සියලු ම හිමිකම් ඇවිරිණි / *All Rights Reserved*]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 285 MARKING

2025/09/21
ictfromabc
Ravindu Bandaranayake

1. Which type of software is distributed for free but may require payment for additional features or functionalities?
නොමිලේ බෙදා හරින නමුත් අමතර විශේෂාංග හෝ ක්‍රියාකාරීත්වයන් සඳහා ගෙවීමක් අවශ්‍ය වන්නේ කුමන මෘදුකාංග වර්ගයටද?

   1) <u>Shareware</u>
   2) Freeware
   3) Open-source software
   4) Closed-source software
   5) Proprietary software

Answer :1)

The type of software that is distributed for free but may require payment for additional features or functionalities is Shareware. Shareware is freely distributed to users on a trial basis, but it usually comes with a time limit, and when the time limit expires, the user is asked to pay for continued services.
නොමිලේ බෙදාහරින නමුත් අමතර විශේෂාංග හෝ ක්‍රියාකාරීත්වයන් සඳහා ගෙවීමක් අවශ්‍ය විය හැකි මෘදුකාංග වර්ගය Shareware වේ. Shareware පරිශීලකයන්ට අත්හදා බැලීමේ පදනමක් මත නොමිලේ බෙදා හරිනු ලැබේ, නමුත් එය සාමාන්‍යයෙන් කාල සීමාවක් සමඟ පැමිණෙන අතර කාල සීමාව කල් ඉකුත් වූ විට, අඛණ්ඩ සේවා සඳහා ගෙවීමට පරිශීලකයාගෙන් ඉල්ලා සිටී.

Freeware refers to software that is made available for use at no cost. Unlike shareware, which typically has limitations or requires payment for full functionality, freeware is fully functional without any payment required.
Freeware යනු කිසිදු වියදමකින් තොරව භාවිතය සඳහා ලබා ගත හැකි මෘදුකාංග වේ. සාමාන්‍යයෙන් සීමාවන් ඇති හෝ සම්පූර්ණ ක්‍රියාකාරීත්වය සඳහා ගෙවීමක් අවශ්‍ය වන shareware මෙන් නොව, කිසිදු ගෙවීමක් අවශ්‍ය නොවීම නොමිලේ ක්‍රියාත්මක වේ.

Open-source software (OSS) refers to software whose source code is made available to the public for use, modification, and distribution under a license that allows anyone to view, modify, and distribute the software freely.
විවෘත-මූලාශ්‍ර මෘදුකාංග (OSS) යනු ඕනෑම කෙනෙකුට නිදහසේ මෘදුකාංග බැලීමට, වෙනස් කිරීමට සහ බෙදා හැරීමට ඉඩ සලසන බලපත්‍රයක් යටතේ භාවිත කිරීම, වෙනස් කිරීම සහ බෙදා හැරීම සඳහා මහජනයාට ලබා ගත හැකි ප්‍රභව කේතය සහිත මෘදුකාංගයකි.

Closed-source software, also known as proprietary software, refers to software whose source code is not available to the public for viewing, modification, or distribution. Instead, the software is typically developed and owned by a single entity, which retains control over its distribution and licensing.
සංවෘත-මූලාශ්‍ර මෘදුකාංග, හිමිකාර මෘදුකාංග ලෙසද හැඳින්වේ, මහජනයාට නැරඹීම, වෙනස් කිරීම හෝ බෙදා හැරීම සඳහා මූලාශ්‍ර කේතය ලබා ගත නොහැකි මෘදුකාංග වේ. ඒ වෙනුවට, මෘදුකාංගය සාමාන්‍යයෙන් සංවර්ධනය කර එහි බෙදාහැරීම සහ බලපත්‍ර ලබා දීම පාලනය කරන තනි ආයතනයක් සතු වේ.

Proprietary software, also known as closed-source software, refers to software that is owned by a specific individual or organization and is protected by copyright, patents, or other forms of intellectual property rights. This type of software is not freely available for modification, redistribution, or inspection by the general public.
හිමිකාර මෘදුකාංග, සංවෘත මූලාශ්‍ර මෘදුකාංග ලෙසද හැඳින්වේ, විශේෂිත පුද්ගලයෙකුට හෝ සංවිධානයකට අයත් වන අතර ප්‍රකාශන හිමිකම, පේටන්ට් බලපත්‍ර හෝ වෙනත් ආකාරයේ බුද්ධිමය දේපල අයිතිවාසිකම් මගින් ආරක්ෂා කර ඇති මෘදුකාංග වේ. මෙම වර්ගයේ මෘදුකාංග සාමාන්‍ය ජනතාව විසින් වෙනස් කිරීම, නැවත බෙදා හැරීම හෝ පරීක්ෂා කිරීම සඳහා නොමිලේ ලබා ගත නොහැක.

2. Among the methods of data validation, which is the check that all the data is filled correctly?
දත්ත වලංගු කිරීමේ ක්‍රම අතුරින් සියලුම දත්ත නිවැරදිව පිරවී ඇත්දැයි සිදු කරන පරීක්ෂාව වන්නේ කුමක්ද?

1) Type check 2) Presence check 3) Range check 4) Format check 5) Code check

Answer:2)

Type Check
Checks whether the correct data type is entered (Text, Number, Date)
නිවැරදි දත්ත වර්ගය ඇතුළත් වී ඇත්දැයි පරීක්ෂා කිරීම සිදු කරයි.

Presence check
Checks whether the data fields are properly filled. If important data fields are empty, it affects the future process.
සියලුම දත්ත නිවැරදිව පිර වී ඇති දැයි පරීක්ෂා කිරීම මෙහිදී සිදුවේ. හිස් දත්ත ජේළි තිබීම දත්ත සැකසීමේ ක්‍රියාවලියට බලපෑ හැක.

Range check
Range check is used when data is entered as numbers. It checks whether the numbers are placed between a proper range according to the expected data.
ඇතුළත් කරනු ලබන දත්ත අංක වන අවස්ථාවේ දී මෙය යොදා ගැනේ. මෙමගින් ඇතුළත් වන අංක නිවැරදි පරාසයක් අතර තිබේදැයි පරීක්ෂා කරයි. මෙම පරාසය බලාපොරොත්තු වන දත්ත වලට අනුකූල විය යුතු ය.

Format check
This is used to ensure that the entered data is in the right format
ඇතුළත් කරන දත්ත නිවැරදි ආකෘතියට ඇත්දැයි මෙමගින් පරීක්ෂා කරයි.

3. If $23_8 + x > 100001_2$, which of the following can be x?
$23_8 + x > 100001_2$ නම්, පහත ඒවායින් x විය හැක්කේ කුමක් ද?

A. 1011_2 B. 15_{16} C. 16_8 D. 20_{10}

1) B only. B පමණි.
2) A and C only. A සහ C පමණි.
3) B and D only. B සහ D පමණි.
4) B, C and D only. B, C සහ D පමණි.
5) All of the above. ඉහත සියල්ලම.

Answer: 3)

Let's first convert all the numbers given in the above question into decimals.
ඉහත ප්‍රශ්නයෙහි දැක්වෙන සියළු සංඛ්‍යා දශමය බවට පරිවර්ථනය කරමු.

- **$23_8 + x > 100001_2$**

23_8	
2	3
8^1	8^0
16	3
16 + 3	
19_{10}	

100001_2					
1	0	0	0	0	1
2^5	2^4	2^3	2^2	2^1	2^0
32	-	-	-	-	1
32 + 1					
33_{10}					

- **$19_{10} + x > 33_{10}$**

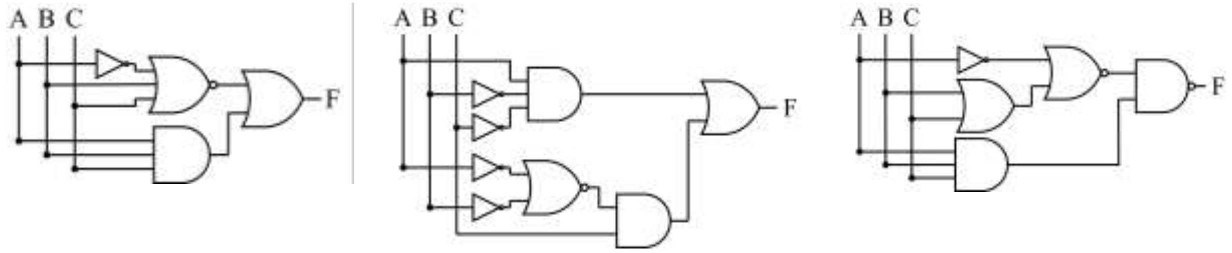
A			
1011_2			
1	0	1	1
2^3	2^2	2^1	2^0
8	-	2	1
8 + 2 + 1			
11_{10}			

B	
15_{16}	
1	5
16^1	16^0
16	5
16 + 5	
21_{10}	

C	
16_8	
1	6
8^1	8^0
8	6
8 + 6	
14_{10}	

Therefore the correct answer is **3).**
එබැවින් නිවැරදි පිළිතුර **3)** වේ.

4. Which of the following circuits does the same logic apply?
පහත පරිපථ අතරින් එක සමාන තර්කයක් ක්‍රියාත්මක වන්නේ කුමන පරිපථ අතරද?



A B C

1) <u>A and B only.</u> A හා B පමණි.
2) A only. A පමණි.
3) B only. B පමණි.
4) B and C only. B හා C පමණි.
5) All of the above. ඉහත සියල්ලම.

Answer : 1)

Let's get the answer below. / පහත පරිදි පිළිතුර ලබාගනිමු.

A. $\overline{(\overline{A}+B+C)}+(ABC)$

$\overline{(\overline{A}+B+C)}+(ABC)$	
$\overline{\overline{A}}\overline{B}\overline{C}+ABC$	De morgan's Law ද මුවාර් න්‍යාය
$A\overline{B}\overline{C}+ABC$	Double Inverse Law ද්විත්ව ප්‍රතිලෝම න්‍යාය

B. $(A\overline{B}\overline{C})+(\overline{\overline{A}+\overline{B}})C$

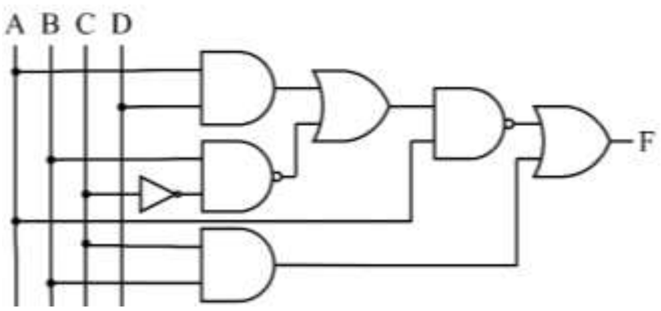
$(A\overline{B}\overline{C})+(\overline{\overline{A}+\overline{B}})C$	
$A\overline{B}\overline{C}+\overline{\overline{A}}\,\overline{\overline{B}}C$	De morgan's Law ද මුවාර් න්‍යාය
$A\overline{B}\overline{C}+A\overline{\overline{B}}C$	Double Inverse Law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$A\overline{B}\overline{C}+ABC$	Double Inverse Law ද්විත්ව ප්‍රතිලෝම න්‍යාය

C. $\overline{\overline{(\overline{A}+B+C)}(ABC)}$

$\overline{\overline{(\overline{A}+B+C)}(ABC)}$	
$\overline{\overline{\overline{A}+B+C}}+\overline{A}+\overline{B}+\overline{C}$	De morgan's Law ද මුවාර් න්‍යාය
$\overline{A}+B+C+\overline{A}+\overline{B}+\overline{C}$	Double Inverse Law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$\overline{A}+C+\overline{A}+\overline{C}+1$	Inverse Law ප්‍රතිලෝම න්‍යාය
1	Identity Law සර්වසාම්‍ය න්‍යාය

So, the answer is 1). / එබැවින්, පිළිතුර 1) වේ.

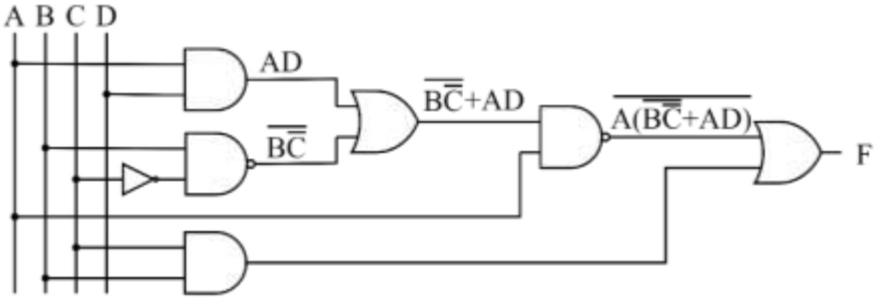
5. What Boolean expression can be obtained from the below logic circuit?
පහත තාර්කික පරිපථයෙන් ලබාගත හැකි බූලිය ප්‍රකාශනය කුමක්ද?



1) $\overline{A}+B\overline{C}\overline{D}$
2) $\overline{A}+B\overline{C}+BD$
3) <u>$\overline{A}+BC+B\overline{D}$</u>
4) $\overline{A}+\overline{BC}+B\overline{D}$
5) $\overline{A}+\overline{BC}+\overline{B}\overline{D}$

Answer : 3)

Let's obtained an expression from the above logical circuit.
ඉහත තාර්කික පරිපථයෙන් ප්‍රකාශනයක් ලබා ගනිමු.



$F = (B.C) + \overline{(\overline{\overline{B\overline{C}}} + AD)A}$	
$F = (B.C) + \overline{(\overline{\overline{B\overline{C}}} + AD)} + \overline{A}$	De morgan's Law ද මුවාර් න්‍යාය
$F = (B.C) + (\overline{\overline{B\overline{C}}}.\overline{AD}) + \overline{A}$	De morgan's Law ද මුවාර් න්‍යාය
$F = (B.C) + (B\overline{C}.\overline{AD}) + \overline{A}$	Double complement Law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$F = (B.C) + (B\overline{C}.(\overline{A} + \overline{D})) + \overline{A}$	De morgan's Law ඩි මෝගන් න්‍යාය
$F = (B.C) + (B\overline{C}\overline{A} + B\overline{C}\overline{D}) + \overline{A}$	Distributive Law විඝටන න්‍යාය
$F = (B.C) + B\overline{C}\overline{A} + \overline{A} + B\overline{C}\overline{D}$	Commutative Law න්‍යාදේශ න්‍යාය
$F = (B.C) + \overline{A}(B\overline{C} + 1) + B\overline{C}\overline{D}$	Distributive Law විඝටන න්‍යාය
$F = (B.C) + \overline{A}(1) + B\overline{C}\overline{D}$	Identity Law සර්වසාම්‍ය න්‍යාය
$F = (B.C) + \overline{A} + B\overline{C}\overline{D}$	Identity Law සර්වසාම්‍ය න්‍යාය
$F = (B.C) + B\overline{C}\overline{D} + \overline{A}$	Commutative Law න්‍යාදේශ න්‍යාය
$F = B(C + \overline{C}\overline{D}) + \overline{A}$	Distributive Law විඝටන න්‍යාය
$F = B(C + \overline{D}) + \overline{A}$	Redundancy Law සමතිරික්ත න්‍යාය
$F = BC + B\overline{D} + \overline{A}$	Distributive Law විඝටන න්‍යාය

So, the correct answer is 3).
එබැවින්, නිවැරදි පිළිතුර 3) වේ.

6. Which of the following Python code(s) would compile without generating an error?
පහත සඳහන් කුමන පයිතන් කේත(ය) දෝෂයක් ජනනය නොකර සම්පාදනය කරයිද?

A)
```
number = (100, 200, 300, 400, 500)
number.pop(2)
print(number)
```

B)
```
number = (100,)
print(number * 2)
```

C)
```
number = {"name": "Mike", "salary": 8000}
variable = number.pop("age")
print(variable)
```

1) A only. A පමණි.
2) <u>B only.</u> B පමණි.
3) C only. C පමණි.
4) B and C only. B හා C පමණි.
5) A and C only. A හා C පමණි.

Answer : 2)

Let's consider the explanation of each code.
එක් එක් කේතයේ පැහැදිලි කිරීම සලකා බලමු.

A)
```
number = (100, 200, 300, 400, 500)
number.pop(2)
print(number)
```

Tuples are immutable in Python, meaning you cannot modify them. The `pop()` method is used to remove elements from lists, not tuples. This will raise an error.
පයිතන්හි Tuple, immutable වේ, එනම් ඔබට ඒවා වෙනස් කළ නොහැක. `pop()` method භාවිතා කරනුයේ ලැයිස්තු වලින් මූලාංග ඉවත් කිරීමට මිස tuple වලින් නොවේ. මෙය දෝෂයක් මතු කරයි.

```
B) number = (100,)
   print(number * 2)
```

This creates a tuple with one element, and multiplying the tuple by 2 will repeat the contents, producing `(100, 100)`. This code will compile and run without errors.
මෙය එක් මූලාංගයක් සහිත tuple එකක් සාදන අතර, tuple එක 2 කින් ගුණ කිරීමෙන් අන්තර්ගතය නැවත නැවත සිදු කරන අතර `(100, 100)` නිපදවයි. මෙම කේතය සම්පාදනය කර දෝෂයකින් තොරව ක්‍රියාත්මක වේ.

```
C) number = {"name": "Mike", "salary": 8000}
   variable = number.pop("age")
   print(variable)
```

The `pop()` method tries to remove and return the value associated with the key "age", but since "age" doesn't exist in the dictionary, it will raise a KeyError.
`pop()` method "age" යතුර හා සම්බන්ධ අගය ඉවත් කර ආපසු ලබා දීමට උත්සාහ කරයි, නමුත් dictionary එකේ "age" නොපවතින බැවින්, එය KeyError මතු කරයි.

B will compile without errors, while A and C will produce errors.
B දෝෂයකින් තොරව සම්පාදනය කරන අතර A සහ C දෝෂ ඇති කරයි.

The correct option is 2) B only.
නිවැරදි විකල්පය වන්නේ 2) B පමණි.

7. What will be the output of the following python operations?
පහත සඳහන් පයිතන් මෙහෙයුම් වල ප්‍රතිදානය කුමක්ද?

```
print(12 & 5)
print(12 | 5)
```

1)	2)	3)	4)	5)
4 13	8 17	12 5	0 1	9 16

Answer : 5)

Answer 5), all the other answer will increase beyond 5 so the loop will stop thereafter, but in answer 5 though the function keeps increasing it will reach a point where x will be 2 here onwards x will be always 2 because when you divide by 2 the answer will be 1 and after when you add 1 the answer will be 1, so after x reach 2 in the 5th function it will remain 2 making the loop execute infinite number of times.
පිළිතුර 5) වේ, අනෙක් සියලුම පිළිතුරු 5 න් ඔබ්බට වැඩි වන නිසා ලූප් එක නතර වනු ඇත, නමුත් පිළිතුර 5 හි ශ්‍රිතය වැඩිවෙමින් පැවතුනද, එය x 2 වන ස්ථානයට ළඟා වේ, මන්ද, ඔබ බෙදූ විට x සැමවිටම 2 වනු ඇත. 2 පිළිතුර 1 වන අතර ඔබ 1 එකතු කළ පසු පිළිතුර 1 වනු ඇත, එබැවින් 5 වන ශ්‍රිතයේ x 2 ට ළඟා වූ පසු එය 2 ලෙස පවතිනු ඇති අතර එමඟින් ලූපය අනන්ත වාර ගණනක් ක්‍රියාත්මක වේ.

8. What is the output of the following python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
true=12.0//1
y=true+True
print(y)
```

1) True 2) 12 3) 12.0 4) 13.0 5) syntax error

Answer : 4)

The output of the Python code is `13.0`. In this code, `true = 12.0 // 1` sets the variable `true` to `12.0`, as the `//` operator performs floor division, which rounds down the result to the nearest whole number (but keeps it as a float). Then, `y = true + True` adds `true` (which is 12.0) to `True`. In Python, `True` is treated as `1`, so the expression becomes `y = 12.0 + 1`, which results in 13.0. Therefore, the `print(y)` statement outputs 13.0.
පයිතන් කේතයේ ප්‍රතිදානය `13.0` වේ. මෙම කේතයෙහි, `//` ක්‍රියාකරු බිම් බෙදීම සිදු කරන බැවින්, `true = 12.0 // 1` විචල්‍යය `true 12.0` ලෙස සකසයි, එය ප්‍රතිඵලය ආසන්නතම සම්පූර්ණ සංඛ්‍යාවට වට කරයි (නමුත් එය float ලෙස තබා ගනී) . එවිට, `y = true + True` වෙත `true` (එය `12.0`) එකතු කරයි. Python හි, `True 1` ලෙස සලකනු ලැබේ, එබැවින් ප්‍රකාශනය `y = 12.0 + 1` බවට පත් වේ, එහි ප්‍රතිඵලය `13.0` වේ. එබැවින්, `print(y)` ප්‍රකාශය 13.0 ප්‍රතිදානය කරයි.

So, the correct answer is 4. / එබැවින්, නිවැරදි පිළිතුර 4 වේ.

9. What is the output of following python code?
පහත පයිතන් ක්‍රමලේඛයේ ප්‍රතිදානය කුමක්ද?

```
my_list = [10, 20, 30, 40, 50, 60, 70]
sliced_list = my_list[1:6:2]
result = sliced_list[1] + my_list[-3]
print(result)
```

1) `50` 2) `60` 3) `70` 4) `80` 5) `90`

Answer : 5)

Here you should know the concept of slicing.
මෙහිදී ඔබ විසින් slicing සංකල්පය දැන සිටිය යුතුය.

And what has happened in the code is further explained by the following table.
එමෙන්ම මෙහිදී කේතයේ සිදු වී ඇති දෑ ඔබට පහත වගුව මගින් වැඩිදුරටත් පැහැදිලි වේ.

Step	Variable	Value
1	my_list	[10, 20, 30, 40, 50, 60, 70]
2	sliced_list	[20, 40, 60]
3	sliced_list[1]	40
4	my_list[-3]	50
5	result	40 + 50
6	result	90
7	print(result)	90 (output)

Accordingly, the output here is 90. So the answer number is 5.
ඒ අනුව මෙහිදී ප්‍රතිදානය වන්නේ 90 වේ. එම නිසා පිළිතුරු අංකය වන්නේ 5 වේ.

10. What is the output of the following Python code when 8 is given as input?
පහත පයිතන් කේතයෙහි ආදානය ලෙස 8 ලබා දුන් විට ප්‍රතිදානය කුමක්ද?

```
a=int(input("enter value:"))
if a > 0:
      n="positive"
else:
      n="not positive"
print("Number is",n)
```

1) `8`
2) `Number is negative`
3) `Number is non positive`
4) `Number is positive`
5) `Number is 5`

Answer : 4)

The output of the Python code when 8 is given as input is 4) `Number is positive`.
8 ආදානය ලෙස ලබා දුන් විට පයිතන් කේතයේ ප්‍රතිදානය 4) `Number is positive` වේ.

`a = int(input("enter value:"))`
This line takes input from the user and converts it to an integer, storing it in the variable `a`.
මෙම පේළිය පරිශීලකයාගෙන් ආදානය ලබාගෙන එය පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය කරයි, එය `a` විචල්‍යයේ ගබඩා කරයි.

`if a > 0:`
This checks if the value of `a` is greater than `0`. Since `8` is greater than `0`, the condition is true.
මෙය `a` හි අගය `0` ට වඩා වැඩි දැයි පරීක්ෂා කරයි. `8`, `0` ට වඩා වැඩි බැවින්, කොන්දේසිය සත්‍ය වේ.

`n = "positive"`
This line sets the value of the variable `n` to "`positive`" because the condition in the if statement was true.
මෙම පේළිය `n` විචල්‍යයේ අගය `"positive"` ලෙස සකසන්නේ if ප්‍රකාශයේ කොන්දේසිය සත්‍ය වූ බැවිනි.

`else:`
This part of the code is executed if the condition in the if statement is false. Since it's true, this part is skipped.
if ප්‍රකාශයේ කොන්දේසිය අසත්‍ය නම් කේතයේ මෙම කොටස ක්‍රියාත්මක වේ. ඒක සත්‍ය බැවින් මේ කොටස නොසලකා හරී.

`print("Number is",n)`
This line prints the message "`Number is`" followed by the value of the variable `n`, which is `positive`.
මෙම පේළිය මගින් `"Number is"` යන පණිවිඩය මුද්‍රණය කර පසුව `n` විචල්‍යයේ අගය `positive` ලෙස මුද්‍රණය කරයි.

Therefore, the output of the code is "`Number is positive`".
එබැවින්, කේතයේ ප්‍රතිදානය `"Number is positive"` වේ.