G.C.E. O/L ICT · 11 ශ්‍රේණිය · පාඩම 1

1.7 පුනරාවර්තනය: FOR, WHILE සහ REPEAT–UNTIL ලූප

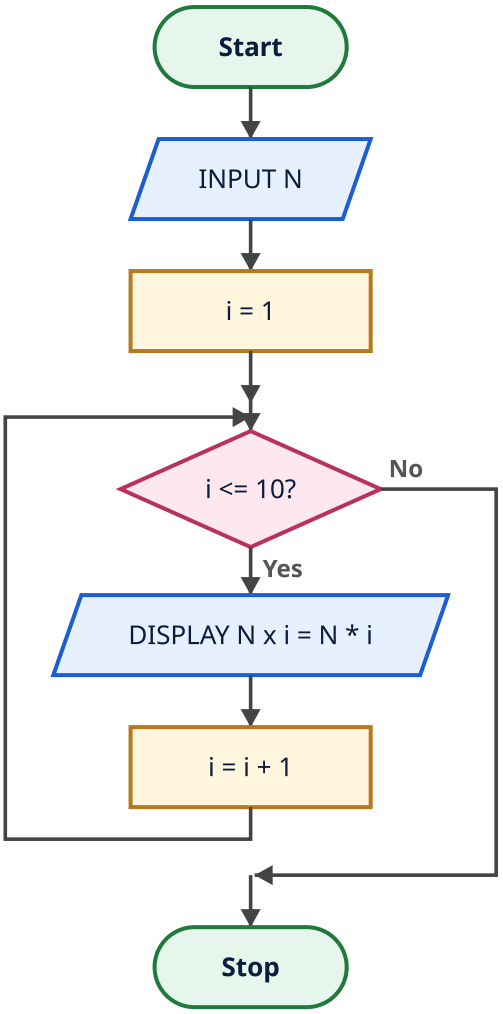
විෂය නිර්දේශය 10.7. ලූපයක් ප්‍රකාශ නැවත නැවත ක්‍රියාත්මක කරයි. **කී වරක්** කළ යුතුදැයි **දන්නේ නම්** (උදා 10 වරක්) **FOR** ලූපයක් භාවිත කරමු. **නොදන්නේ නම්** (උදා “පරිශීලකයා −1 ටයිප් කරන තුරු”) **ආරම්භයේදී කොන්දේසිය පරීක්ෂා කරන WHILE** ලූපයක් හෝ **අවසානයේදී පරීක්ෂා කරන REPEAT–UNTIL** ලූපයක් භාවිත කරමු. එක් ලූපයක් තවත් එකකට **පරිවර්තනය** කිරීමද ඉගෙන ගනිමු.

1. දන්නා වාර ගණන: FOR

FOR ලූපයක් ආරම්භක අගයක සිට අවසාන අගයක් දක්වා යන **ගණකයක්** (counter) භාවිත කරයි. සෑම අගයකටම ලූපයේ ඇතුළත එක් වරක් ධාවනය වේ. උදාහරණය: සංඛ්‍යාවක ගුණන වගුව, 1 සිට 10 දක්වා.

ව්‍යාජ කේතය (Pseudocode)

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



ගැලීම් සටහන – For loop (දන්නා වාර ගණන): ගුණන වගුව

For loop (දන්නා වාර ගණන): ගුණන වගුව

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)
```

නියැදි ධාවනය

```
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
```

FOR ලූපයක අනුරේඛනය – 5 හි සාධකීය (F = 1 × 2 × 3 × 4 × 5):

FOR loop මගින් ක්‍රමාරෝපිතය (factorial)

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)
```

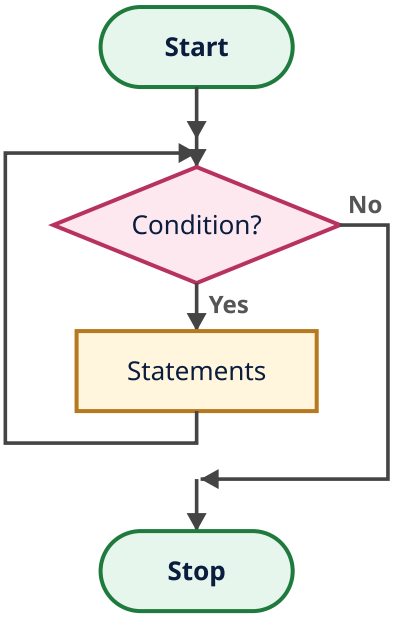
නියැදි ධාවනය

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

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

2. නොදන්නා වාර ගණන, ආරම්භයේ පරීක්ෂාව: WHILE

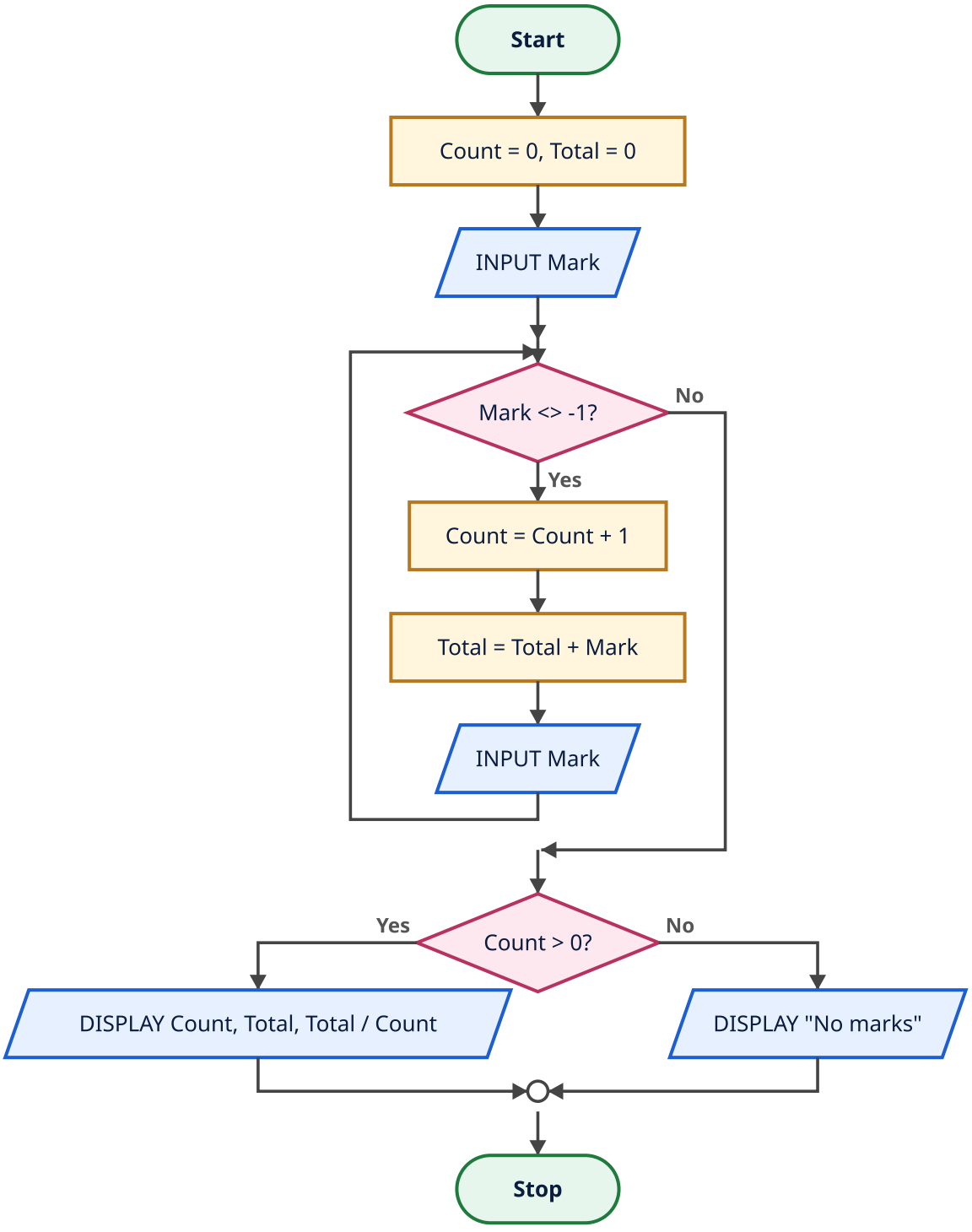
WHILE ලූපයක් කොන්දේසිය **මුලින්ම** පරීක්ෂා කරයි. සත්‍ය නම් ඇතුළත ධාවනය වී කොන්දේසිය නැවත පරීක්ෂා කරයි. ආරම්භයේදීම අසත්‍ය නම් ඇතුළත **ශූන්‍ය වරක්** ධාවනය වේ. උදාහරණය: −1 ටයිප් කරන තුරු ලකුණු ඇතුළත් කිරීම. −1 යනු ලකුණක් නොව **sentinel** (නැවතීමේ සංඥාවක්) අගයකි.



WHILE ලූපයක සාමාන්‍ය ආකාරය (ආරම්භයේ පරීක්ෂාව)

ව්‍යාජ කේතය (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
```



ගැලීම් සටහන – While loop (නොදන්නා වාර ගණන): −1 දක්වා ලකුණු

While loop (නොදන්නා වාර ගණන): −1 දක්වා ලකුණු

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")
```

නියැදි ධාවනය 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
```

නියැදි ධාවනය 2

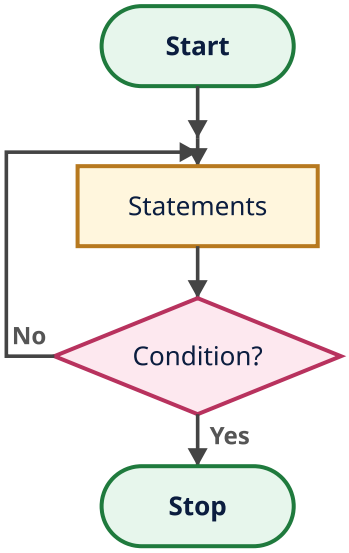
```
Enter mark (-1 to stop): -1
No marks entered
```

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

Count = 3, Total = 219, එබැවින් සාමාන්‍යය = 219 ÷ 3 = 73.00. පළමු ආදානය −1 නම් ලූපයේ ඇතුළත කිසි විටෙක ධාවනය නොවී “No marks entered” කියයි.

3. නොදන්නා වාර ගණන, අවසානයේ පරීක්ෂාව: REPEAT–UNTIL

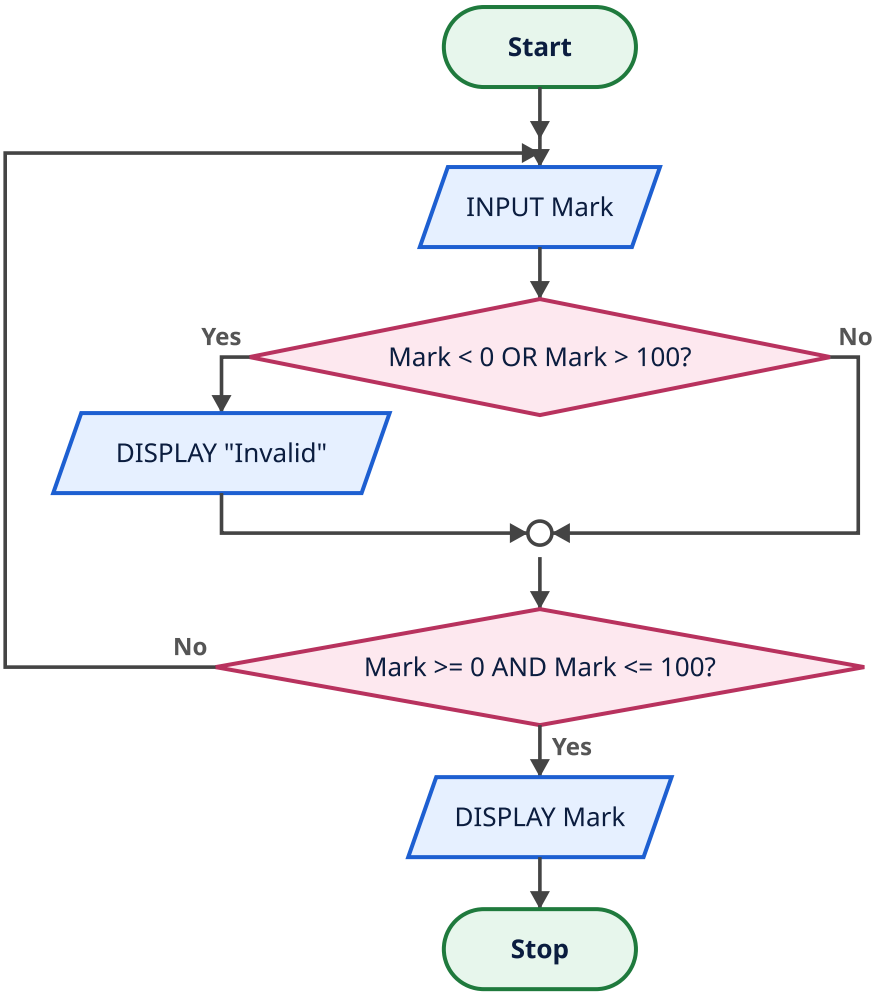
REPEAT–UNTIL ලූපයක් ඇතුළත **මුලින්** ධාවනය කර **අවසානයේදී** කොන්දේසිය පරීක්ෂා කරයි, එබැවින් ඇතුළත සැමවිටම **අවම වශයෙන් එක් වරක්** ධාවනය වේ. කොන්දේසිය **සත්‍ය** වූ විට නවතී. ආදාන පරීක්ෂාවට ඉතා හොඳයි: වලංගු ලකුණක් (0–100) ටයිප් කරන තුරු දිගටම ඉල්ලීම.



REPEAT–UNTIL ලූපයක සාමාන්‍ය ආකාරය (අවසානයේ පරීක්ෂාව)

ව්‍යාජ කේතය (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
```



ගැලීම් සටහන – Repeat–Until (අවසානයේ පරීක්ෂාව): වලංගු ලකුණක් ලබාගැනීම

Repeat-Until (අවසානයේ පරීක්ෂාව): වලංගු ලකුණක් ලබාගැනීම

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)
```

නියැදි ධාවනය

```
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 හි **REPEAT-UNTIL නැත**. while True: … සහ නැවතීමේ කොන්දේසිය ලැබුණු විට break භාවිතයෙන් එය අනුකරණය කරමු.

4. ලූප තුන සැසඳීම

	FOR	WHILE	REPEAT-UNTIL
වාර ගණන	දනී	නොදනී	නොදනී
කොන්දේසිය පරීක්ෂාව	ගණකය ආරම්භයේ	ආරම්භයේ	අවසානයේ
අවම ධාවන	0	0	1
නවතින්නේ	ගණකය අවසාන අගය ඉක්මවූ විට	කොන්දේසිය අසත්‍ය වූ විට	කොන්දේසිය සත්‍ය වූ විට
Pascal	for i := 1 to 10 do	while … do	repeat … until …;
Python	for i in range(1, 11):	while …:	while True: … break

5. එක් ලූපයක් තවත් එකකට පරිවර්තනය

එකම කාර්යය – 2, 4, 6, 8, 10 ඉරට්ටේ සංඛ්‍යා පෙන්වීම – එක් එක් ලූපයෙන් ලියා ඇත. FOR එකක් WHILE එකකට වෙනස් කිරීමට: (1) ලූපයට **පෙර** ගණකය සකසන්න, (2) කොන්දේසිය ලියන්න, (3) ලූපය **තුළ ගණකය වැඩි කරන්න**. REPEAT–UNTIL සඳහා කොන්දේසිය **ප්‍රතිවිරුද්ධ** කරන්න (WHILE i <= 10 වන්නේ UNTIL i > 10).

ව්‍යාජ කේතය (Pseudocode)

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

FOR මගින් 2–10 ඉරට්ටේ සංඛ්‍යා

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)
```

නියැදි ධාවනය

```
2
4
6
8
10
```

ව්‍යාජ කේතය (Pseudocode)

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

WHILE මගින් 2–10 ඉරට්ටේ සංඛ්‍යා

Pascal

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

Python

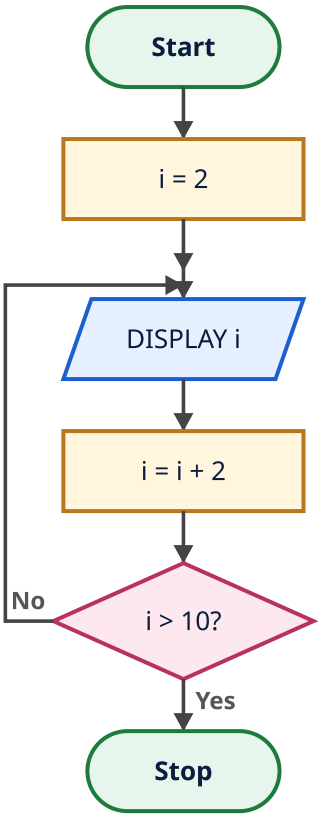
```
i = 2
while i <= 10:
    print(i)
    i = i + 2
```

නියැදි ධාවනය

```
2
4
6
8
10
```

ව්‍යාජ කේතය (Pseudocode)

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



ගැලීම් සටහන – REPEAT-UNTIL මගින් 2-10 ඉරට්ටේ සංඛ්‍යා

REPEAT-UNTIL මගින් 2-10 ඉරට්ටේ සංඛ්‍යා

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
```

නියැදි ධාවනය

```
2
4
6
8
10
```

💡 අනතුර – **අනන්ත ලූපය**: WHILE ලූපයක් තුළ i = i + 2 අමතක වුවහොත් i සදහටම 2 ලෙස පවතින අතර ලූපය කිසි විටෙක නොනවතී. ලූපය තුළ යමක් එය අවසානය දෙසට ගෙන යන බව සැමවිටම පරීක්ෂා කරන්න.

ප්‍රධාන පද

ලූපය – නැවත නැවත ක්‍රියාත්මක වන ප්‍රකාශ.

FOR ලූපය – ගණකයක් භාවිතයෙන් දන්නා වාර ගණනක් ධාවනය වන ලූපය.

WHILE ලූපය – ආරම්භයේදී කොන්දේසිය පරීක්ෂා කරන ලූපය.

REPEAT–UNTIL ලූපය – අවසානයේදී කොන්දේසිය පරීක්ෂා කරන ලූපය; අවම වශයෙන් එක් වරක් ධාවනය වේ.

ගණකය – ලූප ධාවන ගණන් කරන විචල්‍යයක්, උදා i.

පූර්ව-පරීක්ෂණ ලූපය – ඇතුළතට පෙර කොන්දේසිය පරීක්ෂාව (WHILE, FOR).

පශ්චාත්-පරීක්ෂණ ලූපය – ඇතුළතට පසු කොන්දේසිය පරීක්ෂාව (REPEAT–UNTIL).

Sentinel අගය – “නවතින්න” යන්න අදහස් කරන විශේෂ අගයක්, උදා −1.

අනන්ත ලූපය – කිසි විටෙක නොනවතින ලූපයක්.

වලංගුකරණය – ආදානය සාධාරණදැයි පරීක්ෂාව, උදා 0 සිට 100 දක්වා ලකුණක්.

range() – FOR ලූපයකට සංඛ්‍යා දෙන Python ශ්‍රිතය; range(1, 11) යනු 1 සිට 10.

break – ලූපයකින් වහාම පිටවන Python වචනය.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. FOR ලූපයක් භාවිත කරන්නේ කවදාද?

පිළිතුර: කී වරක් පුනරාවර්තනය කළ යුතුදැයි දන්නා විට.

2. WHILE සහ REPEAT–UNTIL අතර ප්‍රධාන වෙනස කුමක්ද?

පිළිතුර: WHILE ආරම්භයේ පරීක්ෂා කරයි (0 වරක් ධාවනය විය හැක); REPEAT–UNTIL අවසානයේ පරීක්ෂා කරයි (අවම වශයෙන් එක් වරක්).

3. Sentinel අගයක් යනු කුමක්ද?

පිළිතුර: ලූපයට නවතින ලෙස කියන −1 වැනි විශේෂ අගයක්.

4. 5 හි සාධකීය කුමක්ද?

පිළිතුර: 120.

5. for i := 1 to 10 කී වරක් ධාවනය වේද?

පිළිතුර: 10 වරක්.

6. අනන්ත ලූපයක් ඇති වන්නේ කුමක් නිසාද?

> **පිළිතුර:** කොන්දේසිය කිසි විටෙක වෙනස් නොවීම, උදා ගණකය වැඩි කිරීම අමතක වීම.

බහුවරණ ප්‍රශ්න

1. දන්නා වාර ගණනක් සඳහා ලූපය...

A. FOR
B. WHILE
C. IF
D. REPEAT–UNTIL

2. සැමවිටම අවම වශයෙන් එක් වරක් ධාවනය වන ලූපය කුමක්ද?

A. WHILE
B. FOR
C. කිසිවක් නැත
D. REPEAT–UNTIL

3. ආරම්භයේදී කොන්දේසිය පරීක්ෂා කරන ලූපය කුමක්ද?

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

4. REPEAT ... UNTIL කොන්දේසිය නවතින්නේ කොන්දේසිය...

A. ශූන්‍ය වූ විට
B. අසත්‍ය වූ විට
C. සත්‍ය වූ විට
D. 10 ට සමාන වූ විට

5. Python හි range(1, 11) දෙන්නේ...

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

6. N = 5 සඳහා සාධකීය වැඩසටහනේ ප්‍රතිදානය...

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

7. ලකුණු 70, 85, 64 පසුව −1 ඇතුළත් කරයි. පෙන්වන සාමාන්‍යය...

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

8. ලකුණු වැඩසටහනේ −1 හඳුන්වන්නේ...

A. දත්ත වර්ගයක්
B. ගණකයක්
C. නියත ලකුණක්
D. sentinel අගයක්

9. පළමු ආදානය −1 නම් WHILE ලූපයේ ඇතුළත ධාවනය වන්නේ...

A. 0 වරක්
B. සදහටම
C. 3 වරක්
D. 1 වරක්

10. Python හි REPEAT–UNTIL නැත. අප භාවිත කරන්නේ...

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

11. “WHILE i <= 10” REPEAT–UNTIL ට වෙනස් කළ විට...

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

12. WHILE ලූපයක් තුළ ගණකය වැඩි කිරීම අමතක වීමෙන් ඇති වන්නේ...

A. syntax දෝෂයක්
B. ප්‍රතිදානයක් නැත
C. අනන්ත ලූපයක්
D. වේගවත් වැඩසටහනක්

13. ගුණන වගු වැඩසටහන ආදානයෙන් පසු පේළි කීයක් මුද්‍රණය කරයි?

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

14. වලංගු ලකුණක් ටයිප් කරන තුරු නැවත නැවත ඉල්ලීමට වඩාත් සුදුසු...

A. FOR
B. නියතයක්
C. CASE
D. REPEAT–UNTIL

15. for i := 1 to 5 do write(i * 2) මුද්‍රණය කරන්නේ...

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

ව්‍යුහගත ප්‍රශ්න

1. (i) පූර්ව-පරීක්ෂණ සහ පශ්චාත්-පරීක්ෂණ ලූප අතර එක් වෙනසක් දෙන්න. (ලකුණු 2)
(ii) Pascal හි එක් එකකට උදාහරණයක් දෙන්න. (ලකුණු 2) *(ලකුණු 4)*

> (i) පූර්ව-පරීක්ෂණ ලූපය ඇතුළතට පෙර පරීක්ෂා කරන අතර 0 වරක් ධාවනය විය හැක; පශ්චාත්-පරීක්ෂණ ලූපය පසුව පරීක්ෂා කර අවම වශයෙන් එක් වරක් ධාවනය වේ. (ii) while ... do (පූර්ව); repeat ... until (පශ්චාත්).

2. සංඛ්‍යාවක් ආදානය කර එහි ගුණන වගුව 1 සිට 10 දක්වා පෙන්වන වැඩසටහනක් (Pascal හෝ Python) ලියන්න. (ලකුණු 4) *(ලකුණු 4)*

N ආදානය (1); FOR i = 1 සිට 10 (1); N × i ගණනය (1); 7 x 1 = 7 ආකාරයෙන් පෙන්වීම (1).

For loop (දන්නා වාර ගණන): ගුණන වගුව

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)
```

නියැදි ධාවනය

```
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
```

3. −1 ඇතුළත් කරන තුරු ලකුණු කියවා, පසුව ගණන, එකතුව සහ සාමාන්‍යය පෙන්වන ව්‍යාජ කේතය ලියන්න. (ලකුණු 5) *(ලකුණු 5)*

Count = 0, Total = 0 (1); ලූපයට පෙර Mark ආදානය (1); WHILE Mark <> −1 Count සහ Total ට එකතු කිරීම (1); ලූපය තුළ Mark ආදානය (1); Count, Total, Total / Count පෙන්වීම (1).

ව්‍යාජ කේතය (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. මෙම FOR ලූපය WHILE ලූපයක් ලෙස නැවත ලියන්න: FOR i = 1 TO 5 : DISPLAY i * 2 : ENDFOR. (ලකුණු 3) *(ලකුණු 3)*

i = 1 (1); WHILE i <= 5 DO DISPLAY i * 2 (1); i = i + 1; ENDWHILE (1). (නැතහොත් i = 2 … i <= 10 … i = i + 2 සමාන ආකාරය.)

ව්‍යාජ කේතය (Pseudocode)

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

5. N = 5 සමඟ සාධකීය වැඩසටහන සඳහා අනුරේඛන වගුවක් සම්පූර්ණ කර ප්‍රතිදානය දෙන්න. (ලකුණු 3) *(ලකුණු 3)*

i = 1 සිට 5 සඳහා F = 1, 2, 6, 24, 120 (2); ප්‍රතිදානය Factorial = 120 (1).

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

සාරාංශය

- FOR: දන්නා වාර ගණන.
- WHILE: ආරම්භයේ පරීක්ෂාව; 0 වරක් ධාවනය විය හැක.
- REPEAT–UNTIL: අවසානයේ පරීක්ෂාව; අවම වශයෙන් එක් වරක්.
- REPEAT නවතින්නේ කොන්දේසිය සත්‍ය වූ විටය.
- Sentinel (−1) ලූපයට නවතින ලෙස කියයි.
- Python: for … in range(); while; while True + break.
- පරිවර්තනය: ගණකය සකසන්න, පරීක්ෂා කරන්න, තුළ වැඩි කරන්න.
- ලූපය අවසානය දෙසට ගෙන යන්න – අනන්ත ලූප වළකින්න.

MCQ පිළිතුරු – 1.7

1. **A** – දන්නා ගණන → FOR.
2. **D** – අවසානයේ පරීක්ෂාව.
3. **B** – පූර්ව-පරීක්ෂණ.
4. **C** – සත්‍ය වන තුරු.
5. **A** – අවසානය ඇතුළත් නැත.
6. **B** – 1×2×3×4×5.
7. **C** – 219 ÷ 3.
8. **D** – නැවතීමේ සංඥාව.
9. **A** – ආරම්භයේදීම අසත්‍යයි.
10. **A** – break සමඟ while True.
11. **C** – කොන්දේසිය ප්‍රතිවිරුද්ධ කරන්න.
12. **C** – නොනවතී.
13. **B** – i = 1 සිට 10.
14. **D** – අවම වශයෙන් එක් වරක් ඉල්ලිය යුතුය.
15. **B** – 1–5 දෙගුණ.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.