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 225 MARKING

2025/07/23
ictfromabc
Ravindu Bandaranayake

1. "When entering the username into the relevant software, it cannot contain symbols and spaces"
The test that can be used to satisfy this requirement is,
"අදාළ මෘදුකාංගයට පරිශීලක නාමය ඇතුලත් කිරීමේදී ඒ තුල සංකේත හා හිස්තැන් පැවතිය නොහැක"
මෙම අවශ්‍යතාවය තෘප්තිමත් කිරීම සඳහා භාවිතා කළ හැකි පරීක්ෂාව වන්නේ,

   1) Space Checking / හිස්තැන් පරීක්ෂා කිරීම.
   2) Type Checking / වර්ග පරීක්ෂාව.
   3) Character Checking / අනුලක්ෂණ පරීක්ෂා කිරීම.
   4) Availability Checking / ඇති බව පරීක්ෂාව.
   5) Range Checking / පරාස පරීක්ෂාව.

   Answer : 3)

   When entering a username into relevant software, it's important to ensure that the input adheres to specific criteria, particularly the exclusion of symbols and spaces. This requirement is crucial for maintaining data consistency, security, and usability within the software.
   අදාළ මෘදුකාංගයට පරිශීලක නාමයක් ඇතුළු කරන විට, ආදානය නිශ්චිත නිර්ණායකවලට, විශේෂයෙන් සංකේත සහ අවකාශයන් බැහැර කරන බව සහතික කිරීම වැදගත් වේ. මෘදුකාංගය තුළ දත්ත අනුකූලතාව, ආරක්ෂාව සහ උපයෝගිතාව පවත්වා ගැනීම සඳහා මෙම අවශ්‍යතාවය ඉතා වැදගත් වේ.

   To satisfy this requirement, different types of validation tests can be considered. Among them, **Character Checking** is the most appropriate test.
   මෙම අවශ්‍යතාවය තෘප්තිමත් කිරීම සඳහා, විවිධ වර්ගයේ වලංගුකරණ පරීක්ෂණ සලකා බැලිය හැකිය. ඒ අතරින් (Character Checking) වඩාත් සුදුසු පරීක්ෂණයයි.

   Character Checking involves verifying each character of the username against a predefined set of acceptable characters. In this context, the acceptable characters typically include letters (both uppercase and lowercase) and numbers. By implementing Character Checking, the software can immediately identify and reject any input containing symbols (such as @, #, $, etc.) and spaces, which are not allowed in usernames.
   අනුලක්ෂණ පරීක්ෂාව යනු පරිශීලක නාමයේ එක් එක් අක්ෂරය පූර්ව නිශ්චිත පිළිගත හැකි අක්ෂර කට්ටලයකට එරෙහිව සත්‍යාපනය කිරීමයි. මෙම සන්දර්භය තුළ, පිළිගත හැකි අක්ෂරවලට සාමාන්‍යයෙන් අකුරු (ලොකු අකුරු සහ කුඩා අකුරු දෙකම) සහ අංක ඇතුළත් වේ. අක්ෂර පරීක්ෂාව ක්‍රියාත්මක කිරීමෙන්, මෘදුකාංගයට පරිශීලක නාමවල ඉඩ නොදෙන සංකේත (@, #, $, etc.) සහ හිස්තැන් අඩංගු ඕනෑම ආදානයක් වහාම හඳුනාගෙන ප්‍රතික්ෂේප කළ හැක.

   Other validation tests like Space Checking, Type Checking, Availability Checking, and Range Checking are also useful but serve different purposes. Space Checking specifically looks for spaces but doesn't account for symbols. Type Checking ensures the data type is correct (e.g., string), but doesn't check the content of the string. Availability Checking confirms if the username is already taken, and Range Checking ensures the input falls within a specified range, such as length.
   අවකාශ පරීක්ෂාව, වර්ගය පරීක්ෂාව, පවතින බව පරීක්ෂාව සහ පරාසය පරීක්ෂාව වැනි වෙනත් වලංගුකරණ පරීක්ෂාවන් ද ප්‍රයෝජනවත් නමුත් විවිධ අරමුණු සඳහා සේවය කරයි. අවකාශ පරීක්ෂාව විශේෂයෙන් අවකාශයන් සොයන නමුත් සංකේත සඳහා ගණන් නොගනී. වර්ගය පරීක්ෂාව දත්ත වර්ගය නිවැරදි බව සහතික කරයි (උදා: string), නමුත් string හි අන්තර්ගතය පරීක්ෂා නොකරයි. පවතින බව පරීක්ෂාව පරිශීලක නාමය දැනටමත් ගෙන තිබේද යන්න තහවුරු කරයි, සහ පරාසය පරීක්ෂාව ආදානය දිග වැනි නිශ්චිත පරාසයක් තුළට වැටෙන බව සහතික කරයි.

Thus, Character Checking is the comprehensive and suitable approach to ensure usernames do not contain prohibited symbols and spaces, fulfilling the software's input requirements effectively.
මේ අනුව, මෘදුකාංගයේ ආදාන අවශ්‍යතා වලදායී ලෙස ඉටු කරමින්, පරිශීලක නාමවල තහනම් සංකේත සහ අවකාශයන් අඩංගු නොවන බව සහතික කිරීම සඳහා වන විස්තිරණ සහ සුදුසු ප්‍රවේශය අනුලක්ෂණ පරීක්ෂාවයි.

2. Which of the following statements about the IDE and SATA connectors on a motherboard is false?
මවුපුවරුවක පවතින IDE සහ SATA සම්බන්ධක පිළිබඳ පහත ප්‍රකාශ වලින් අසත්‍ය වන්නේ කුමන ප්‍රකාශයද?

   1) IDE is serial and SATA is parallel.
   IDE කෙවෙනි ශ්‍රේණිගත වන අතර SATA කෙවෙනි සමාන්තරගත වේ.
   2) Newer motherboards only feature SATA connectors.
   නව මවුපුවරු වල SATA සම්බන්ධක පමණක් දක්නට ලැබේ.
   3) IDE consumes more power while SATA consumes less power.
   IDE කෙවෙනි වැඩි ශක්තියක් පරිභෝජනය කරන අතර SATA කෙවෙනි අඩු ශක්තියක් පරිභෝජනය කරයි.
   4) IDE connectors have lower data transfer speeds while SATA connectors have higher data transfer speeds.
   IDE කෙවෙනිවලට අඩු දත්ත සම්ප්‍රේෂණ වේගයක් ඇති අතර SATA කෙවෙනිවලට වැඩි දත්ත සම්ප්‍රේෂණ වේගයක් පවති.
   5) IDE is more prone to interference in transmission while SATA is less prone to interference in transmission.
   IDE මගින් සම්ප්‍රේෂණයේදී බාධා ඇතිවීමේ වැඩි ඉඩකඩක් පවතින අතර SATA මගින් සම්ප්‍රේෂණයේදී බාධා ඇතිවීම අවම වේ.

Answer : 1)

Let's consider a comparison of IDE and SATA connectors in a table.
IDE සහ SATA සම්බන්ධක පිළිබඳ සංසන්දනයක් වගුවක් මගින් සලකා බලමු.

Feature / ලක්ෂණය	**IDE (Integrated Drive Electronics)**	**SATA (Serial ATA)**
Technology තාක්ෂණය	Parallel data transmission සමාන්තර දත්ත සම්ප්‍රේෂණය	Serial data transmission ශ්‍රේණිගත දත්ත සම්ප්‍රේෂණය
Compatibility භාවිතය	Older technology; compatible with older systems and hard drives. Limited support in modern devices. පැරණි තාක්ෂණය; පැරණි පද්ධති සහ දෘඩ තැටි සමඟ අනුකූල වේ. නවීන උපාංග සඳහා සීමිත සහාය.	Widely used in modern devices, backward compatible with earlier SATA versions. නවීන උපාංගවල බහුලව භාවිතා වේ, පෙර SATA අනුවාද සමඟ පසුගාමී අනුකූල වේ.
Power Consumption බලශක්ති පරිභෝජනය	Higher power consumption due to more pins and parallel data transmission. වැඩි පින් සහ සමාන්තර දත්ත සම්ප්‍රේෂණය හේතුවෙන් වැඩි බලශක්ති පරිභෝජනය.	Lower power consumption due to fewer pins and serial data transmission. අල්ප පින් සහ ශ්‍රේණිගත දත්ත සම්ප්‍රේෂණය හේතුවෙන් අඩු බලශක්ති පරිභෝජනය.
Data Transfer Speed දත්ත හුවමාරු වේගය	100 MB/s to 133 MB/s	1.5 Gbps (SATA I), 3 Gbps (SATA II), 6 Gbps (SATA III), up to 16 Gbps (SATA Express)

Interference in Transmission සම්ප්‍රේෂණයට බාධා වීම	Higher susceptibility to electromagnetic interference (EMI) due to parallel data transmission and ribbon cables. සමාන්තර දත්ත සම්ප්‍රේෂණය සහ රිබන් කේබල් නිසා විද්‍යුත් චුම්භක බලපෑම් (EMI) වලට වැඩි ඉඩකඩක් පවතී.	Lower susceptibility to interference due to differential signaling and shielded cables. අවකල සංඥා සහ ආරක්ෂිත කේබල් නිසා බලපෑම් වලට අඩු ඉඩකඩක් පවතී.

Accordingly, the first statement is wrong here. So, the correct answer number here is 1.
ඒ අනුව මෙහි වැරදි වන්නේ පළමු ප්‍රකාශයයි. එමනිසා, මෙහි නිවැරදි පිළිතුරු අංකය 1 වේ.

3. Which of the following answers is not a valid Binary Coded Decimal (BCD) value?
පහත සඳහන් පිලිතුරු වලින් වලංගු BCD අගයක් නොවන්නේ කුමක් ද?

1) 10001000_{BCD} 2) $\underline{11110001}_{BCD}$ 3) 10010111_{BCD} 4) 00010101_{BCD} 5) 00110011_{BCD}

Answer : 2)

Binary Coded Decimal (BCD) is a representation of decimal numbers where each digit of a decimal number is represented by its own 4-bit binary equivalent. Valid BCD values only range from 0000 (which represents the decimal digit 0) to 1001 (which represents the decimal digit 9). Any 4-bit binary value above 1001 is not valid in BCD.
Binary Coded Decimal (BCD) යනු දශම සංඛ්‍යා නියෝජනය කිරීමකි, එහිදී දශම සංඛ්‍යාවක එක් එක් ඉලක්කම් එහිම (4-bit) ද්විමය සමානතාවයකින් නිරූපණය කෙරේ. වලංගු BCD අගයන් 0000 (දශම ඉලක්කම් 0 නියෝජනය කරන) සිට 1001 දක්වා (එය දශම ඉලක්කම් 9) නියෝජනය කරයි) පමණී. 1001 ට වැඩි ඕනෑම 4-බිට් ද්විමය අගයක් BCD හි වලංගු නොවේ.

1) 10001000 BCD
This value can be split into two groups 1000 and 1000, which represent decimal digits 8 and 8. Both are valid BCD digits.
මෙම අගය කණ්ඩායම් දෙකකට බෙදිය හැකිය 1000 සහ 1000, දශම ඉලක්කම් 8 සහ 8 නියෝජනය කරයි. දෙකම වලංගු BCD ඉලක්කම් වේ.

2) 11110001 BCD
When split into two groups 1111 and 0001, 1111 represents a binary value of 15, which is not a valid BCD digit. This makes this value an invalid BCD.
කණ්ඩායම් දෙකකට බෙදූ විට 1111 සහ 0001, 1111 වලංගු BCD ඉලක්කම් නොවන 15 ක ද්විමය අගයක් නියෝජනය කරයි. මෙය මෙම අගය වලංගු නොවන BCD බවට පත් කරයි.

3) 10010111 BCD
Split into 1001 and 0111, which represent decimal digits 9 and 7. Both are valid BCD digits.
දශම ඉලක්කම් 9 සහ 7 නියෝජනය කරන 1001 සහ 0111 වලට බෙදන්න. දෙකම වලංගු BCD ඉලක්කම් වේ.

4) 00010101 BCD
Split into 0001 and 0101, which represent decimal digits 1 and 5. Both are valid BCD digits.
දශම ඉලක්කම් 1 සහ 5 නියෝජනය කරන 0001 සහ 0101 වලට බෙදන්න. දෙකම වලංගු BCD ඉලක්කම් වේ.

5) 00110011 BCD
Split into 0011 and 0011, which represent decimal digits 3 and 3. Both are valid BCD digits.
දශම ඉලක්කම් 3 සහ 3 නියෝජනය කරන 0011 සහ 0011 වලට බෙදන්න. දෙකම වලංගු BCD ඉලක්කම් වේ.

So, option **2)** is the correct answer.
එබැවින්, විකල්පය 2) නිවැරදි පිළිතුර වේ.

4. $89_{10} + 17_{10} = \ldots\ldots\ldots$

What is the binary value equivalent to the answer ?
මෙහි පිළිතුරට තුල්‍ය වන ද්විමය අගය වන්නේ කුමක්ද?

1) 01101011 2) 01100011 3) 01101010 4) 01100010 5) 01111010

Answer : 3)

Here, $89_{10} + 17_{10} = 106_{10}$.
මෙහි, $89_{10} + 17_{10} = 106_{10}$ **වේ.**

Let's convert addition of 89 and 17 which is 106 into binary.
106 **වන** 89 **සහ** 17 **හි එකතුව ද්විමය බවට පරිවර්තනය කරමු.**

Converting 106_{10} into Binary **106_{10} ද්විමය බවට පරිවර්තනය කිරීම**							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	1	0	1	0	1	0
01101010_2							

Therefore, the answer is 3).
එබැවින්, පිළිතුර 3) **වේ.**

5. Which of the following expressions is equivalent to the Boolean expression?
පහත සඳහන් කුමන ප්‍රකාශය බූලිය ප්‍රකාශයට සමාන ද?

$\overline{BC} + (DC + DD)D$

1) D 2) C 3) BD 4) $\overline{B}+\overline{C}+D$ 5) 1

Answer : 4)

$= \overline{BC} + (DC + DD)D$ De Morgan's Law (ඩ මුවාර් ප්‍රමේයය)
$= \overline{B} + \overline{C} + (DC + DD)D$ Idempotent Law (තද්ේවභාවී නීතිය)
$= \overline{B} + \overline{C} + (DC + D)D$ Redundancy Law (සමතිරික්ත නීතිය)
$= \overline{B} + \overline{C} + \overline{DD}$ Idempotent Law (තද්ේවභාවී නීතිය)
$= \overline{B} + \overline{C} + D$

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

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

Solve the above Boolean expression.
ඉහත බූලියානු ප්‍රකාශනය සුළු කරන්න.

1) $A + B + C.D$ 2) $A.\overline{C} + B$ 3) $A + B.D + C$ 4) $A.\overline{B}.D + C$ 5) 1

Answer : 3)

This boolean expression can be simplfied as follows.
මෙම බූලියානු ප්‍රකාශනය පහත ආකාරයට සුළු කළ හැකිය.

$A\bar{C} + D\big((BB) \oplus (CA)\big) + C$	
$A\bar{C} + D\big(B \oplus (CA)\big) + C$	Idempotent law තදේවභාවී නීතිය
$A + D\big(B \oplus (CA)\big) + C$	Redundancy law සමතිරික්ත නීතිය
$A + D(\bar{B}CA + B\bar{C}\bar{A}) + C$	$A \oplus B = \bar{A}B + A\bar{B}$
$A + D(\bar{B}CA + B(\bar{C} + \bar{A})) + C$	De Morgan's law ද මුවාර් ප්‍රමේයය
$A + D(\bar{B}CA + B\bar{C} + B\bar{A})) + C$	Distributive law විසටන නීතිය
$A + D\bar{B}CA + DB\bar{C} + DB\bar{A} + C$	Distributive law විසටන නීතිය
$A + DB\bar{C} + DB\bar{A} + C$	Redundancy law සමතිරික්ත නීතිය
$A + DB\bar{C} + DB + C$	Redundancy law සමතිරික්ත නීතිය
$A + DB + DB + C$	Redundancy law සමතිරික්ත නීතිය
$A + DB + C$	Idempotent law තදේවභාවී නීතිය

So, the correct answer number is 3.
එමනිසා, නිවැරදි පිළිතුරු අංකය 3 වේ.

7. What is the primary difference between a half adder and a full adder?
අර්ධ ආකලකය සහ පූර්ණ ආකලකය අතර ඇති මූලික වෙනස කුමක්ද?

1) A half adder can add more than two binary numbers simultaneously.
අර්ධ ආකලකයකින් එකවර ද්විමය සංඛ්‍යා දෙකකට වඩා එකතු කළ හැක
2) A half adder considers carry from previous additions.
අර්ධ ආකලකයකින් පෙර එකතු කිරීම් වලින් රැගෙන යාම සලකා බලයි.
3) A full adder can handle carry input from previous additions.
පූර්ණ ආකලකයකින් පෙර එකතු කිරීම් වලින් රැගෙන ආ ආදානය හැසිරවිය හැක
4) A full adder outputs a binary number in two's complement form.
පූර්ණ ආකලකයකින් ද්විමය සංඛ්‍යාවක් දෙකෙහි අනුපූරක ආකාරයෙන් ප්‍රතිදානය කරයි
5) A half adder has more logic gates than a full adder.
අර්ධ ආකලකයක පූර්ණ ආකලකයකට වඩා තාර්කික ද්වාර ඇත

Answer : 3)

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

- **Statement / ප්‍රකාශය 1)**

This statement is incorrect. A half adder is designed to add only two single-bit binary numbers and cannot handle more than two numbers simultaneously.
මෙම ප්‍රකාශය නිවැරදි නොවේ. අර්ධ ආකලකය නිර්මාණය කර ඇත්තේ තනි-බිට් ද්විමය සංඛ්‍යා දෙකක් පමණක් එකතු කිරීමට වන අතර එකවර සංඛ්‍යා දෙකකට වඩා හැසිරවිය නොහැක.

- **Statement / ප්‍රකාශය 2)**

This statement is incorrect. A half adder does not consider any carry from previous additions; it only adds two single-bit numbers and produces a sum and a carry output.
මෙම ප්‍රකාශය නිවැරදි නොවේ. අර්ධ ආකලකය පෙර එකතු කිරීම් වලින් කිසිදු රැගෙන යාමක් නොසලකයි එය තනි-බිට් අංක දෙකක් පමණක් එකතු කරන අතර එකතුවක් සහ රැගෙන යාමේ ප්‍රතිදානයක් නිපදවයි.

- **Statement / ප්‍රකාශය 3)**

This statement is correct. A full adder can handle the carry input from previous additions in addition to adding two single-bit binary numbers, making it suitable for use in multi-bit binary addition.
මෙම ප්‍රකාශය නිවැරදි වේ. පූර්ණ ආකලකයට තනි-බිට් ද්විමය සංඛ්‍යා දෙකක් එකතු කිරීමට අමතරව පෙර එකතු කිරීම් වලින් රැගෙන ආ ආදානය හැසිරවිය හැක, එය බහු-බිට් ද්විමය එකතු කිරීමේදී භාවිතා කිරීමට සුදුසු වේ.

- **Statement / ප්‍රකාශය 4)**

This statement is incorrect. A full adder outputs the sum and carry bits but does not convert the result into two's complement form.
මෙම ප්‍රකාශය නිවැරදි නොවේ. පූර්ණ ආකලකය එකතුව සහ බිටු ප්‍රතිදානය කරන නමුත් ප්‍රතිවලය ලෙස දෙකේ අනුපූරක ස්වරූපයට පරිවර්තනය නොකරයි.

- **Statement / ප්‍රකාශය 5)**

This statement is incorrect. A half adder generally has fewer logic gates compared to a full adder, as it performs simpler operations.
මෙම ප්‍රකාශය නිවැරදි නොවේ. පූර්ණ ආකලකයට සාපේක්ෂව අර්ධ ආකලකයට සාමාන්‍යයෙන් අඩු තාර්කික ද්වාර සංඛ්‍යාවක් ඇත, මන්ද එය සරල මෙහෙයුම් සිදු කරයි.

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

8. What statement is used to exit a loop in Python?
පයිතන් හි ලූපයකින් පිටවීමට භාවිතා කරන ප්‍රකාශය කුමක් ද?

1) break 2) continue 3) exit 4) stop 5) stop

Answer : 1)

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

1) break

This statement is correct. The `break` statement is used to immediately exit a loop in Python. When `break` is encountered inside a loop (whether a `for loop` or `while loop`), the loop is terminated, and the control flow of the program moves to the next statement following the loop. This is typically used when a specific condition is met, and continuing the loop is no longer necessary.
මෙම ප්‍රකාශය නිවැරදි වේ. පයිතන් හි ලූපයකින් වහාම පිටවීමට `break` ප්‍රකාශය භාවිතා කරයි. ලූපයක් තුළ `break` හමු වූ විට `for loop` හෝ while loop වේවා), ලූපය අවසන් වන අතර, වැඩසටහනේ පාලන ප්‍රවාහය ලූපයෙන් පසුව ඊළඟ ප්‍රකාශය වෙත ගමන් කරයි. නිශ්චිත කොන්දේසියක් සපුරා ඇති විට මෙය සාමාන්‍යයයෙන් භාවිතා වන අතර, ලූපය දිගටම ක්‍රියාත්මක වීම තවදුරටත් අවශ්‍ය නොවේ.

2) continue

This statement is incorrect. The `continue` statement skips the rest of the code inside the loop for the current iteration and proceeds with the next iteration of the loop. It does not exit the loop entirely.
මෙම ප්‍රකාශය නිවැරදි නොවේ. `continue` ප්‍රකාශය වත්මන් පුනරාවර්තනය සඳහා ලූපය තුළ ඇති ඉතිරි කේතය මඟහරින අතර ලූපයේ ඊළඟ පුනරාවර්තනය සමඟ ඉදිරියට යයි. එය ලූපයෙන් සම්පූර්ණයෙන්ම පිටවන්නේ නැත.

3) exit

This statement is incorrect. The `exit()` function is used to terminate the entire Python program, not just a loop. It is commonly used to exit from scripts.
මෙම ප්‍රකාශය නිවැරදි නොවේ. ලූපයක් පමණක් නොව සම්පූර්ණ පයිතන් ක්‍රමලේඛය අවසන් කිරීමට `exit()` ශ්‍රිතය භාවිතා කරයි. එය සාමාන්‍යයෙන් ස්ක්‍රිප්ට් වලින් පිටවීමට භාවිතා කරයි.

4) stop

This statement is incorrect. Python does not have a stop statement to exit a loop.
මෙම ප්‍රකාශය නිවැරදි නොවේ. ලූපයකින් පිටවීමට Python හට stop ප්‍රකාශයක් නොමැත.

5) else

This statement is incorrect. The `else` statement is used in combination with loops to execute code when the loop finishes normally, without encountering a `break` statement.
මෙම ප්‍රකාශය නිවැරදි නොවේ. ලූපය සාමාන්‍යයෙන් අවසන් වන විට, `break` ප්‍රකාශයක් හමු නොවී, කේතය ක්‍රියාත්මක කිරීමට ලූප සමඟ ඒකාබද්ධව `else` ප්‍රකාශය භාවිතා වේ.

So the answer is 1).
එම නිසා පිළිතුර වන්නේ 1) **වේ.**

9. Assume that the following Python statement is executed :
`D={1:'76', 2:'77', 3:'78' }` Which of the following statements would generate an error?

පහත පයිතන් ප්‍රකාශය ක්‍රියාත්මක කර ඇතැයි උපකල්පනය කරන්න
`D={1:'76', 2:'77', 3:'78' }` **පහත සඳහන් කුමන ප්‍රකාශය දෝෂයක් ජනනය කරයිද?**

1) `A=D[2];` 2) `D[1]=[1,2,3]` 3) `A=D['1']` 4) `A=D.keys()` 5) `A=D.values()`

Answer : 3)

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

- **`A = D[2]`**
  This statement accesses the value associated with the key 2 in the dictionary D. Since 2 is a valid key in D, this statement will not generate an error. It will assign the value '77' to the variable A.
  මෙම ප්‍රකාශය D dictionary 2 **යතුර හා සම්බන්ධ අගයට ප්‍රවේශ වේ.** 2 **යනු** D **හි වලංගු යතුරක් වන බැවින්, මෙම ප්‍රකාශය දෝෂයක් ජනනය නොකරයි. එය** A **විචල්‍යයට** 77 **අගය පවරනු ඇත.**

- **`D[1] = [1,2,3]`**
  This statement updates the value associated with the key 1 in the dictionary D to be a list [1, 2, 3]. Python dictionaries allow values to be of any type, including lists. Therefore, this statement will not generate an error. It will change the value associated with the key 1 to [1, 2, 3].
  මෙම ප්‍රකාශය D dictionary **යතුර** 1 **හා සම්බන්ධ අගය ලැයිස්තුවක් ලෙස යාවත්කාලීන කරයි** [1, 2, 3] **පයිතන්** dictionary **ලැයිස්තු ඇතුළුව ඕනෑම වර්ගයක අගයන් වීමට ඉඩ සලසයි. එබැවින්, මෙම ප්‍රකාශය දෝෂයක් ජනනය නොකරනු ඇත. එය යතුර** 1 **හා සම්බන්ධ අගය** [1, 2, 3] **ලෙස වෙනස් කරයි.**

- **`A = D['1']`**
  This statement attempts to access the value associated with the key '1', which is a string. However, in the dictionary D, the keys are integers (1, 2, 3). Since '1' (a string) is not a valid key in D, this will generate a KeyError.
  මෙම ප්‍රකාශය තන්තුවක් වන 1 **යතුර හා සම්බන්ධ අගය වෙත ප්‍රවේශ වීමට උත්සාහ කරයි. කෙසේ වෙතත්,** dictionary D **යතුරු නිබිල** (1,2,3) **වේ.** D **හි** 1 **(තන්තුවක්) වලංගු යතුරක් නොවන බැවින්, මෙය** KeyError **එකක් ජනනය කරයි.**

- **`A = D.keys()`**
  This statement retrieves the keys of the dictionary D. The keys() method returns a view object that displays a list of all the keys in the dictionary. Since this is a valid operation, it will not generate an error.
  මෙම ප්‍රකාශය D **ශබ්දකෝෂයේ යතුරු ලබා ගනී.** values() **ක්‍රමය මඟින් ශබ්ද කෝෂයේ ඇති සියලුම යතුරු ලැයිස්තුවක් පෙන්වන දර්ශන වස්තුවක් ආපසු ලබා දේ. මෙය වලංගු මෙහෙයුමක් බැවින්, එය දෝෂයක් ජනනය නොකරයි.මෙම**

- **`A = D.values()`**
  This statement retrieves the values of the dictionary D. The values() method returns a view object that displays a list of all the values in the dictionary. This is a valid operation and will not generate an error.
  මෙම ප්‍රකාශය D **ශබ්දකෝෂයේ අගයන් ලබා ගනී.** values() **ක්‍රමය මඟින් ශබ්දකෝෂයේ ඇති සියලුම අගයන් ලැයිස්තුවක් පෙන්වන දර්ශන වස්තුවක් ලබා දෙයි. මෙය වලංගු මෙහෙයුමක් වන අතර දෝෂයක් ජනනය නොවේ.**

The statement that would generate an error is **`3) A = D['1']`**. This is because '1' is a string, and the dictionary D only contains integer keys.
දෝෂයක් ඇති කරන ප්‍රකාශය වන්නේ: **`3) A = D['1']`**. **මෙයට හේතුව** 1 **යනු තන්තුවක් වන අතර** D **ශබ්දකෝෂයේ නිබිල යතුරු පමණක් අඩංගු වීමයි.**

10. The value of the Python expression given below would be,
පහත දක්වා ඇති පයිතන් ප්‍රකාශනයේ අගය වනුයේ,

```
4+25//10
```

1) 77 2) 0 3) 3 4) 6 5) 3.5

Answer : 4)

The correct answer is 4) 6.
නිවැරදි පිළිතුර 4) 6 වේ.

In Python, the "//" operator performs integer (floor) division. This means that it divides the first number by the second and rounds down to the nearest whole number.
පයිතන් හි "//" මෙහෙයවනය පූර්ණ සංඛ්‍යා (floor) බෙදීම සිදු කරයි. මෙයින් අදහස් කරන්නේ එය පළමු සංඛ්‍යාව දෙවැන්නෙන් බෙදන අතර ආසන්නතම සම්පූර්ණ සංඛ්‍යාවට වට කරන බවයි.

In the expression "25//10", the result is 2. This is because 25 divided by 10 equals 2.5, but the decimal part is discarded in integer division, leaving only the integer part, 2.
"25//10" ප්‍රකාශනයේ ප්‍රතිඵලය 2 වේ. මෙයට හේතුව 25 10න් බෙදීම 2.5 ට සමාන වීමයි, නමුත් දශම කොටස නිඛිල බෙදීමේදී ඉවතලන අතර, නිඛිල කොටස 2 පමණක් ඉතිරි වේ.

Therefore, when we evaluate the overall expression "4+25//10", we are calculating 4+2, which equals 6.
එබැවින්, "4+25//10" සමස්ත ප්‍රකාශනය ඇගයීමේදී, 6ට සමාන වන 4+2 ගණනය කරමු.