

Pseudocode Solutions

Worked Solutions in Structured Pseudocode

Pseudocode Conventions

Every solution in this document uses the same small set of structured-English keywords, shown in bold below:

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

INPUT x - reads a value from the user into variable x.

DISPLAY x - outputs the value of x to the user.

IF condition **THEN** ... **ELSE** ... **ENDIF** - branches the flow depending on a condition.

WHILE condition ... **ENDWHILE** - repeats a block while a condition holds (checked before each pass).

LOOP ... **EXIT LOOP** ... **ENDLOOP** - a loop that can only be exited from the middle, used when the exit test does not naturally sit at the very top or bottom of the loop body.

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

RANDOM(a, b) - generates a random whole number between a and b inclusive.

= is assignment; ==, >, <, >=, <= are comparisons.

Question 1

Draw a flow chart to find and display the largest of the two inputted numbers.

```
1 BEGIN
2   INPUT A, B
3   IF A > B THEN
4     DISPLAY A
5   ELSE
6     DISPLAY B
7   ENDIF
8 END
```

Question 2

Draw a flow chart to find and display the larger of two input numbers. Add a validation to check whether both numbers are equal, if it is display 'both numbers are the same'.

```
1  BEGIN
2      INPUT A, B
3      IF A == B THEN
4          DISPLAY "both numbers are the same"
5      ELSE
6          IF A > B THEN
7              DISPLAY A
8          ELSE
9              DISPLAY B
10         ENDIF
11     ENDIF
12 END
```

Question 3

Draw a flow chart to calculate pay.

Hint: Pay = hours x rate

```
1  BEGIN
2      INPUT hours, rate
3      pay = hours * rate
4      DISPLAY pay
5  END
```

Question 4

Draw a flow chart to calculate weekly wages including normal hours and overtime hours, overtime hours are paid 'time-and-half' of rate of pay. Hours worked more than 35 hours are considered as overtime hours.

```
1  BEGIN
2    INPUT hours, rate
3    IF hours > 35 THEN
4      overtime = hours - 35
5      normalPay = 35 * rate
6      overtimePay = overtime * rate * 1.5
7      pay = normalPay + overtimePay
8    ELSE
9      pay = hours * rate
10   ENDIF
11   DISPLAY pay
12  END
```

Question 5

Draw a flowchart to find the largest of three input numbers A, B, and C.

```
1  BEGIN
2      INPUT A, B, C
3      IF A > B THEN
4          IF A > C THEN
5              DISPLAY A
6          ELSE
7              DISPLAY C
8          ENDIF
9      ELSE
10         IF B > C THEN
11             DISPLAY B
12         ELSE
13             DISPLAY C
14         ENDIF
15     ENDIF
16 END
```

Question 6

Draw a flowchart to log in to Facebook account.

```
1  BEGIN
2      LOOP
3          INPUT username, password
4          IF username and password valid THEN
5              DISPLAY "Login successful"
6              DISPLAY "Open news feed"
7              EXIT LOOP
8          ELSE
9              DISPLAY "Invalid login, try again"
10         ENDIF
11     ENDLOOP
12 END
```

Question 7

Draw a flow chart to find and display a given number is odd or even.

```
1 BEGIN
2   INPUT number
3   remainder = number MOD 2
4   IF remainder == 0 THEN
5     DISPLAY "Even"
6   ELSE
7     DISPLAY "Odd"
8   ENDIF
9 END
```

Question 8

Flowchart – How to find profit or loss.

Hint: Profit = $SP - CP$; Loss = $CP - SP$

```
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

Draw a flow chart to print “Hello World” 10 times.

```
1  BEGIN
2      i = 1
3      WHILE i <= 10
4          DISPLAY "Hello World"
5          i = i + 1
6      ENDWHILE
7  END
```

Question 10

Draw a flowchart to find the total and average of several input numbers.

The user is first asked how many numbers (N) they want to enter; the loop then runs exactly N times, reading one number per pass and keeping a running count, before the average is calculated.

```
1  BEGIN
2    INPUT N
3    sum = 0
4    count = 0
5    WHILE count < N
6      INPUT number
7      sum = sum + number
8      count = count + 1
9    ENDWHILE
10   average = sum / count
11   DISPLAY sum, average
12  END
```

Question 11

Flowchart of multiplication table for a given number N.

```
1  BEGIN
2      INPUT N
3      i = 1
4      WHILE i <= 10
5          result = N * i
6          DISPLAY N, "x", i, "=", result
7          i = i + 1
8      ENDWHILE
9  END
```

Question 12

Draw a flowchart to find and display the sum of first 50 natural numbers.

```
1  BEGIN
2      sum = 0
3      i = 1
4      WHILE i <= 50
5          sum = sum + i
6          i = i + 1
7      ENDWHILE
8      DISPLAY sum
9  END
```

Question 13

Draw a flow to chart to display Grade of input mark.

A	80 and over
B	70 to 79
C	60 to 69
D	50 to 59
F	Less than 50

```
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

Draw a flowchart to find the sums of all the even numbers between 1 and 20 inclusive and then displays the sum.

```
1  BEGIN
2      sum = 0
3      i = 1
4      WHILE i <= 20
5          IF i MOD 2 == 0 THEN
6              sum = sum + i
7          ENDIF
8          i = i + 1
9      ENDWHILE
10     DISPLAY sum
11     END
```

Question 15

Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right.

Hint: remember to use the user input lessons from the very first exercise.

Extras: keep the game going until the user types “exit”. Keep track of how many guesses the user has taken, and when the game ends, print this out.

```
1  BEGIN
2      number = RANDOM(1, 9)
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 == number THEN
12             DISPLAY "Correct!"
13             DISPLAY "Number of guesses: ", guesses
14             EXIT LOOP
15         ELSE
16             IF guess < number THEN
17                 DISPLAY "Too low"
18             ELSE
19                 DISPLAY "Too high"
20             ENDIF
21         ENDIF
22     ENDLOOP
23 END
```

Draw a flowchart for computing factorial N ($N!$).

```
1  BEGIN
2      INPUT N
3      fact = 1
4      i = 1
5      WHILE i <= N
6          fact = fact * i
7          i = i + 1
8      ENDWHILE
9      DISPLAY fact
10 END
```