Disclaimer:

1. The questions in this document have been recreated based on post-test discussions with test takers.
2. While the distribution of questions across topics is expected to be similar, variations may occur.
3. Use this document as a guide for indicative preparation, rather than an exact replica of the question pattern for the Accenture Online Test.

Accenture Roles and Packages

This year Accenture recruitment for 2025 pass-outs has come up with two roles with exciting packages. They are as follows.

- **Associate Software Engineer (4.5 LPA)**
- **Advanced Associate Software Engineer (6.5 LPA)**

Accenture Recruitment Process

Accenture Online Test Pattern

Round	Round Name	#Qs	Sections Name	#Qs in Sections	Duration (In Mins)
1	Cognitive and Technical Assessment	90	English Ability	17	90
			Critical Reasoning and Problem Solving	18	
			Abstract Reasoning	15	
			Common Applications and MS Office	12	
			Pseudocode	18	
			Networking Security and Cloud	10	

2	Coding	2	-	-	45

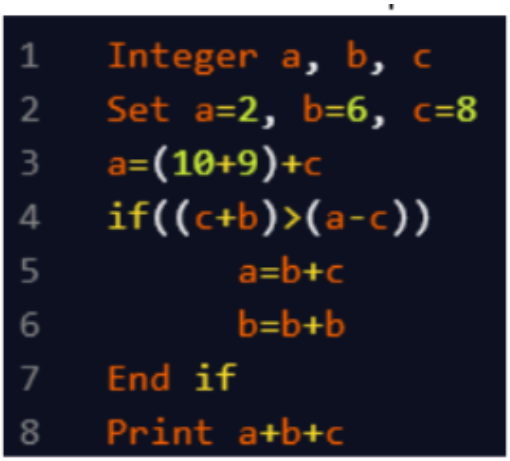
Accenture Online Test Assessment Platform

Assessment Platform	Intra - sectional Navigation	Sectional Navigation Marking Scheme	Marking Scheme
AON-CoCubes/ Wheebox/ Mettl/HirePro	Allowed	Allowed	No Negative Marking

Accenture Online Test Syllabus

Section	Topic
English Ability	● Reading Comprehension ● Articles ● Prepositions ● Sentence Correction, ● Speech and Voice ● Tenses, Synonyms ● Antonyms, Spellings ● Idioms and Phrases
Critical Reasoning and Problem-solving	● Critical reasoning ● Flowcharts ● Data arrangements ● Data sufficiency ● Syllogisms
Abstract Reasoning	● Coding and Decoding ● Visual Reasoning ● Odd man out ● Series
Pseudocode	● Sequence ● While ● Repeat-until ● For ● If-then-else ● Case

Common Applications and MS Office	**MS Word** ● Creating, editing, saving and printing text documents ● Font and paragraph formatting ● Simple character formatting ● Inserting tables, smart art, page breaks ● Using lists and styles ● Working with images ● Using Spelling and Grammar check ● Understanding document properties ● Mail Merge **MS Excel** ● Spreadsheet basics ● Creating, editing, saving, and printing spreadsheets ● Working with functions & formulas ● Modifying worksheets with color & autoformats ● Graphically representing data: Charts & Graphs ● Speeding data entry: Using Data Forms ● Analyzing data: Data Menu, Subtotal, Filtering Data ● Formatting worksheets **MS PowerPoint** ● Securing & Protecting spreadsheets ● Opening, viewing, creating, and printing slides ● Applying auto layouts ● Adding custom animation ● Using slide transitions ● Graphically representing data: Charts & Graphs ● Creating Professional Slides for Presentation.
Network Security and Cloud	● Data and Computer Communication Networks ● Mobile & Wireless Networks ● Cryptography and Network Security ● Database Security ● Software Security ● Biometric Security
Coding	● Data types ● Operators ● Arrays ● Strings ● Decision Making ● Looping ● Functions Scenario-based questions



```
1   Integer a, b, c
2   Set a=2, b=6, c=8
3   a=(10+9)+c
4   if((c+b)>(a-c))
5         a=b+c
6         b=b+b
7   End if
8   Print a+b+c
```

A. 23

B. 41

C. 48

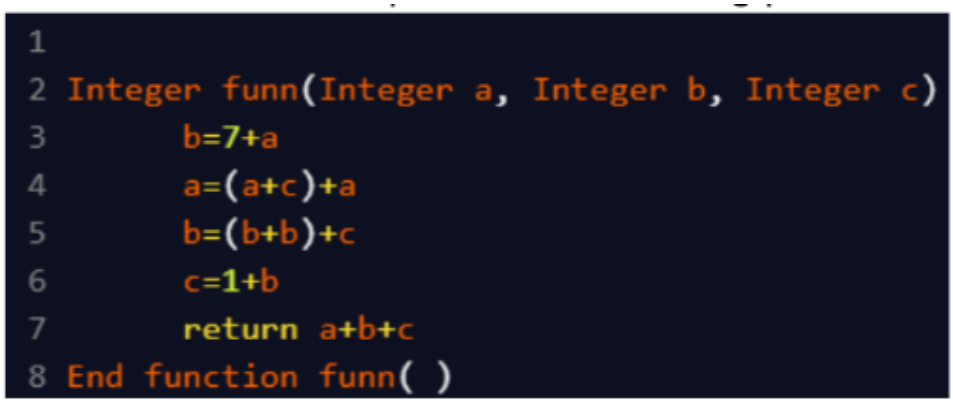
D. 58

Explanation: In the Line (Statement) no. 3, it is updating the value of variable ‘a’ from 2 to (10+9)+c =>19+8=> 27

In the Line (Statement) no. 4, it is a conditional statement and evaluates expression with the updated value of variable a, because order of statement matters. Since (c+b means 8+6 =14) is not greater than (a-c means 27-8=19), the ‘if’ body will not be executed. That’s why it will come directly at statement 8 and will display the sum of a+b+c, which is 27+6+8=41, which means option B will be the correct answer from the above given options.

Answer: B

2. What will be the output of the following pseudocode for a=0, b=2, c=10?



```
1
2 Integer funn(Integer a, Integer b, Integer c)
3       b=7+a
4       a=(a+c)+a
5       b=(b+b)+c
6       c=1+b
7       return a+b+c
8 End function funn( )
```

A. 59

B. 68

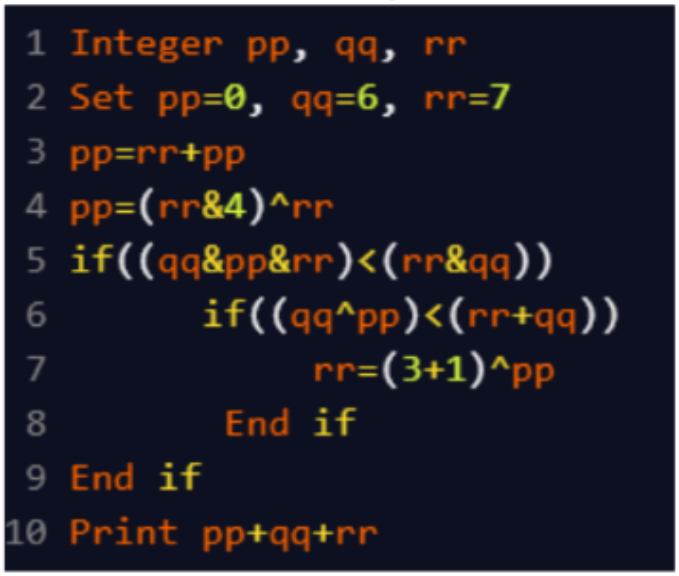
C. 70

D. 39

Explanation: Initial values will be updated as per the order of statements. Like in Line (statement) no. 3: variable ‘b’ will be updated with a value of (7+a) which means 7+0=7. Then, in line 4, variable ‘a’ will be updated with value (a+c)+a which is (0+10)+0=10. Now, in line 5 ‘b’ will be reupdated with (b+b)+c. Keep remember “Precedence of Operators” while assigning the values in ‘b’. First it will execute parenthesis () and will retrieve the current value of b which has updated from line 3 then it will add with value of ‘c’. After execution of line 5 variable ‘b’ will be updated with value (7+7)+10=24. Then, in line 6, the value of variable ‘c’ will be updated with 1+b. Last value of b is 24 so variable ‘c’ will be updated with 1+24=25. Now, from line 7 updated values of variable a, b and c will be added together and return from the function. So the final (return) value will be a+b+c => 10+24+25=59. So the final answer will be option A, 59.

Answer: A

3. What will be the output of the following pseudo code?



```
1 Integer pp, qq, rr
2 Set pp=0, qq=6, rr=7
3 pp=rr+pp
4 pp=(rr&4)^rr
5 if((qq&pp&rr)<(rr&qq))
6       if((qq^pp)<(rr+qq))
7           rr=(3+1)^pp
8        End if
9 End if
10 Print pp+qq+rr
```

Note:

&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

^ is the bitwise XOR operator that compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 29

B. 18

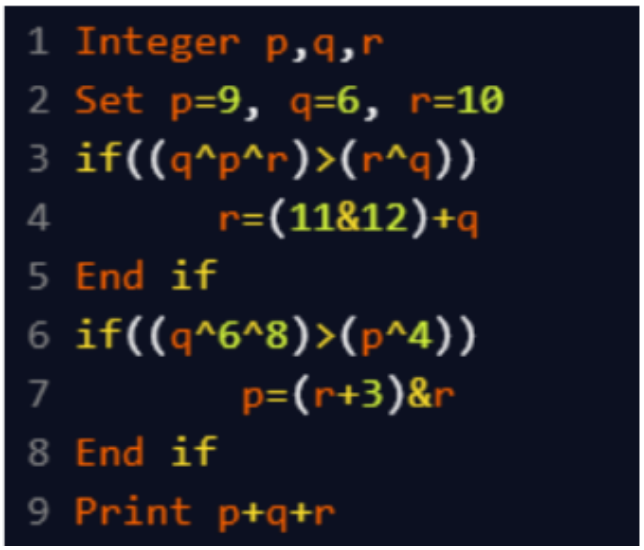
C. 10

D. 16

Explanation: In line 3, variable pp will be updated with value 7+0=7. In line 4, variable pp will be reupdated; first parenthesis will be resolved/evaluated then value will be evaluated with variable rr with XOR operator, means (7&4)^7=4^7=3. In line 5, 'if' condition will be evaluated, (6&3&7) < (7&6) => 2<6, since the condition is True, it will go inside, and then from line 6, it will evaluate the next condition in inner 'if', (6^3)<(7+6) => 5<13, since the condition is True, line 7 will be executed. In line 7, variable rr will be updated with value (4^3)=7. Now statement 10 will be executed with an updated value of variables pp, qq and rr, which means 3+6+7=16. So the final output will be 16, which is option D.

Answer: D

4. What will be the output of the following pseudo code?



Note:

&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

^ is the bitwise XOR operator that compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 20

B. 27

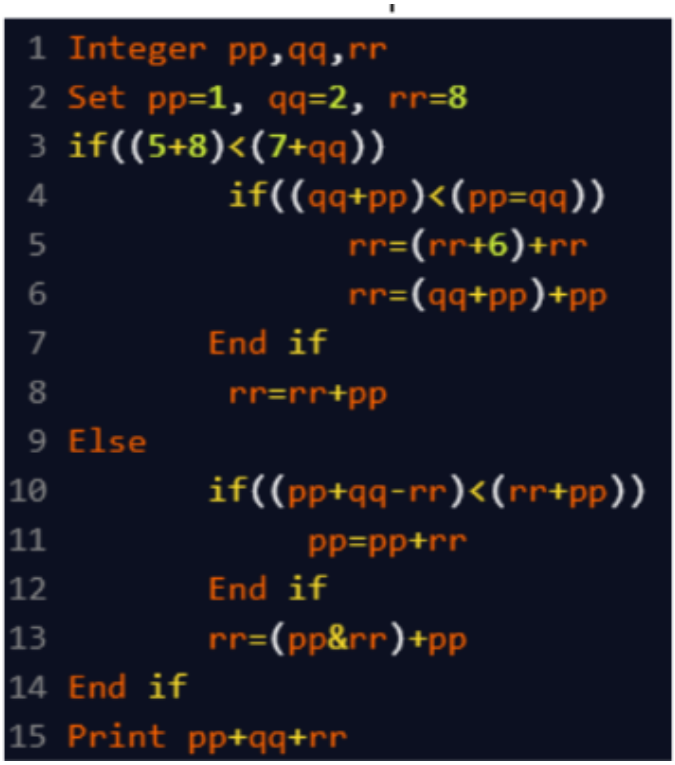
C. 25

D. 36

Explanation: In line 3, 'if' will be checked for the condition: (6^9^10) > (10^6) => 5 > 12 => False. Since the condition is false, line 4 will be skipped, as line 4 is part of 'if'. Then it will come on line 6 and will check the next 'if' condition, which is: (6^6^8) > (9^4) => 8 > 13 => False. Since this 'if' condition is false, it will skip line 7. Then it will come on line 9, and will add all three variables: 9+6+10=25, which is option C.

Answer: C

5. What will be the output of the following pseudocode?



```
1  Integer pp,qq,rr
2  Set pp=1, qq=2, rr=8
3  if((5+8)<(7+qq))
4          if((qq+pp)<(pp=qq))
5                  rr=(rr+6)+rr
6                  rr=(qq+pp)+pp
7          End if
8           rr=rr+pp
9  Else
10         if((pp+qq-rr)<(rr+pp))
11               pp=pp+rr
12         End if
13         rr=(pp&rr)+pp
14 End if
15 Print pp+qq+rr
```

Note:

&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 35

B. 49

C. 19

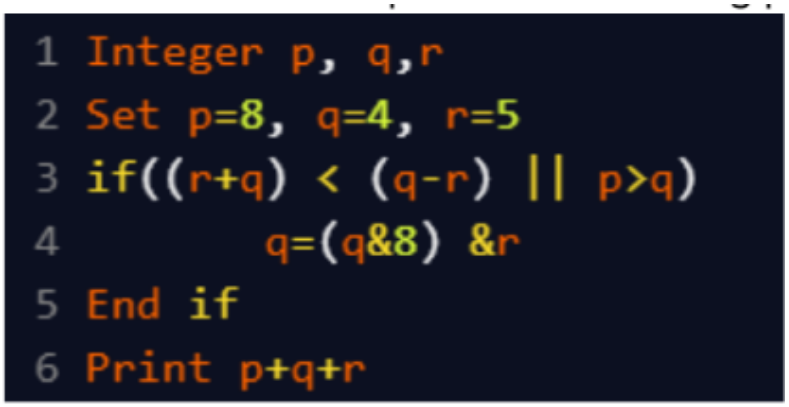
D. 28

Explanation: In line 3, the conditional statement ‘if’ will evaluate on the basis of given condition, which is, 13<(7+2) => 13 < 9, which is False. Since ‘if’ condition is False, the cursor of execution will directly come at line 9; an alternate

part of ‘if’ means else block. Inside else block there is one ‘if’ statement (line 10), so the condition will be executed, which is, (1+2-8)<(8+1) => (-5)<9, which is True, that is why line 11 will be executed and it will update the variable pp. New value for variable pp will be (1+8)=9. Now it will come at line 15 and will add all three variables together and display the result, which will: 9+2+17=28, Option D.

Answer: D

6. What will be the output of the following pseudo code?



Note:

&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

II. Logical OR - The logical OR operator (II) returns the Boolean value TRUE(or 1) if either or both operands is TRUE and FALSE(or 0) otherwise

A. 17

B. 10

C. 23

D. 13

Explanation: After initialization of values in variables p, q and r in line 2, line 3 will be executed. There is an ‘if’ conditional statement, so it will check the condition first. Now, need to understand the ‘precedence of operators’. Logical operators execute after relational. Condition is: (5+4) < (4-5) || (8>4) => (9<-1) || (8>4) => False || True=>

True.

Since the condition is True, line 4 will be executed, which will update the value of variable q: (4&8)&5 => 0. Now it will come at line 6, and will add all three variables with updated values: 8+0+5=13, Option D.

Answer: D

7. What will be the output of the following pseudo code for a=1,b=2,c=9?

```
1 Integer funn( Integer a, Integer b, Integer c)
2         for(each c from 5 to 9)
3             if((b+5)>(a-b))
4               a=(b+5)^a
5             End if
6         b=5^c
7         End for
8   return a+b
```

Note- ^ is the bitwise XOR operator that compares each bit of the first operand to the corresponding bit of the second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 26

B. 55

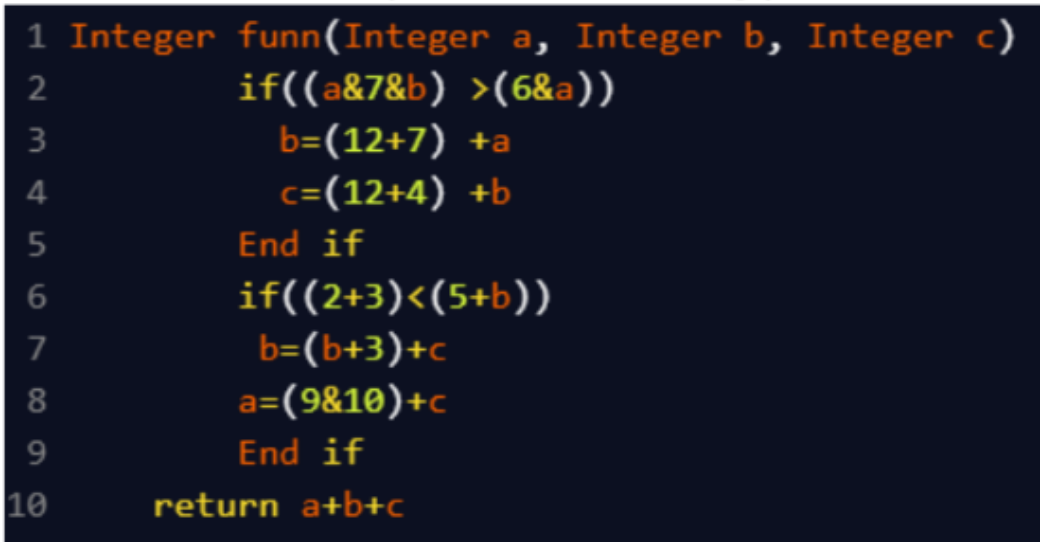
C. 40

D. 44

Explanation: The pseudocode iterates over ‘c’ from 5 to 9, updating ‘a’ and ‘b’ based on the condition (b + 5) > (a - b). The XOR (‘^’) operation is applied to update ‘a’ when the condition is true, and ‘b’ is updated in every iteration using b = 5 ^ c. Starting with a = 1 and b = 2, after processing each value of `c`, the final values are a = 28 and b = 12. The function returns a + b, which equals 40.

Answer: C

8. What will be the output of the following pseudo code for a=2, b=6, c=5?



```
1  Integer funn(Integer a, Integer b, Integer c)
2          if((a&7&b) >(6&a))
3            b=(12+7) +a
4            c=(12+4) +b
5          End if
6          if((2+3)<(5+b))
7           b=(b+3)+c
8          a=(9&10)+c
9          End if
10      return a+b+c
```

Note-&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 33

B. 32

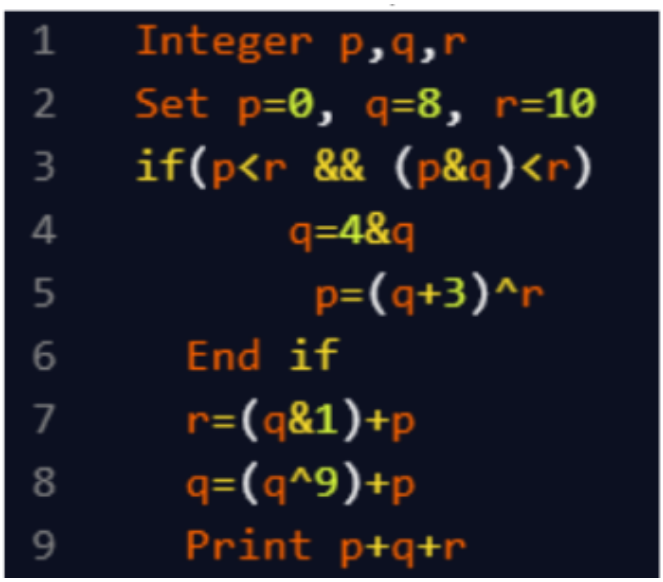
C. 41

D. 28

Explanation: The code first checks if (a & 7 & b) > (6 & a); since this is false for a = 2 and b = 6, the first block is skipped. Next, it checks if (2 + 3) < (5 + b), which is true, so it updates b to 14 and a to 13. Finally, it returns the sum a + b + c = 13 + 14 + 5 = 32.

Answer: B

9. What will be the output of the following pseudocode?



```
Integer p,q,r
Set p=0, q=8, r=10
if(p<r && (p&q)<r)
        q=4&q
         p=(q+3)^r
   End if
   r=(q&1)+p
   q=(q^9)+p
   Print p+q+r
```

Note-&&: Logical AND - The logical AND operator (&&) returns the Boolean value true(or 1) if both operands are true and return false (or 0) otherwise.

&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 45

B. 36

C. 31

D. 38

Explanation: The code checks if p < r and (p & q) < r, which is true for the initial values p = 0, q = 8, and r = 10. Inside the if block, it updates q to 0 and p to 9. Afterward, r becomes 9 and q becomes 18. Finally, the sum p + q + r = 9 + 18 + 9 = 36 is printed.

Answer: B

10. What will be the output of the following pseudocode?

```
1    Integer p,q,r
2    Set p=1, q=4, r=7
3    p=(1+8)+q
4    r=(p&r)+r
5    r=q+q
6    if((q+r)<(r-q) && 7>p)
7                  p=r+q
8                  p=(p+11)+q
9          End if
10         Print p+q+r
11
```

Note-&&: Logical AND - The logical AND operator (&&) returns the Boolean value true(or 1) if both operands are true and return false (or 0) otherwise.

&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 34

B. 20

C. 32

D. 25

Explanation: The code first updates p to 13 and then modifies r to 12 through bitwise AND, and later to 8 by setting r = q + q. The if condition (q + r) < (r - q) is false, so the block inside the if is skipped. Finally, it prints the sum p + q + r = 13 + 4 + 8 = 25.

The output is 25.

Answer: D

11. What will be the output of the following pseudocode?

```
1    Integer p,q,r
2    Set p=6, q=3, r=9
3    if((p&r)<(q-p))
4          p=(2^7)+r
5          p=(p+3)^r
6          q=4^q
7       End if
8       r=(r+p)&q
9    Print p+q+r
```

Note- &: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 10

B. 13

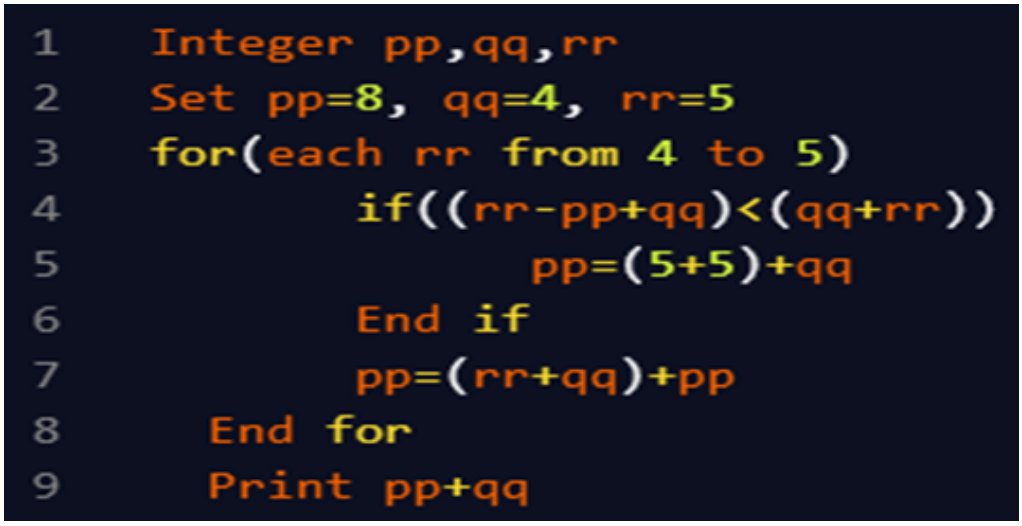
C. 12

D. 19

Explanation: The code initializes p = 6, q = 3, and r = 9. It checks the condition if ((p & r) < (q - p)), where p & r evaluates to 0 and q - p is -3. Since 0 < -3 is false, the if block is skipped. The code then updates r with the expression r = (r + p) & q, resulting in r = 3. Finally, it prints p + q + r, which equals 12.

Answer: C

12. What will be the output of the following pseudocode?



```
Integer pp,qq,rr
Set pp=8, qq=4, rr=5
for(each rr from 4 to 5)
        if((rr-pp+qq)<(qq+rr))
                pp=(5+5)+qq
        End if
        pp=(rr+qq)+pp
  End for
  Print pp+qq
```

A. 22

B. 32

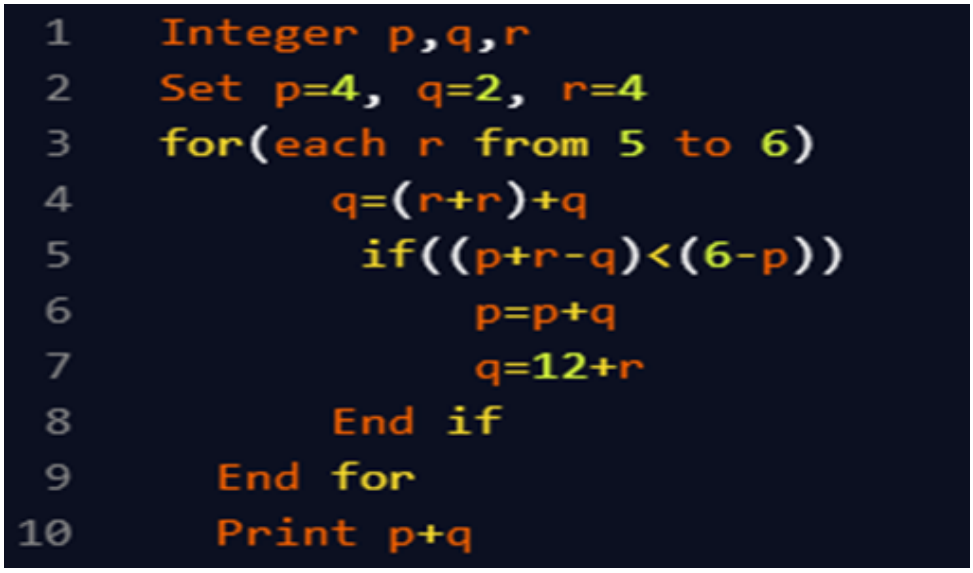
C. 27

D. 45

Explanation: The code initializes pp = 8, qq = 4, and runs a loop for rr values from 4 to 5. For rr = 4, the condition is true, so pp is updated to 14, then to 22. For rr = 5, the condition is true again, updating pp to 14, then to 23. Finally, the code prints pp + qq = 23 + 4 = 27.

Answer: C

13. What will be the output of the following pseudocode?



A. 58

B. 34

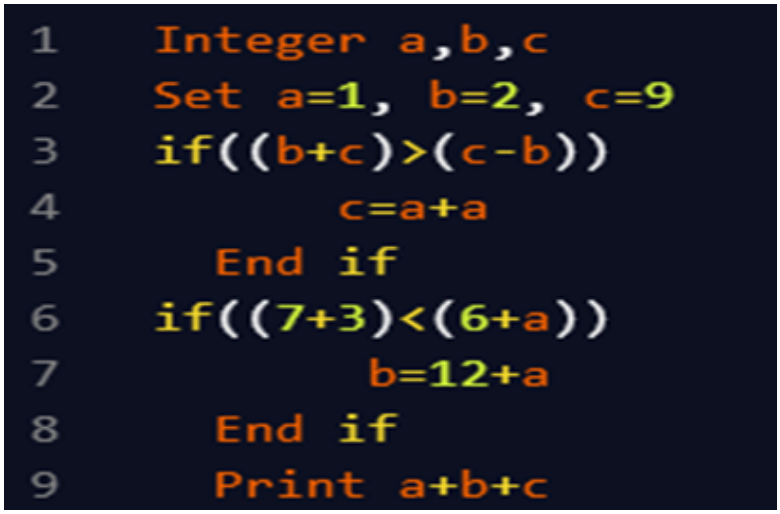
C. 45

D. 49

Explanation: The code initializes p = 4, q = 2, and runs a loop for r = 5 and r = 6. For r = 5, the condition is true, updating p to 16 and q to 17. For r = 6, the condition is false, but q is updated to 29. The final output is p + q = 16 + 29 = 45.

Answer: C

14. What will be the output of the following pseudocode?



A. 5

B. 1

C. 9

D. 20

Explanation:

Initialization: a is set to 1, b is set to 2, and c is set to 9.

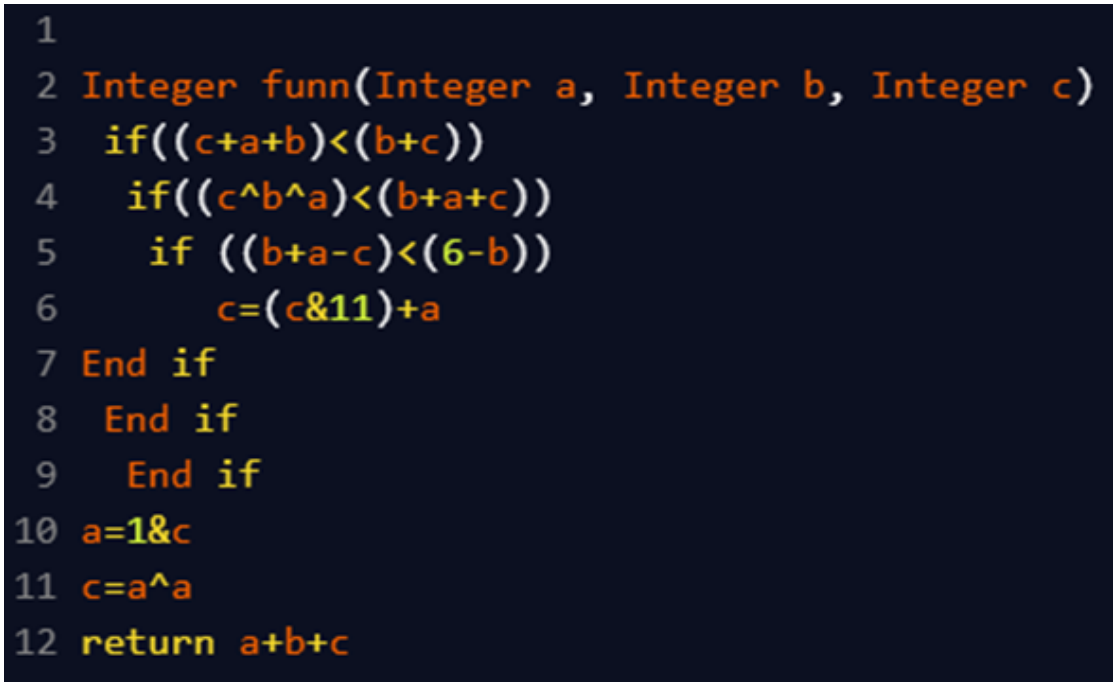
First if: Since (b + c) (11) is greater than (c - b) (7), c becomes 2.

Second if: (7 + 3) (10) is not less than (6 + a) (7), so the second if condition is false.

Output: The final values of a, b, and c are 1, 2, and 2, respectively. So, the output is 1 + 2 + 2 = 5

Answer: A

15. What will be the output of the following pseudo code for a=6, b=8, c=4?



```
1
2 Integer funn(Integer a, Integer b, Integer c)
3  if((c+a+b)<(b+c))
4   if((c^b^a)<(b+a+c))
5    if ((b+a-c)<(6-b))
6       c=(c&11)+a
7 End if
8  End if
9   End if
10 a=1&c
11 c=a^a
12 return a+b+c
```

Note- &: bitwise AND - - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 11

B. 5

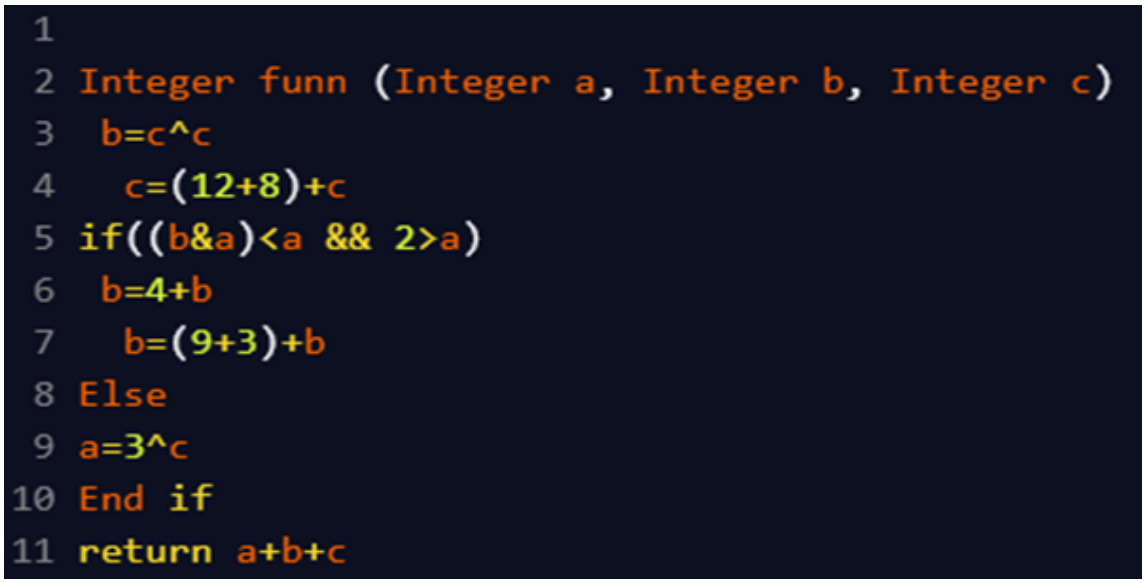
C. 27

D. 8

Explanation: The pseudo code evaluates several conditions, but the first condition ((c + a + b) < (b + c)) is false for the given values (a = 6, b = 8, and c = 4). As a result, none of the nested conditions are executed. After that, a is updated to 0 using a bitwise AND, and c becomes 0 using a bitwise XOR. The final return value is the sum of a + b + c, which results in 8.

Answer: D

16. What will be the output of the following pseudocode for a=3, b=4, c=4?



```
1
2 Integer funn (Integer a, Integer b, Integer c)
3  b=c^c
4    c=(12+8)+c
5 if((b&a)<a && 2>a)
6  b=4+b
7    b=(9+3)+b
8 Else
9 a=3^c
10 End if
11 return a+b+c
```

Note- &&: Logical AND - The logical AND operator (&&) returns the Boolean value true(or 1) if both operands are true and return false (or 0) otherwise.

&: bitwise AND - - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 51

B. 53

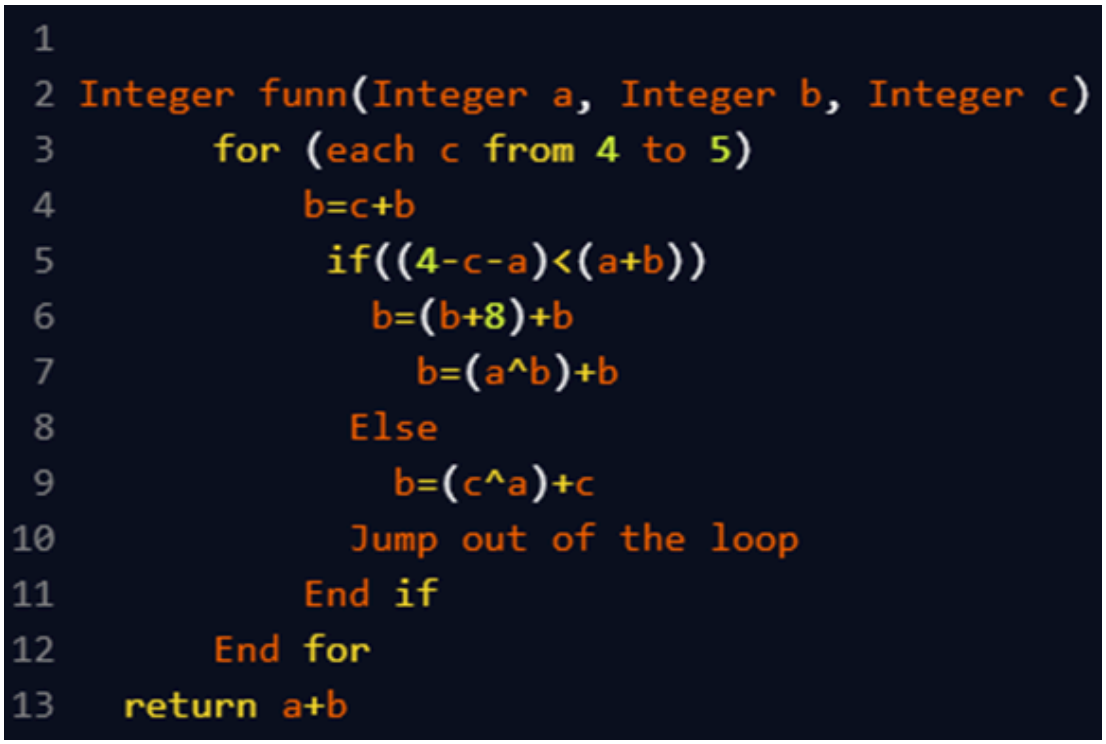
C. 56

D. 47

Explanation: The code first calculates b = 0 using the XOR operation c ^ c. Then, it updates c to 24 by adding 20 to the original c. The condition in the if statement is false, so the else block is executed, updating a to 27 using the XOR operation 3 ^ c. Finally, the return value is the sum of a + b + c, which is 51

Answer: A

17. What will be the output of the following pseudo code for a=1, b=6, c=5?



```
1
2  Integer funn(Integer a, Integer b, Integer c)
3        for (each c from 4 to 5)
4            b=c+b
5             if((4-c-a)<(a+b))
6               b=(b+8)+b
7                 b=(a^b)+b
8              Else
9                b=(c^a)+c
10             Jump out of the loop
11           End if
12       End for
13    return a+b
```

Note- ^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

A. 269

B. 279

C. 266

D. 262

Explanation: The code runs a loop where c takes values 4 and 5. In each iteration, b is updated by adding c, then further modified based on the if condition. The condition is true for both values of c, leading to successive updates of b. After the loop, the return value is the sum of a + b, which results in 266.

Answer: C

Accenture Common Applications and MS Office