G.C.E. O/L Grade 11 ICT · Lesson 1

1.7 Iteration: FOR, WHILE and REPEAT–UNTIL Loops

Syllabus 10.7. A **loop** repeats statements. If we **know how many times** to repeat (e.g. 10 times), we use a **FOR** loop. If we **don't know** (e.g. "until the user types −1"), we use a **WHILE** loop, which **tests the condition at the beginning**, or a **REPEAT–UNTIL** loop, which **tests at the end**. You will also learn to **convert** one loop into another.

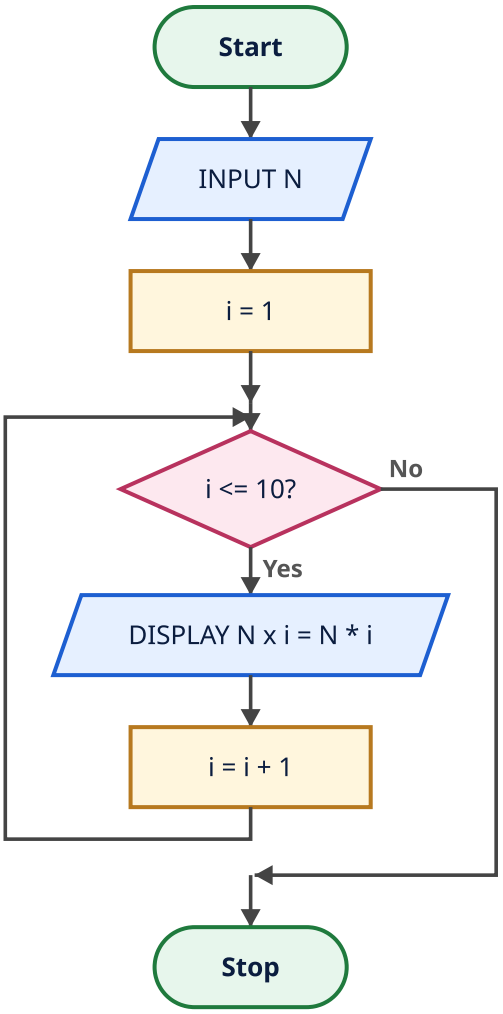
1. Known number of times: FOR

A **FOR** loop uses a **counter** that goes from a start value to an end value. The loop body runs once for each value. Example: the multiplication table of a number, 1 to 10.

Pseudocode

```
BEGIN
  INPUT N
  FOR i = 1 TO 10
    DISPLAY N, " x ", i, " = ", N * i
  ENDFOR
END
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved



Flowchart – For loop (known count): multiplication table

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

For loop (known count): multiplication table

Pascal

```
program TimesTable;
var
  n, i: integer;
begin
  write('Enter a number: ');
  readln(n);
  for i := 1 to 10 do
    writeln(n, ' x ', i, ' = ', n * i);
end.
```

Python

```
n = int(input("Enter a number: "))
for i in range(1, 11):
    print(n, "x", i, "=", n * i)
```

Sample run

```
Enter a number: 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

Trace of a FOR loop – factorial of 5 (F = 1 × 2 × 3 × 4 × 5):

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Factorial with a FOR loop

Pascal

```
program Factorial;
var
  n, i, f: integer;
begin
  write('Enter N: ');
  readln(n);
  f := 1;
  for i := 1 to n do
    f := f * i;
  writeln('Factorial = ', f);
end.
```

Python

```
n = int(input("Enter N: "))
f = 1
for i in range(1, n + 1):
    f = f * i
print("Factorial =", f)
```

Sample run

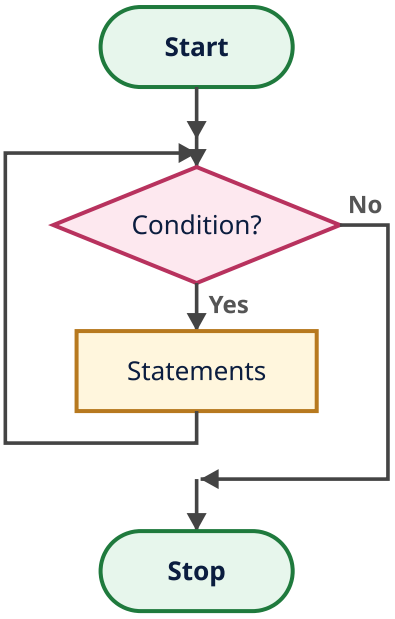
```
Enter N: 5
Factorial = 120
```

i	F = F * i
1	1
2	2
3	6
4	24
5	120

2. Unknown count, test at the beginning: WHILE

A **WHILE** loop checks the condition **first**. If it is True, the body runs and the condition is checked again. If it is False at the start, the body runs **zero times**. Example: enter marks until −1 is typed. −1 is a **sentinel** (a stop signal), not a mark.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

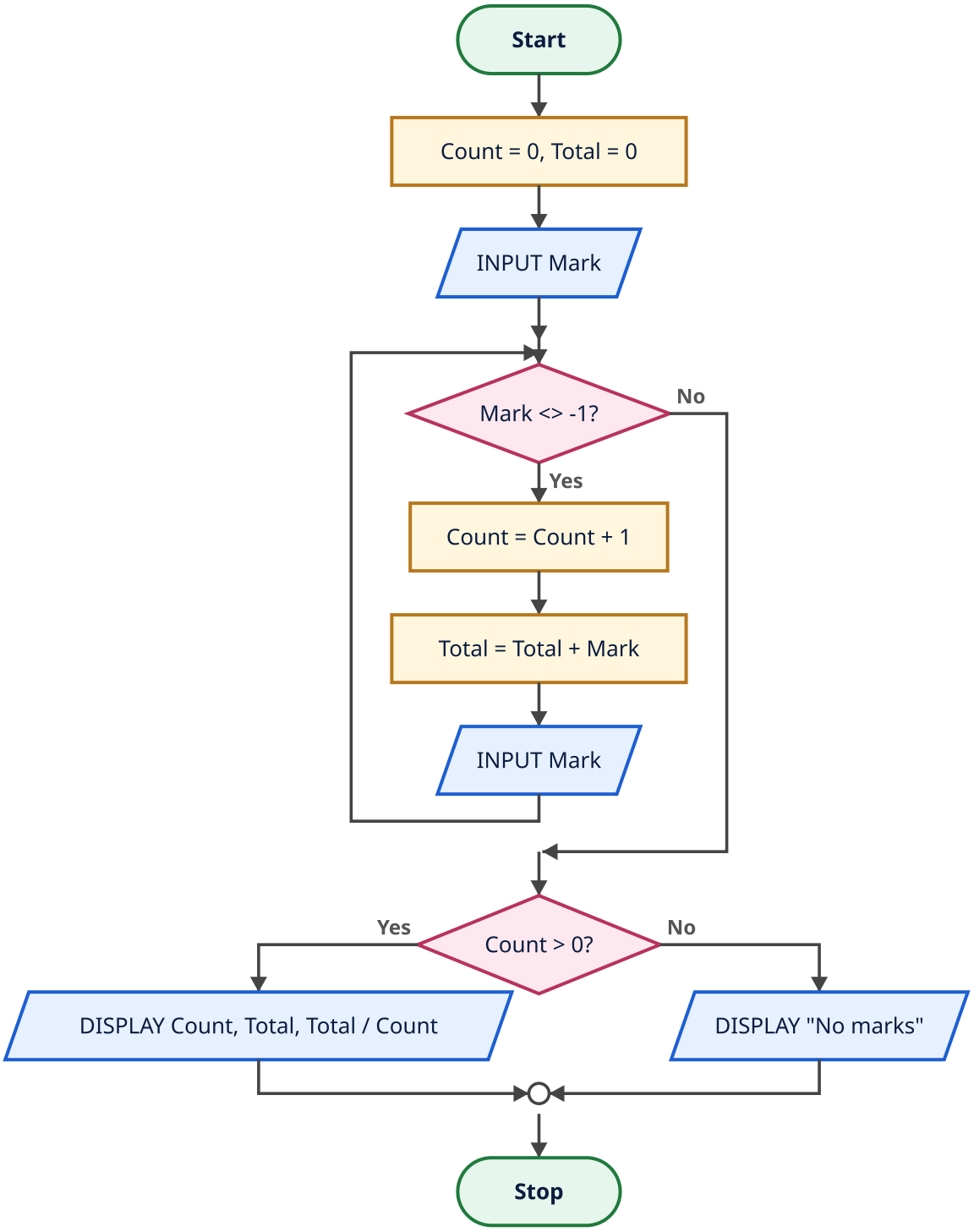


General form of a WHILE loop (test at the beginning)

Pseudocode

```
BEGIN
  Count = 0
  Total = 0
  INPUT Mark
  WHILE Mark <> -1 DO
    Count = Count + 1
    Total = Total + Mark
    INPUT Mark
  ENDWHILE
  IF Count > 0 THEN
    DISPLAY Count, Total, Total / Count
  ELSE
    DISPLAY "No marks entered"
  ENDIF
END
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved



Flowchart – While loop (unknown count): marks until −1

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

While loop (unknown count): marks until −1

Pascal

```
program MarksUntilStop;
var
  mark, count, total: integer;
begin
  count := 0;
  total := 0;
  write('Enter mark (-1 to stop): ');
  readln(mark);
  while mark <> -1 do
  begin
    count := count + 1;
    total := total + mark;
    write('Enter mark (-1 to stop): ');
    readln(mark);
  end;
  if count > 0 then
  begin
    writeln('Count = ', count);
    writeln('Total = ', total);
    writeln('Average = ', total / count:0:2);
  end
  else
    writeln('No marks entered');
end.
```

Python

```
count = 0
total = 0
mark = int(input("Enter mark (-1 to stop): "))
while mark != -1:
    count = count + 1
    total = total + mark
    mark = int(input("Enter mark (-1 to stop): "))
if count > 0:
    print("Count =", count)
    print("Total =", total)
    print(f"Average = {total / count:.2f}")
else:
    print("No marks entered")
```

Sample run 1

```
Enter mark (-1 to stop): 70
Enter mark (-1 to stop): 85
Enter mark (-1 to stop): 64
Enter mark (-1 to stop): -1
Count = 3
Total = 219
Average = 73.00
```

Sample run 2

```
Enter mark (-1 to stop): -1
No marks entered
```

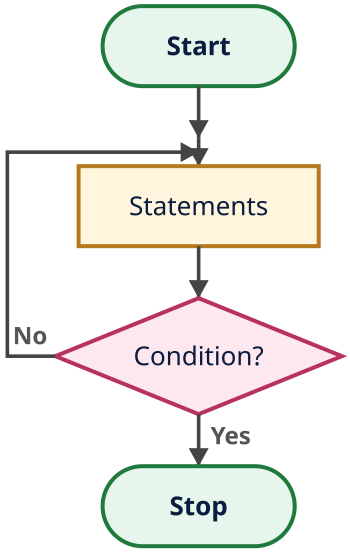
© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Mark	Count	Total
70	1	70
85	2	155
64	3	219
-1	3	219

Count = 3, Total = 219, so Average = 219 ÷ 3 = 73.00. If the first input is −1, the loop body never runs and the program says “No marks entered”.

3. Unknown count, test at the end: REPEAT–UNTIL

A **REPEAT–UNTIL** loop runs the body **first** and checks the condition **at the end**, so the body always runs **at least once**. It stops when the condition becomes **True**. Perfect for checking input: keep asking until a valid mark (0–100) is typed.

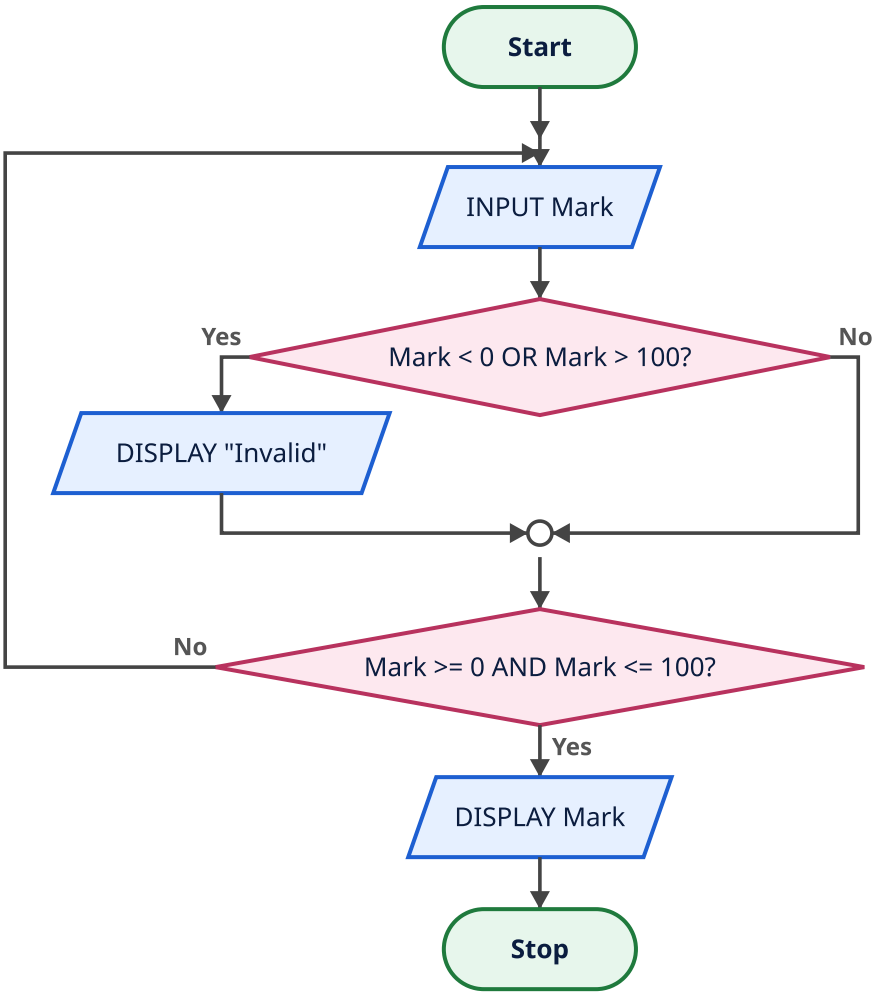


General form of a REPEAT–UNTIL loop (test at the end)

Pseudocode

```
BEGIN
  REPEAT
    INPUT Mark
    IF Mark < 0 OR Mark > 100 THEN
      DISPLAY "Invalid. Try again."
    ENDIF
  UNTIL Mark >= 0 AND Mark <= 100
  DISPLAY Mark
END
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved



Flowchart – Repeat–Until (checked at the end): accept a valid mark

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Repeat–Until (checked at the end): accept a valid mark

Pascal

```
program ValidMark;
var
  mark: integer;
begin
  repeat
    write('Enter mark (0-100): ');
    readln(mark);
    if (mark < 0) or (mark > 100) then
      writeln('Invalid. Try again.');
  until (mark >= 0) and (mark <= 100);
  writeln('Mark accepted: ', mark);
end.
```

Python

```
while True:
    mark = int(input("Enter mark (0-100): "))
    if 0 <= mark <= 100:
        break
    print("Invalid. Try again.")
print("Mark accepted:", mark)
```

Sample run

```
Enter mark (0-100): 120
Invalid. Try again.
Enter mark (0-100): -5
Invalid. Try again.
Enter mark (0-100): 76
Mark accepted: 76
```

> 💡 Python has **no REPEAT–UNTIL**. We copy it with while True: … and break when the stop condition is reached.

4. Comparing the three loops

	FOR	WHILE	REPEAT–UNTIL
Number of times	Known	Unknown	Unknown
Condition tested	Counter checked at the start	At the beginning	At the end
Minimum runs	0	0	1
Stops when	Counter passes the end value	Condition is False	Condition is True
Pascal	for i := 1 to 10 do	while … do	repeat … until …;
Python	for i in range(1, 11):	while …:	while True: … break

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. Converting one loop into another

The same task – display the even numbers 2, 4, 6, 8, 10 – written with each loop. To change a FOR into a WHILE: (1) set the counter **before** the loop, (2) write the condition, (3) **increase the counter inside** the loop. For REPEAT–UNTIL, **reverse** the condition (WHILE i <= 10 becomes UNTIL i > 10).

Pseudocode

```
BEGIN
  FOR i = 1 TO 5
    DISPLAY i * 2
  ENDFOR
END
```

Even numbers 2–10 with FOR

Pascal

```
program EvenFor;
var
  i: integer;
begin
  for i := 1 to 5 do
    writeln(i * 2);
end.
```

Python

```
for i in range(2, 11, 2):
    print(i)
```

Sample run

```
2
4
6
8
10
```

Pseudocode

```
BEGIN
  i = 2
  WHILE i <= 10 DO
    DISPLAY i
    i = i + 2
  ENDWHILE
END
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Even numbers 2–10 with WHILE

Pascal

```
program EvenWhile;
var
  i: integer;
begin
  i := 2;
  while i <= 10 do
  begin
    writeln(i);
    i := i + 2;
  end;
end.
```

Python

```
i = 2
while i <= 10:
    print(i)
    i = i + 2
```

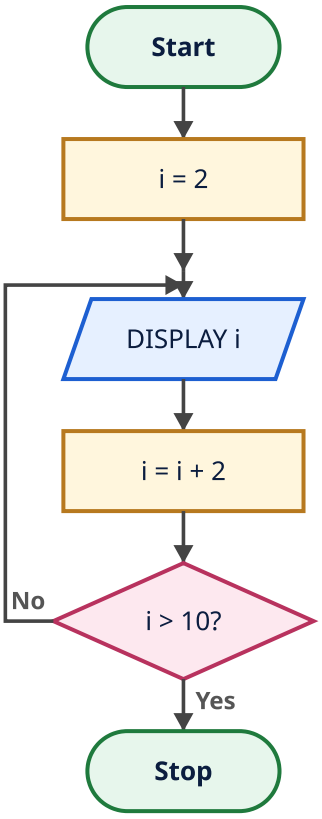
Sample run

```
2
4
6
8
10
```

Pseudocode

```
BEGIN
  i = 2
  REPEAT
    DISPLAY i
    i = i + 2
  UNTIL i > 10
END
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved



Flowchart – Even numbers 2–10 with REPEAT–UNTIL

Even numbers 2–10 with REPEAT–UNTIL

Pascal

```
program EvenRepeat;
var
  i: integer;
begin
  i := 2;
  repeat
    writeln(i);
    i := i + 2;
  until i > 10;
end.
```

Python

```
i = 2
while True:          # Python has no repeat-until
    print(i)
    i = i + 2
    if i > 10:
        break
```

Sample run

```
2
4
6
8
10
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

💡 Danger – an **infinite loop**: if you forget i = i + 2 inside a WHILE loop, i stays 2 for ever and the loop never stops. Always check that something inside the loop moves it towards the end.

Key Words

Loop – Statements that are repeated.	**FOR loop** – A loop that runs a known number of times using a counter.
WHILE loop – A loop that tests its condition at the beginning.	**REPEAT–UNTIL loop** – A loop that tests its condition at the end; runs at least once.
Counter – A variable that counts the loop runs, e.g. i.	**Pre-test loop** – Condition checked before the body (WHILE, FOR).
Post-test loop – Condition checked after the body (REPEAT–UNTIL).	**Sentinel value** – A special value that means “stop”, e.g. −1.
Infinite loop – A loop that never stops.	**Validation** – Checking that input is sensible, e.g. a mark from 0 to 100.
range() – Python function giving numbers for a FOR loop; range(1, 11) is 1 to 10.	**break** – Python word that leaves a loop at once.

Sample Q&A

1. When do we use a FOR loop?

> **Answer:** When we know how many times to repeat.

2. What is the main difference between WHILE and REPEAT–UNTIL?

> **Answer:** WHILE tests at the beginning (may run 0 times); REPEAT–UNTIL tests at the end (runs at least once).

3. What is a sentinel value?

> **Answer:** A special value such as −1 that tells the loop to stop.

4. What is the factorial of 5?

> **Answer:** 120.

5. How many times does for i := 1 to 10 run?

> **Answer:** 10 times.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. What causes an infinite loop?

> **Answer:** The condition never changes, e.g. forgetting to increase the counter.

MCQ

1. A loop for a known number of times is...

A. FOR
B. WHILE
C. IF
D. REPEAT–UNTIL

2. Which loop always runs at least once?

A. WHILE
B. FOR
C. None
D. REPEAT–UNTIL

3. Which loop tests the condition at the beginning?

A. CASE
B. WHILE
C. ENDIF
D. REPEAT–UNTIL

4. REPEAT ... UNTIL condition stops when the condition is...

A. zero
B. False
C. True
D. equal to 10

5. In Python, range(1, 11) gives...

A. 1 to 10
B. 0 to 10
C. 1 to 11
D. 0 to 11

6. The output of the factorial program for N = 5 is...

A. Factorial = 60
B. Factorial = 120
C. Factorial = 15
D. Factorial = 25

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

7. Marks 70, 85, 64 then −1 are entered. The average shown is…

A. 54.50
B. 72.67
C. 73.00
D. 219.00

8. In the marks program, −1 is called a…

A. data type
B. counter
C. constant mark
D. sentinel value

9. If the first input is −1, the WHILE loop body runs…

A. 0 times
B. forever
C. 3 times
D. 1 time

10. Python has no REPEAT–UNTIL. We use…

A. while True: … break
B. repeat()
C. do … loop
D. for … until

11. Changing “WHILE i <= 10” to REPEAT–UNTIL gives…

A. UNTIL i < 10
B. UNTIL i <= 10
C. UNTIL i > 10
D. UNTIL i = 2

12. Forgetting to increase the counter inside a WHILE loop causes…

A. a syntax error
B. no output
C. an infinite loop
D. a faster program

13. How many lines does the multiplication-table program print after the input?

A. 9
B. 10
C. 11
D. 7

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

14. Asking again and again until a valid mark is typed is best done with…

A. FOR
B. a constant
C. CASE
D. REPEAT–UNTIL

15. for i := 1 to 5 do write(i * 2) prints…

A. 1 2 3 4 5
B. 2 4 6 8 10
C. 1 3 5 7 9
D. 2 4 6 8

Structured Questions

1. (i) Give one difference between a pre-test and a post-test loop. (2 marks)
(ii) Give one example of each in Pascal. (2 marks) *(4 marks)*

> (i) A pre-test loop checks the condition before the body and may run 0 times; a post-test loop checks after and runs at least once. (ii) while … do (pre-test); repeat … until (post-test).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Write a program (Pascal or Python) to input a number and display its multiplication table from 1 to 10. (4 marks) *(4 marks)*

Input N (1); FOR i = 1 to 10 (1); calculate N × i (1); display in the form 7 x 1 = 7 (1).

For loop (known count): multiplication table

Pascal

```
program TimesTable;
var
  n, i: integer;
begin
  write('Enter a number: ');
  readln(n);
  for i := 1 to 10 do
    writeln(n, ' x ', i, ' = ', n * i);
end.
```

Python

```
n = int(input("Enter a number: "))
for i in range(1, 11):
    print(n, "x", i, "=", n * i)
```

Sample run

```
Enter a number: 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. Write pseudocode that keeps reading marks until −1 is entered, then displays the count, total and average. (5 marks) *(5 marks)*

Count = 0, Total = 0 (1); input Mark before the loop (1); WHILE Mark <> −1 add to Count and Total (1); input Mark inside the loop (1); display Count, Total, Total / Count (1).

Pseudocode

```
BEGIN
  Count = 0
  Total = 0
  INPUT Mark
  WHILE Mark <> -1 DO
    Count = Count + 1
    Total = Total + Mark
    INPUT Mark
  ENDWHILE
  IF Count > 0 THEN
    DISPLAY Count, Total, Total / Count
  ELSE
    DISPLAY "No marks entered"
  ENDIF
END
```

Mark	Count	Total
70	1	70
85	2	155
64	3	219
-1	3	219

4. Rewrite this FOR loop as a WHILE loop: FOR i = 1 TO 5 : DISPLAY i * 2 : ENDFOR. (3 marks) *(3 marks)*

i = 1 (1); WHILE i <= 5 DO DISPLAY i * 2 (1); i = i + 1; ENDWHILE (1). (Or the equivalent with i = 2 … i <= 10 … i = i + 2.)

Pseudocode

```
BEGIN
  i = 2
  WHILE i <= 10 DO
    DISPLAY i
    i = i + 2
  ENDWHILE
END
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. Complete a trace table for the factorial program with N = 5 and give the output. (3 marks)
(3 marks)

F = 1, 2, 6, 24, 120 for i = 1 to 5 (2); output Factorial = 120 (1).

i	F = F * i
1	1
2	2
3	6
4	24
5	120

Summary

- FOR: known number of times.
- WHILE: test at the beginning; may run 0 times.
- REPEAT–UNTIL: test at the end; runs at least once.
- REPEAT stops when the condition is True.
- A sentinel (−1) tells the loop to stop.
- Python: for … in range(); while; while True + break.
- Convert: set counter, test, increase inside.
- Always move the loop towards its end – avoid infinite loops.

MCQ answer key – 1.7

1. **A** – Known count → FOR.
2. **D** – Test at the end.
3. **B** – Pre-test.
4. **C** – Until it becomes true.
5. **A** – End not included.
6. **B** – 1×2×3×4×5.
7. **C** – 219 ÷ 3.
8. **D** – Stop signal.
9. **A** – Condition false at start.
10. **A** – while True with break.
11. **C** – Reverse the condition.
12. **C** – Never stops.
13. **B** – i = 1 to 10.
14. **D** – Must ask at least once.
15. **B** – Doubles of 1–5.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.