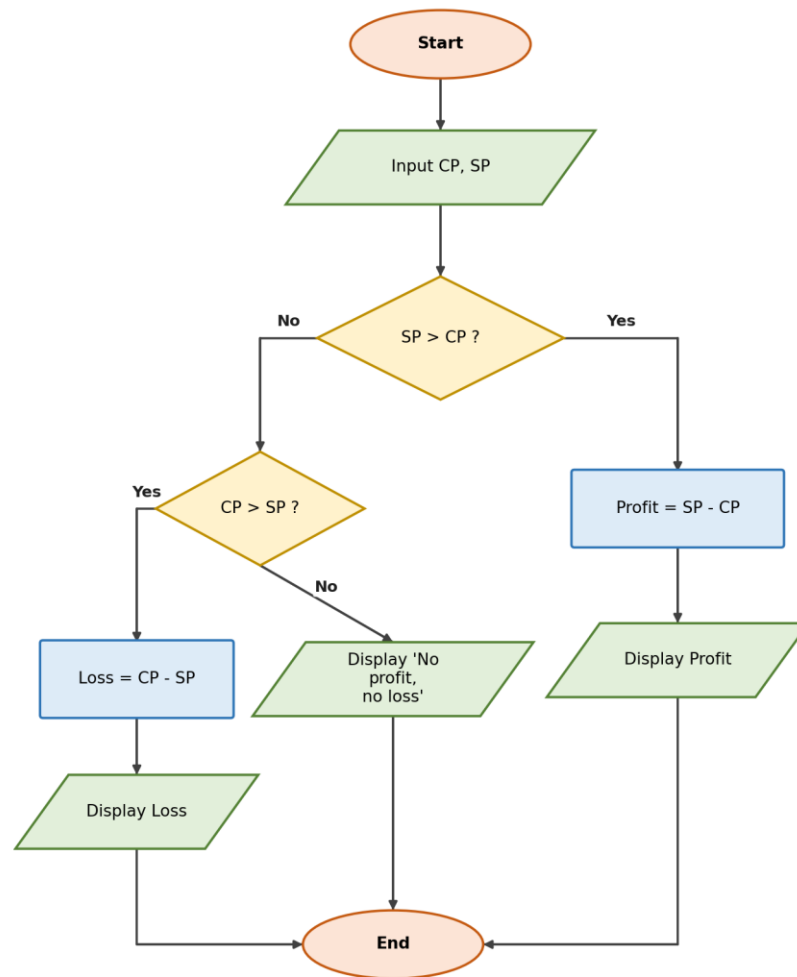
Identified Variables

Variable	Type	Description
CP	<i>Number</i>	Cost price paid for the item
SP	<i>Number</i>	Selling price received for the item
Profit	<i>Number</i>	$SP - CP$ (when $SP > CP$)
Loss	<i>Number</i>	$CP - SP$ (when $CP > SP$)

IPO Chart

Input	Process	Output
CP, SP	If $SP > CP$: Profit = $SP - CP$ If $CP > SP$: Loss = $CP - SP$ Otherwise: no profit, no loss	Profit, Loss, or "No profit, no loss"

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT CP, SP
3    IF SP > CP THEN
4      Profit = SP - CP
5      DISPLAY Profit
6    ELSE
7      IF CP > SP THEN
8        Loss = CP - SP
9        DISPLAY Loss
10     ELSE
11       DISPLAY "No profit, no loss"
12     ENDIF
13   ENDIF
14  END

```

Question 9 – Letter-to-Number Encryption

A program that takes the letters in a first name one at a time until the user presses 0, then outputs an encrypted string where A/a = 1, B/b = 2, ..., Z/z = 26. Example: F a h a d → 6 1 8 1 4.

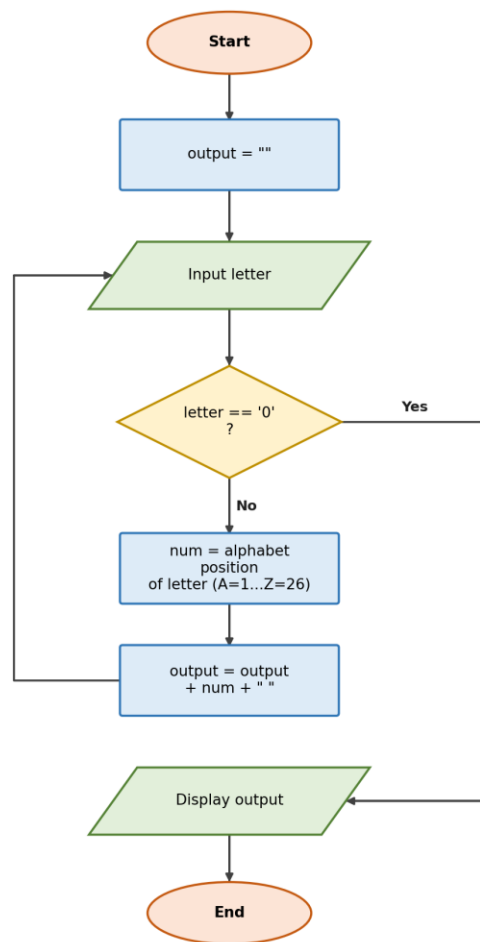
Identified Variables

Variable	Type	Description
letter	<i>Character</i>	Single letter entered each iteration
num	<i>Integer</i>	Numeric value of the letter (1–26)
output	<i>String</i>	Accumulated space-separated numbers

IPO Chart

Input	Process	Output
One letter at a time (until '0' is entered)	For each letter: num = position of letter in alphabet Append num and a space to output	Space-separated number string (e.g. "6 1 8 1 4")

Flowchart



Pseudocode

```

1  BEGIN
2      output = ""
3      LOOP
4          INPUT letter
5          IF letter == "0" THEN
6              EXIT LOOP
7          ENDIF
8          num = alphabet position of letter (A=1, ..., Z=26)
9          output = output + num + " "
10     ENDLOOP
11     DISPLAY output
12 END

```

Question 10 – Five Test Marks

Input five test marks, then calculate and display the total and average.

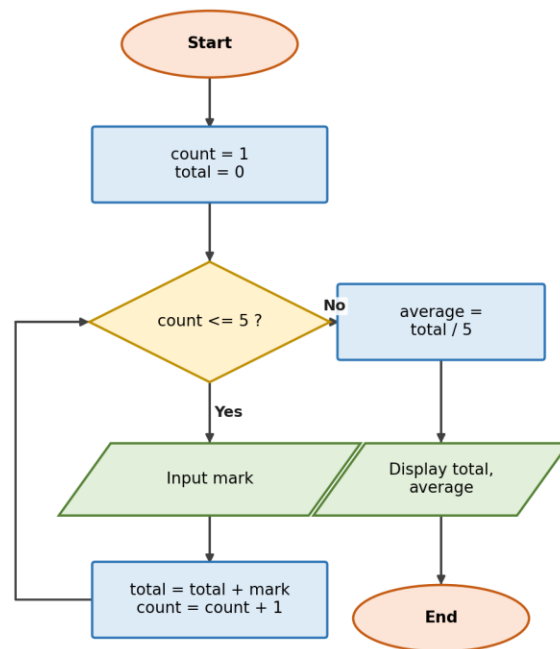
Identified Variables

Variable	Type	Description
mark	<i>Number</i>	A single test mark entered each iteration
total	<i>Number</i>	Running sum of all five marks
count	<i>Integer</i>	Loop counter (1 to 5)
average	<i>Number</i>	$\text{total} \div 5$

IPO Chart

Input	Process	Output
5 test marks (one per loop pass)	sum all marks; $\text{average} = \text{total} \div 5$	total, average

Flowchart



Pseudocode

```

1  BEGIN
2      count = 1
3      total = 0
4      WHILE count <= 5
5          INPUT mark
6          total = total + mark
7          count = count + 1
8      ENDWHILE
9      average = total / 5
10     DISPLAY total, average
11  END

```

Question 11 – Multiplication Table (1 to 12)

Display the multiplication table for a given number N from 1 to 12.

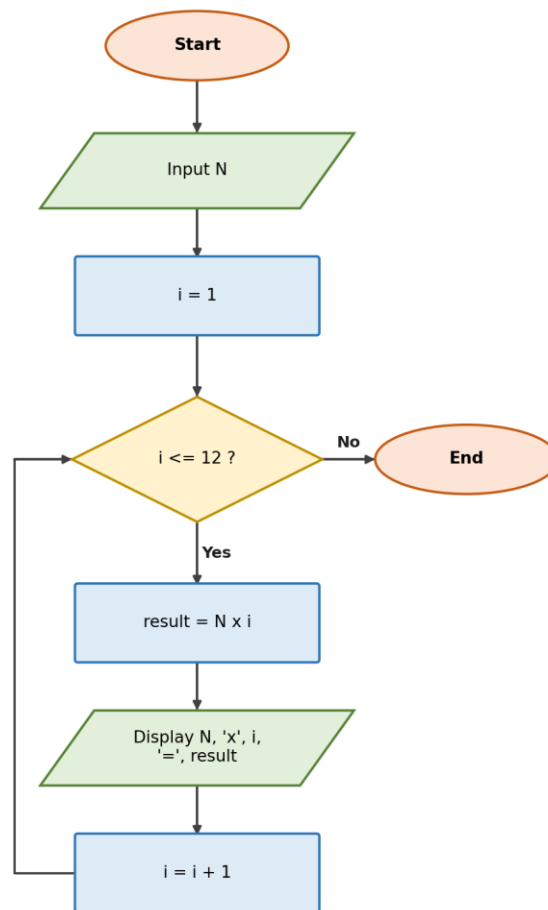
Identified Variables

Variable	Type	Description
N	<i>Integer</i>	The number whose table is to be printed
i	<i>Integer</i>	Multiplier, counts from 1 to 12
result	<i>Integer</i>	$N \times i$

IPO Chart

Input	Process	Output
N	For i = 1 to 12: result = $N \times i$	$N \times i$ = result for each i from 1 to 12

Flowchart



Pseudocode

```
1 BEGIN
2   INPUT N
3   i = 1
4   WHILE i <= 12
5     result = N * i
6     DISPLAY N, "x", i, "=", result
7     i = i + 1
8   ENDWHILE
9   END
```

Question 12 – Sum of First 100 Odd Numbers

Find and display the sum of the first 100 odd numbers ($1 + 3 + 5 + \dots + 199$).

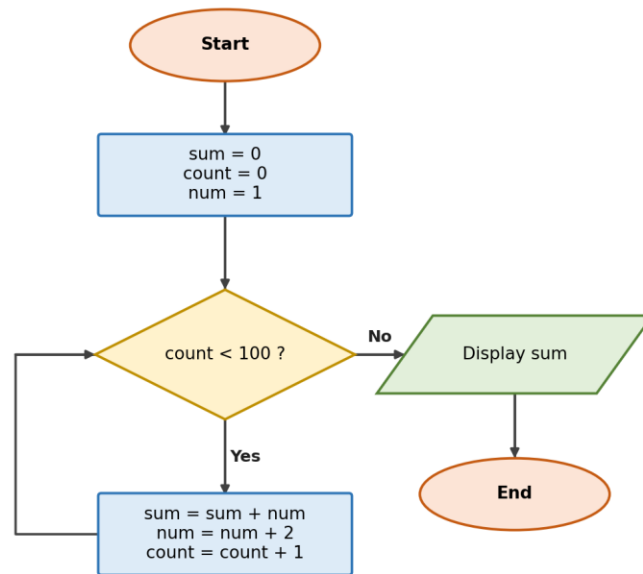
Identified Variables

Variable	Type	Description
sum	<i>Integer</i>	Running total of odd numbers
num	<i>Integer</i>	Current odd number (starts at 1, increases by 2)
count	<i>Integer</i>	How many odd numbers have been added so far

IPO Chart

Input	Process	Output
None (100 terms are fixed)	Start num=1, count=0 While count < 100: sum = sum + num num = num + 2 count = count + 1	sum (= 10,000)

Flowchart



Pseudocode

```
1  BEGIN
2      sum = 0
3      count = 0
4      num = 1
5      WHILE count < 100
6          sum = sum + num
7          num = num + 2
8          count = count + 1
9      ENDWHILE
10     DISPLAY sum
11  END
```

Question 13 – Grade from Mark

Display the grade for an input mark using the following ranges:

A	80 and above
B	70–79
C	60–69
D	50–59
F	Below 50

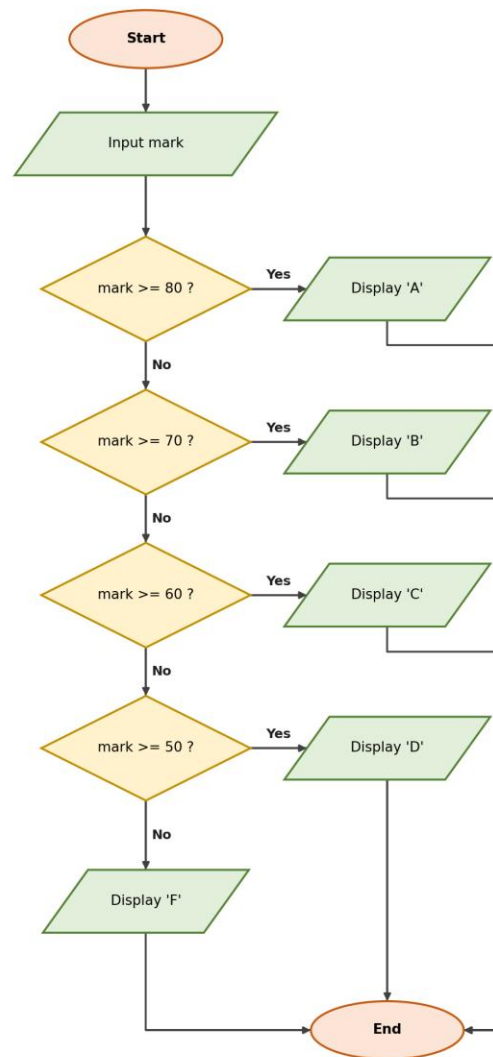
Identified Variables

Variable	Type	Description
mark	<i>Integer</i>	Test mark entered by the user (0–100)
grade	<i>Character</i>	Calculated grade letter (A, B, C, D, or F)

IPO Chart

Input	Process	Output
mark	Compare mark against grade boundaries in descending order (80, 70, 60, 50)	Grade letter: A, B, C, D, or F

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT mark
3    IF mark >= 80 THEN
4      DISPLAY "A"
5    ELSE
6      IF mark >= 70 THEN
7        DISPLAY "B"
8      ELSE
9        IF mark >= 60 THEN
10       DISPLAY "C"
11     ELSE
12       IF mark >= 50 THEN
13         DISPLAY "D"
14       ELSE
15         DISPLAY "F"
16     ENDIF
  
```

```
17         ENDIF  
18     ENDIF  
19 ENDIF  
20 END
```

Question 14 – Prime Numbers up to N

Output all the prime numbers between 1 and a user-input N.

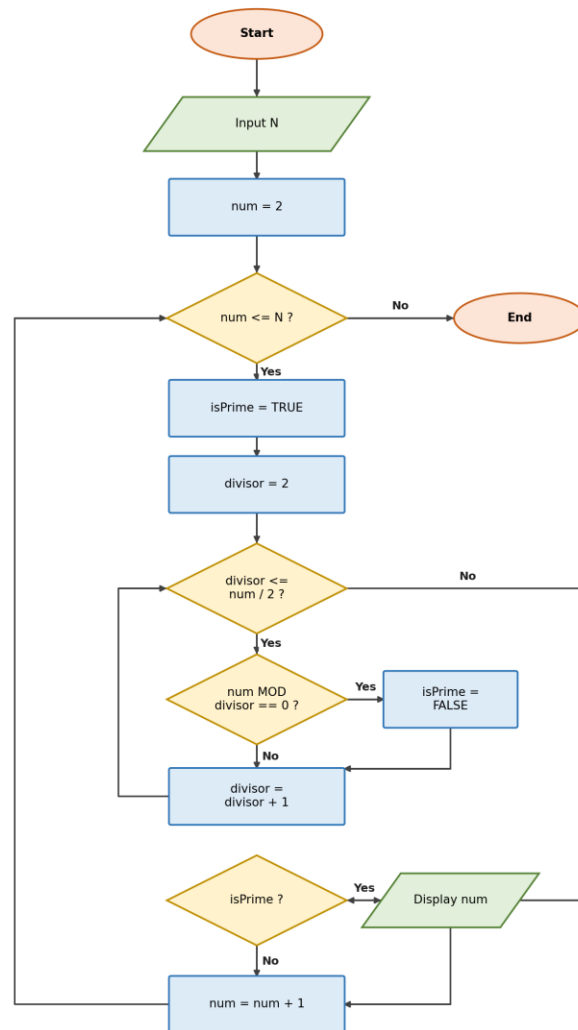
Identified Variables

Variable	Type	Description
N	<i>Integer</i>	The upper limit entered by the user
num	<i>Integer</i>	The candidate number being tested (2 to N)
divisor	<i>Integer</i>	Trial divisor (2 to num÷2)
isPrime	<i>Boolean</i>	TRUE if no divisor divides num evenly

IPO Chart

Input	Process	Output
N	For num = 2 to N: isPrime = TRUE For divisor = 2 to num÷2: If num MOD divisor == 0: isPrime = FALSE If isPrime: output num	All prime numbers from 2 to N

Flowchart



Pseudocode

```

1  BEGIN
2    INPUT N
3    num = 2
4    WHILE num <= N
5      isPrime = TRUE
6      divisor = 2
7      WHILE divisor <= num / 2
8        IF num MOD divisor == 0 THEN
9          isPrime = FALSE
10       ENDIF
11       divisor = divisor + 1
12     ENDWHILE
13     IF isPrime THEN
14       DISPLAY num
15     ENDIF
16     num = num + 1

```

17

ENDWHILE

18

END

Question 15 – Number Guessing Game

Generate a random number from 1 to 10. Ask the user to guess; display whether the guess is too low, too high, or correct.

Extension: keep counting guesses until the correct answer is found or the user types "exit". Display the total guess count when the game ends.

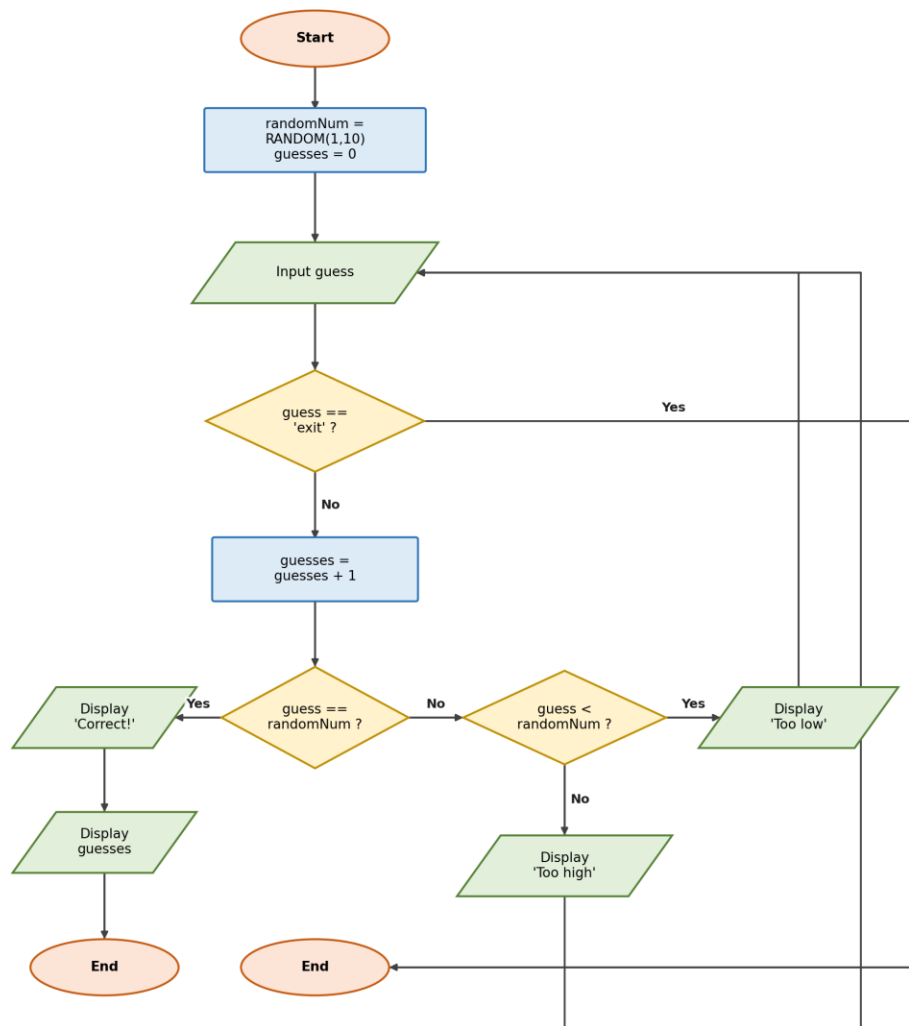
Identified Variables

Variable	Type	Description
randomNum	<i>Integer</i>	The secret number, generated randomly (1–10)
guess	<i>Integer/String</i>	User's guess, or "exit" to quit
guesses	<i>Integer</i>	Count of guesses made so far

IPO Chart

Input	Process	Output
Random number (1–10); user guesses (until correct or "exit")	Compare guess with randomNum; if correct or exit: display count and stop; otherwise: display Too low / Too high	"Too low", "Too high", "Correct!", guess count

Flowchart



Pseudocode

```

1  BEGIN
2    randomNum = RANDOM(1, 10)
3    guesses = 0
4    LOOP
5      INPUT guess
6      IF guess == "exit" THEN
7        DISPLAY "Number of guesses: ", guesses
8        EXIT LOOP
9      ENDIF
10     guesses = guesses + 1
11     IF guess == randomNum THEN
12       DISPLAY "Correct!"
13       DISPLAY "Number of guesses: ", guesses
14       EXIT LOOP
15     ELSE
16       IF guess < randomNum THEN

```

```
17         DISPLAY "Too low"
18     ELSE
19         DISPLAY "Too high"
20     ENDIF
21 ENDIF
22 ENDLOOP
23 END
```