සියලු ම හිමිකම් ඇවිරිණි /*All Rights Reserved*]

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

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 473 MARKING

2026/03/28
ictfromabc
Ravindu Bandaranayake

1. What modern output device, introduced in the early 1970s, revolutionized personal computing by providing a visual interface for interacting with computers?
පරිගණක සමඟ අන්තර්ක්‍රියා කිරීම සඳහා දෘශ්‍ය අතුරු මුහුණතක් ලබා දීම මගින් පරිගණක වල විප්ලවීය වෙනසක් ඇති කිරීමට 1970 කාලයේ මුල් භාගයේදී හඳුන්වා දුන් කුමන නිමැවුම් උපාංගය හේතු වූයේද?

   1) Printer 2) Mouse 3) Scanner 4) Speakers 5) Monitor

   Answer: 5)

   Explanation / පැහැදිලි කිරීම:

   - **Revolution in Personal Computing / පුද්ගලික පරිගණකකරණයේ විප්ලවය**

   In the early 1970s, personal computing underwent a significant transformation with the introduction of the monitor as a modern output device. This device played a pivotal role in making computers more accessible and user-friendly for individuals. Unlike earlier computing systems, which often relied on punch cards, printed output, or simple LED displays, monitors provided a visual interface that allowed users to interact with their computers more intuitively.
   1970 ගණන්වල මුල් භාගයේදී, නවීන නිමැවුම් උපාංගයක් ලෙස සංදර්ශකය හඳුන්වාදීමත් සමඟ පුද්ගලික පරිගණක සැලකිය යුතු පරිවර්තනයකට ලක් විය. මෙම උපාංගය පුද්ගලයන්ට පරිගණක වඩාත් ප්‍රවේශ විය හැකි සහ පරිශීලක-හිතකාමී බවට පත්කිරීමේදී ප්‍රධාන කාර්යභාරයක් ඉටු කළේය. බොහෝ විට සිදුරුපත්, මුද්‍රිත ප්‍රතිදානය හෝ සරල LED සංදර්ශක මත රඳා පවතින පෙර පරිගණක පද්ධති මෙන් නොව, සංදර්ශක මඟින් පරිශීලකයින්ට ඔවුන්ගේ පරිගණක සමඟ වඩාත් බුද්ධිමත්ව අන්තර් ක්‍රියා කිරීමට ඉඩ සලසන දෘශ්‍ය අතුරු මුහුණතක් ලබා දී ඇත.

   - **Importance of Visual Interface / දෘශ්‍ය අතුරුමුහුණතේ වැදගත්කම**

   The visual interface provided by monitors enabled users to see text and graphics directly on the screen, facilitating a more immediate and engaging interaction with the computer. This shift from non-visual to visual interaction was revolutionary, as it allowed users to receive instant feedback from their commands and see the results of their work in real time. This capability was crucial in making personal computers practical and appealing to a broader audience.
   සංදර්ශක මගින් සපයන ලද දෘශ්‍ය අතුරුමුහුණත මඟින් පරිශීලකයින්ට පාඨ සහ චිත්‍රක තිරය මත කෙලින්ම දැකීමට හැකි වූ අතර, පරිගණකය සමඟ වඩාත් ක්ෂණික සහ ආකර්ෂණීය අන්තර්ක්‍රියා සඳහා පහසුකම් සපයයි. දෘශ්‍ය නොවන සිට දෘශ්‍ය අන්තර්ක්‍රියාව දක්වා මෙම මාරුව විප්ලවීය විය, එය පරිශීලකයින්ට ඔවුන්ගේ විධානවලින් ක්ෂණික ප්‍රතිපෝෂණ ලබා ගැනීමට සහ ඔවුන්ගේ කාර්යයේ ප්‍රතිඵල තත්‍ය කාලීනව දැකීමට ඉඩ සලසන බැවිනි. පුද්ගලික පරිගණක ප්‍රායෝගික කිරීමට සහ පුළුල් ප්‍රේක්ෂක පිරිසක් අතර ප්‍රචලිත කිරීමට මෙම හැකියාව ඉතා වැදගත් විය.

   - **Comparison with Other Devices / අනෙකුත් උපාංග සමඟ සැසඳීම**

   While printers, mice, scanners, and speakers all contributed to the evolution of computing, they did not provide the same level of impact on user interaction as the monitor did. Printers produced hard copies of documents but lacked interactivity. The mouse, introduced around the same era, enhanced navigation but relied on the monitor to display graphical elements. Scanners digitized physical documents but were not interactive. Speakers added sound but did not offer visual feedback.
   මුද්‍රණ යන්ත්‍ර, මූසික, සුපිරික්සක සහ ස්පීකර් යන සියල්ල පරිගණනයේ පරිණාමය සඳහා දායක වූ අතර, ඒවා සංදර්ශකය මෙන් පරිශීලක අන්තර්ක්‍රියාවලට සමාන මට්ටමේ බලපෑමක් ලබා දුන්නේ නැත. මුද්‍රණ යන්ත්‍ර මගින් ලේඛනවල දෘඪ පිටපත් නිපදවන ලද නමුත් අන්තර් ක්‍රියාකාරීත්වයක් නොතිබුණි. එම යුගයේදීම හඳුන්වා දුන් මූසිකය සංචාලනය වැඩිදියුණු කළ නමුත් චිත්‍රක මූලද්‍රව්‍ය පෙන්වීමට සංදර්ශකය මත රඳා පැවතුනි. සුපිරික්සක භෞතික ලේඛන අංකිත කළ නමුත් අන්තර්ක්‍රියාකාරී නොවීය. ස්පීකර් ශබ්දය එක් කළ නමුත් දෘශ්‍ය ප්‍රතිපෝෂණ ලබා දුන්නේ නැත.

- **Impact on Personal Computing / පුද්ගලික පරිගණකකරණයට බලපෑම්**

The introduction of the monitor set the stage for the development of graphical user interfaces (GUIs), which further revolutionized personal computing. With GUIs, users could interact with their computers using icons, windows, and menus, all visually represented on the monitor. This development significantly lowered the barrier to entry for using computers, making them accessible to people without specialized technical knowledge.
සංදර්ශකය හඳුන්වාදීම මගින් චිත්‍රක පරිශීලක අතුරුමුහුණත් (GUI) සංවර්ධනය සඳහා වේදිකාව සකසා ඇති අතර එය පුද්ගලික පරිගණකකරණයේ තවත් විප්ලවීය වෙනසක් ඇති කළේය. GUIs සමඟින්, පරිශීලකයින්ට අයිකන, කවුළු සහ මෙනු භාවිතයෙන් ඔවුන්ගේ පරිගණක සමඟ අන්තර් ක්‍රියා කළ හැකි අතර, ඒවා සියල්ලම සංදර්ශකයේ දෘශ්‍ය ලෙස නිරූපණය කෙරේ. විශේෂිත තාක්ෂණික දැනුමක් නොමැති පුද්ගලයින්ට ඒවා ප්‍රවේශ විය හැකි පරිදි පරිගණක භාවිතය සඳහා ඇතුල්වීමේ බාධාව මෙම වර්ධනය මගින් සැලකිය යුතු ලෙස අඩු කළේය.

So, the monitor's ability to provide a visual interface for interacting with computers was a groundbreaking advancement in the early 1970s that greatly contributed to the widespread adoption and usability of personal computers.
එබැවින්, පරිගණක සමඟ අන්තර්ක්‍රියා කිරීම සඳහා දෘශ්‍ය අතුරු මුහුණතක් සැපයීමට සංදර්ශකයට ඇති හැකියාව 1970 දශකයේ මුල් භාගයේ පෙරළිකාර දියුණුවක් වූ අතර එය පුද්ගල පරිගණක පුළුල් ලෙස භාවිතා කිරීමට සහ භාවිතයට බෙහෙවින් දායක විය.

Therefore, the correct answer is 5).
එබැවින්, නිවැරදි පිළිතුර 5) වේ.

2. Among these, the number of bits possible for Sign, Exponent, and Mantissa in IEEE 754 - Double Precision is,
මේ අතරින් IEEE 754 - Double Precision හි Sign, Exponent සහ Mantissa සඳහා තිබිය හැකි බිටු ගණන වන්නේ,

1) 1, 11, 32 2) 1, 7, 24 3) 1, 8, 23 4) 1, 11, 52 5) 1, 12, 33

Answer: 4)

The correct answer is 4) 1, 11, 52. / නිවැරදි පිළිතුර 4) 1, 11, 52 වේ.

IEEE 754 Double Precision Format / IEEE 754 ද්විත්ව නිරවද්‍යතා ආකෘතිය

IEEE 754 double precision format, also known as binary64, represents floating-point numbers using three components: a sign bit, an exponent, and a mantissa (also known as the significand). The total number of bits used is 64
IEEE 754 ද්විත්ව නිරවද්‍යතා ආකෘතිය, binary64 ලෙසද හැඳින්වේ, සංරචක තුනක් භාවිතා කරමින් floating-point සංඛ්‍යා නියෝජනය කරයි: sign bit, exponent සහ mantissa (significand) ලෙසද හැඳින්වේ. භාවිතා කරන ලද මුළු බිටු ගණන 64 කි.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම

1) **1, 11, 32**
In this option, the sign bit is correctly assigned 1 bit. However, the exponent and mantissa bits do not add up correctly to 64 bits. The exponent is given 11 bits (which is correct), but the mantissa is given only 32 bits, resulting in a total of 44 bits when combined with the sign bit, which is incorrect.
මෙම විකල්පය තුළ, සලකුණු බිටු නිවැරදිව 1 බිටු ලබා දී ඇත. කෙසේ වෙතත්, exponent සහ mantissa බිටු බිටු 64 දක්වා නිවැරදිව එකතු නොවේ. ඝාතකයට බිටු 11 ක් ලබා දී ඇත (එය නිවැරදියි), නමුත් mantissa ලබා දී ඇත්තේ බිටු 32ක් පමණි, එහි ප්‍රතිඵලය ලෙස ලකුණු බිටු සමඟ එකතු වූ විට සම්පූර්ණ බිටු 44ක් ලැබේ, එය වැරදිය.

2) **1, 7, 24**
In this option, the sign bit is correctly assigned 1 bit. However, the exponent is given 7 bits, and the mantissa 24 bits, totaling only 32 bits, which is the format for single precision, not double precision.
මෙම විකල්පය තුළ, සලකුණු බිටු නිවැරදිව 1 බිටු ලබා දී ඇත. කෙසේ වෙතත්, exponent ට බිටු 7ක් ලබා දී ඇති අතර, mantissa ට බිටු 24ක් ලබා දී ඇති අතර, එය බිටු 32ක් පමණි, එය ද්විත්ව නිරවද්‍යතාවයක් නොව තනි නිරවද්‍යතාව සඳහා වන ආකෘතියයි.

3) **1, 8, 23**
This option correctly assigns 1 bit for the sign. However, 8 bits for the exponent and 23 bits for the mantissa represent the IEEE 754 single precision format, not double precision, totaling 32 bits.
මෙම විකල්පය ලකුණ සඳහා බිටු 1ක් නිවැරදිව පවරයි. කෙසේ වෙතත්, exponent සඳහා බිටු 8ක් සහ mantissa සඳහා බිටු 23ක් නියෝජනය කරන්නේ IEEE 754 තනි නිරවද්‍යතා ආකෘතිය මිස ද්විත්ව නිරවද්‍යතාවයක් නොව, මුළු බිටු 32කි.

4) **1, 11, 52**
This option correctly assigns 1 bit for the sign, 11 bits for the exponent, and 52 bits for the mantissa. This adds up to 64 bits in total, which is the correct format for IEEE 754 double precision representation. The 11 bits for the exponent allow for a range of values, and the 52 bits for the mantissa provide the precision needed for double precision numbers.
මෙම විකල්පය නිවැරදිව ලකුණ සඳහා බිටු 1 ක්, exponent සඳහා බිටු 11 ක් සහ mantissa සඳහා බිටු 52 ක් නියම කරයි. මෙය IEEE 754 ද්විත්ව නිරවද්‍යතා නිරූපණය සඳහා නිවැරදි ආකෘතිය වන මුළු බිටු 64ක් දක්වා එකතු කරයි. exponent සඳහා වන බිටු 11 අගයන් පරාසයකට ඉඩ සලසයි, සහ mantissa සඳහා බිටු 52 ද්විත්ව නිරවද්‍යතා සංඛ්‍යා සඳහා අවශ්‍ය නිරවද්‍යතාවය සපයයි.

5) **1, 12, 33**
In this option, the sign bit is correctly assigned 1 bit. However, 12 bits for the exponent and 33 bits for the mantissa add up to 46 bits, which is not the correct total for double precision (64 bits).
මෙම විකල්පය තුළ, සලකුණු බිට් නිවැරදිව 1 බිට් ලබා දී ඇත. කෙසේ වෙතත්, exponent සඳහා බිටු 12 ක් සහ mantissa සඳහා බිටු 33 ක් බිටු 46 ක් දක්වා එකතු වේ, එය ද්විත්ව නිරවද්‍යතාවය බිටු 64 සඳහා නිවැරදි එකතුව නොවේ.

So, the correct configuration for the IEEE 754 double precision format is 1 bit for the sign, 11 bits for the exponent, and 52 bits for the mantissa.
එබැවින්, IEEE 754 ද්විත්ව නිරවද්‍යතා ආකෘතිය සඳහා නිවැරදි වින්‍යාසය ලකුණ සඳහා බිට් 1 ක්, exponent සඳහා බිටු 11 ක් සහ mantissa සඳහා බිටු 52 ක් වේ.

Therefore, the correct answer is 4).
එබැවින්, නිවැරදි පිළිතුර 4) වේ.

3. What is the output of a full adder when A = 1, B = 1, and C_{in} = 1?
A = 1, B = 1, and Cin = 1 විට පූර්ණ ආකලකයක ප්‍රතිදානය කුමක්ද?

1) Sum = 1, Cout = 1
2) Sum = 0, Cout = 0
3) Sum = 1, Cout = 0
4) Sum = 0, Cout = 1
5) None of the above / ඉහත කිසිවක් නොවේ.

Answer : 1)

The truth table of a Full adder circuit
පූර්ණ ආකලයක සත්‍යතා වගුව

Cin	A	B	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

According to this answer should be 1.
මේ අනුව පිලිතුරු අංක 1 නිවැරදි වේ.

4. Which of the following is NOT an example of application software?
යෙදුම් මෘදුකාංග සඳහා උදාහරණයක් නොවන්නේ පහත ඒවායින් කුමක් ද?

1) Database Management System (DBMS) / දත්ත සමුදා කළමනාකරණ පද්ධති
2) Web Browser / වෙබ් අතිරික්සු
3) Disk Defragmenter / තැටි ප්‍රතිඛණ්ඩණය
4) Media Player / මාධ්‍ය වාදන මෘදුකාංග
5) Word Processing Software / වදන් සැකසුම් මෘදුකාංග

Answer : 3)

The correct answer is 3) Disk Defragmenter.
නිවැරදි පිළිතුර 3) Disk Defragmenter වේ.

Application software refers to programs designed to perform specific tasks for users, such as browsing the web, playing media, or managing data.
යෙදුම් මෘදුකාංග යනු අන්තර්ජාලයේ සැරිසැරීම, මාධ්‍ය වාදනය කිරීම හෝ දත්ත කළමනාකරණය කිරීම වැනි පරිශීලකයින් සඳහා නිශ්චිත කාර්යයන් ඉටු කිරීම සඳහා නිර්මාණය කර ඇති වැඩසටහන් වේ.

Disk Defragmenter is a utility software, not an application software. It is used to reorganize the data on a disk to improve access speed and efficiency, which is a system-level task rather than a user-focused application.
Disk Defragmenter යනු උපයෝගිතා මෘදුකාංගයක් මිස යෙදුම් මෘදුකාංගයක් නොවේ. ප්‍රවේශ වේගය සහ කාර්යක්ෂමතාව වැඩි දියුණු කිරීම සඳහා තැටියක දත්ත ප්‍රතිසංවිධානය කිරීමට එය භාවිතා කරයි, එය පරිශීලක-නාභිගත යෙදුමකට වඩා පද්ධති මට්ටමේ කාර්යයකි.

The other options, Database Management System (DBMS), Web Browser, Media Player, and Antivirus Software are all examples of application software designed for specific user tasks.
අනෙකුත් විකල්ප, Database Management System (DBMS), Web Browser, Media Player සහ Word Processing Software යන සියල්ලම විශේෂිත පරිශීලක කාර්යයන් සඳහා නිර්මාණය කර ඇති යෙදුම් මෘදුකාංග සඳහා උදාහරණ වේ.

5. Which of the following SQL statements is used to retrieve all rows and columns from a table named 'customers' where the 'city' column contains the value 'New York'?
'city' තීරුවේ 'New York' අගය අඩංගු 'customers' නම් වගුවකින් සියලුම පේළි සහ තීරු ලබා ගැනීමට පහත සඳහන් SQL ප්‍රකාශවලින් කුමන ප්‍රකාශයක් භාවිතා කරයිද?

1) `SELECT * FROM customers WHERE city = 'New York';`
2) `SELECT * FROM customers WHERE city = New York;`
3) `SELECT customers.* FROM customers WHERE city = 'New York';`
4) `SELECT ALL FROM customers WHERE city = New York;`
5) `DESCRIBE customers WHERE city = 'New York';`

Answer: 1)

Explanation / පැහැදිලි කිරීම:

`SELECT *`
This part of the statement instructs the database to select all columns (*) from the specified table.
ප්‍රකාශයේ මෙම කොටස සඳහන් කර ඇති වගුවෙන් සියලුම තීරු (*) තෝරා ගැනීමට දත්ත සමුදායට උපදෙස් දෙයි.

`FROM customers`
Here, we specify the table name from which we want to retrieve data. In this case, it's the `customers` table.
මෙහිදී, අපි දත්ත ලබා ගැනීමට අවශ්‍ය වගුවේ නම සඳහන් කරමු. මෙම අවස්ථාවේදී, එය `customers` වගුවයි.

`WHERE city = 'New York'`
This part is a condition that filters the rows returned by the query. It instructs the database to only retrieve rows where the value in the `city` column is `New York`.
මෙම කොටස විමසුම මගින් ආපසු ලබා දෙන පේළි පෙරහන කොන්දේසියකි. එය `city` තිරුවේ අගය `New York` වන පේළි පමණක් ලබා ගැනීමට දත්ත ගබඩාවට උපදෙස් දෙයි.

Therefore, the correct answer is 1).
එබැවින්, නිවැරදි පිළිතුර වන්නේ 1) වේ.

6. Consider the following Python code and select the correct answer.
පහත පයිතන් කේතය සලකා බලා නිවැරදි පිළිතුර තෝරන්න.

```
prices = [10.99, 5.99, 3.49, 8.99, 12.99]
total_amount = sum(prices)
print(total_amount)
```

1) `424.5` 2) `4.245` 3) `43.45` 4) <u>`42.45`</u> 5) `4245`

Answer: 4)

Explanation / පැහැදිලි කිරීම

In this Python code, several different prices are stored in a list called `prices`. The main purpose of this code is to calculate the sum of these prices. The function `sum(prices)` is used to achieve this. It returns the sum of all the values in the list. The sum is stored in a variable called `total_amount`. Finally, `print(total_amount)` displays the sum of the values.
මෙම පයිතන් කේතයේ විවිධ මිල ගණන් කිහිපයක් `prices` නම් ලැයිස්තු වල ගබඩා කර ඇත. මෙම කේතයේ ප්‍රධාන අරමුණ වන්නේ මෙම මිල ගණන්වල එකතුව ගණනය කිරීමයි. ඒ සඳහා `sum(prices)` භාවිතා වේ. මෙය ලැයිස්තුවේ ඇති සියලුම අගයන්හි එකතුව ලබා දෙයි. එකතුව `total_amount` නම් ශ්‍රිතයේ ගබඩා වේ අවසානයේ `print(total_amount)` අගයන්හි එකතුව මුද්‍රණය කරයි.

Therefore, the correct answer is 4).
එබැවින්, නිවැරදි පිළිතුර වන්නේ 4) වේ.

7.
```
summation = 0

for _ in range(10):
      number = int(input("Enter a number between 0 and 9: "))
      summation += number
average = summation / 10
print(f"Summation:_{summation}")
print(f"Average:_{average}")
```

What is the output of the above Python code if the inputs are `3,6,2,8,1,4,9,0,7,5` ?
ආදාන `3,6,2,8,1,4,9,0,7,5` නම් ඉහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

1) `Summation:45`
`Average:4.50`
2) `Summation:45`
`Average:4.5`
<u>3)</u> `Summation:_45`
`Average:_4.5`
4) `Summation:-45`
`Average:-4.5`
5) `Error`

Answer: 3)

Explanation / පැහැදිලි කිරීම:

In the for loop of this code, a variable is indicated as "`_`". In this regard, you can further clarify by studying the rules of variable naming.
මෙම කේතයෙහි for ලූපය තුල "`_`" ලෙස දක්වා ඇත්තේ විචල්‍යකි. ඒ පිලිබඳව ඔබට විචල්‍ය නාමකරණයේ නීති පිලිබඳව අධ්‍යනයෙන් වැඩිදුරටත් පැහැදිලි කරගත හැක.

What has happened here is that the loop is executed 10 times by the range Function. Each time it runs like this, it takes an input from the user and adds them together. And the value obtained by adding is divided by 10 and its mean is found.
මෙහිදී සිදුවී ඇත්තේ range Function එක මගින් ලූපය 10 වතාවක් ක්‍රියාත්මක වීමයි. එලෙස ක්‍රියාත්මක වන සෑම අවස්ථාවේම පරිශීලකයාගෙන් අදානයක් ලබාගන්නා අතර ඒවා එකිනෙක එකතු කිරීම සිදුකරයි. තවද එකතු කිරීමෙන් ලැබෙන අගය 10න් බෙදා එහි මධ්‍යනය සොයනු ලබයි.

Here `f"Summation` is the data type called f-strings. Here is the data type called.
මෙහි `f"Summation` යනුවෙන් දක්වා ඇත්තේ f-strings නම් වූ දත්ත වර්ගයයි.

And `print(f"Summation:_{summation}")` is the standard way to output F-strings. Learn more about this.
එමෙන්ම `print(f"Summation:_{summation}")` ලෙස දක්වා ඇත්තේ F-strings ප්‍රතිදානය කරන සම්මත ආකාරය වේ. මේ පිලිබදව වැඩිදුර අධ්‍යනය කරන්න.

Accordingly, here the answer is as follows.
ඒ අනුව මෙහි පිලිතුර පහත පරිදි වේ.

```
Summation:_45
Average:_4.5
```

Therefore, the correct answer is option 3).
එබැවින්, නිවැරදි පිළිතුර විකල්පය 3) වේ.

8. Answer question according to the code below. / පහත කේතයට ප්‍රශ්ණයට පිළිතුරු සපයන්න.

```
players = {
    "Vaas": {"Combat": 85, "Survival": 90, "Leadership": 75},
    "Pagan": {"Intelligence": 92, "Manipulation": 85, "Strategy": 80}}
def total(player):
    total = sum(player.values())
    return total
def average(player):
    average = sum(player.values()) // len(player)
    return average
for name, scores in players.items():
    score = total(scores)
    score = average(scores)
    print(f"Player: {name}")
    print(f"Total Score: {score}")
    print(f"Average Score: {score:.2f}")
    print()
```

What will be the output? / ප්‍රතිදානය කුමක් වේද?

1)
```
Player: Vaas
Total Score: 83
Average Score: 83.00

Player: Pagan
Total Score: 85
Average Score: 85.00
```

2)
```
Player: Vaas
Total Score: 83
Average Score: 83

Player: Pagan
Total Score: 85
Average Score: 85.00
```

3)
```
Player: Vaas
Total Score: 88
Average Score: 83.00

Player: Pagan
Total Score: 85
Average Score: 85.00
```

4)
```
Player: Vaas
Total Score: 88
Average Score: 83

Player: Pagan
Total Score: 85
Average Score: 85
```

5)
```
Player: pagan
Total Score: 88
Average Score: 83.00

Player: vass
Total Score: 85
Average Score: 85.00
```

Answer : 1)

`total(player)` calculates the total score by summing up all values in the `player` dictionary.
`total(player)` මගින් `player` dictionary එකෙහි ඇති සියලුම අගයන් සාරාංශ කිරීමෙන් මුළු ලකුණු ගණනය කරයි.

`average(player)` calculates the average score by dividing the total sum of values by the number of items (attributes) in the `player` dictionary. Note the use of integer division (`//`), which truncates any decimal places.
`average(player)` මගින් සාමාන්‍ය ලකුණු ගණනය කරන්නේ `player` dictionary එකෙහි ඇති අයිතම ගණන (ගුණාංග) ගණනින් මුළු අගයන් බෙදීමෙනි. ඕනෑම දශම ස්ථානයක් කප්පාදු කරන පූර්ණ සංඛ්‍යා බෙදීම (`//`) භාවිතා කිරීම සලකන්න.

The for loop iterates over each player (name and scores) in the `players` dictionary.
for ලූපය `players` dictionary එකෙහි එක් එක් ක්‍රීඩකයා (නම සහ ලකුණු) මත පුනරාවර්තනය වේ.

Then, for each player, it calculates both the total and average scores using the defined functions. It prints the player's name (`name`), total score, and average score.
ඉන්පසුව, එක් එක් ක්‍රීඩකයා සඳහා, එය අර්ථ දක්වා ඇති කාර්යයන් භාවිතා කරමින් සම්පූර්ණ සහ සාමාන්‍ය ලකුණු දෙකම ගණනය කරයි. එය ක්‍රීඩකයාගේ නම (`name`), සම්පූර්ණ ලකුණු සහ සාමාන්‍ය ලකුණු මුද්‍රණය කරයි.

But here in the given code, the variable score will end up containing the result of `average(scores)` because it overwrites the previous assignment from `total(scores)`. Therefore, average score will be printed for both total score and average score.
නමුත් මෙහි දී ලබා දී ඇති කේතයේ, විචල්‍ය ලකුණු අවසන් වන්නේ `average(scores)` ප්‍රතිඵලය අඩංගු වන බැවින් එය පෙර පැවරුම `total(scores)` නැවත ලියන බැවිනි. එබැවින්, සම්පූර්ණ ලකුණු සහ සාමාන්‍ය ලකුණු යන දෙකම සඳහා සාමාන්‍ය ලකුණු මුද්‍රණය කරනු ලැබේ.

```
score = total(scores)
score = average(scores)
```

Let's compute the average scores for each player:
එක් එක් ක්‍රීඩකයා සඳහා සාමාන්‍ය ලකුණු ගණනය කරමු:

`"Vaas"`:

Total score: `85+90+75=250`
Average score: `250//3=83`

`"Pagan"`:

Total score: `92+85+80=257`
Average score: `257//3=85`

Therefore, the output of the above Python code;
එබැවින්, ඉහත පයිතන් කේතයේ ප්‍රතිදානය;

```
Player: Vaas
Total Score: 83
Average Score: 83.00

Player: Pagan
Total Score: 85
Average Score: 85.00
```

9. How would create a 20-pixel wide, solid red border around an element?
මූලාංගයක් වටා පික්සල්(pixel) 20ක් පළල, තද රතු පැහැති මායිමක් නිර්මාණය කරන්නේ කෙසේ ද?

1) `border: 20px solid red;`
2) `border-style: solid 20px red;`
3) `border-color: red solid 20px;`
4) `outline: 20px solid red;`
5) `border-style: 20px; color: solid red;`

Answer: 1)

Let's consider each option one by one.
එක් එක් විකල්පය එකින් එක සලකා බලමු.

1) `border: 20px solid red;`
This option is valid. The border property correctly sets the width, style, and color of the border.
මෙම විකල්පය නිවැරදියි. border මූලාංගය නිවැරදිව width, style, සහ color සකසයි.

2) `border-style: solid 20px red;`
This option is invalid. The `border-style` property should only define the style of the border, not the width or color.
මෙම විකල්පය වැරදියි. `border-style` මූලාංගය යොදාගන්නේ style සඳහා පමණක් වේ. පළල(width) හෝ වර්ණය(color) සඳහා නොවේ.

3) `border-color: red solid 20px;`
This option is invalid. The `border-color` property should only define the color, not the width or style.
මෙම විකල්පය වැරදියි. `border-color` මූලාංගය යොදාගන්නේ වර්ණය (color) සඳහා පමණක් වේ. width හෝ style සඳහා නොවේ.

4) `outline: 20px solid red;`
This option is invalid. The `outline` property adds a line around an element, but it is not the same as a border.
මෙම විකල්පය වැරදියි. `outline` මූලාංගය, අංගය අවට රේඛාවක් එක් කරයි. නමුත් එය බෝඩරයකට (border) සමාන වේ.

5) `border-style: 20px; color: solid red;`
This option is invalid.The syntax is wrong and does'nt correctly apply the border width, style, or color.
මෙම විකල්පය වැරදියි. කේත කාණ්ඩය වැරදි අතර, border width, style, හෝ color නිවැරදිව යෙදුවා නොවේ.

So, the correct answer is 1).
එබැවින්, නිවැරදි පිළිතුර 1) වේ.

10. Which HTML tag is used to create a dropdown menu?
dropdown menu එකක් සෑදීමට භාවිතා කරන්නේ පහත කුමන HTML උසුලනය ද?

1) `<select>` 2) `<dropdown>` 3) `<menu>` 4) `<list>` 5) `<dropdown-menu>`

Answer: 1)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) **`<select>`**
The `<select>` tag in HTML is used to create a dropdown menu. It is often used in forms to allow users to select one or more options from a list.
HTML හි `<select>` උසුලනය dropdown මෙනුවක් සෑදීමට භාවිතා කරයි. ලැයිස්තුවකින් විකල්ප එකක් හෝ කිහිපයක් තෝරා ගැනීමට පරිශීලකයින්ට ඉඩ දීම සඳහා එය බොහෝ විට ආකෘතිවල භාවිතා වේ.

2) **`<dropdown>`**
This tag does not exist in HTML.
මෙම උසුලනය HTML හි නොපවතී.

3) **`<menu>`**
This tag is used to define a list or menu, but not specifically a dropdown menu.
මෙම උසුලනය ලැයිස්තුවක් හෝ මෙනුවක් පිලිබිඹු කිරීමට භාවිතා කරයි, නමුත් dropdown මෙනුවක් සඳහා නොවේ.

4) **`<list>`**
This tag does not exist in HTML.
මෙම උසුලනය HTML හි නොපවතී.

5) **`<dropdown-menu>`**
This tag does not exist in HTML.
මෙම උසුලනය HTML හි නොපවතී.

Therefore, the correct answer is 1).
එබැවින්. නිවැරදි පිළිතුර 1) වේ.