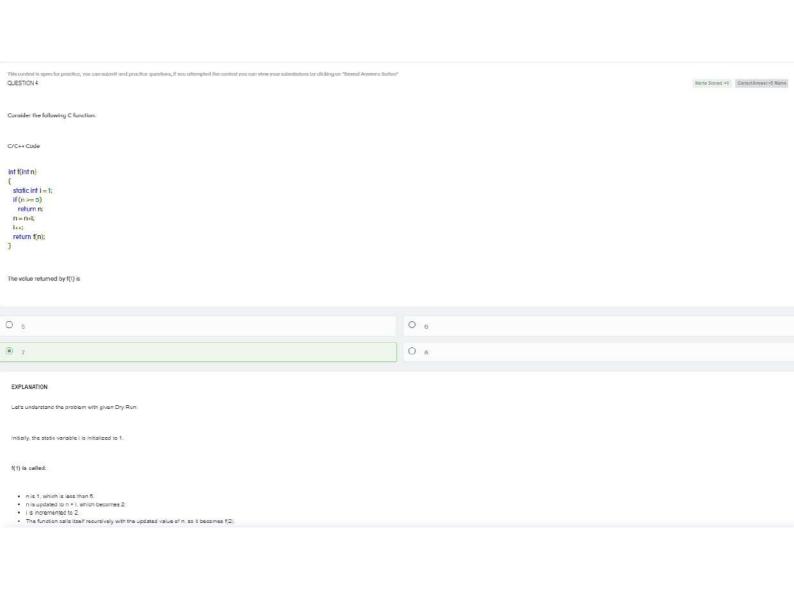


ar .	Marks Scores; +9 Coresci Ariswer; +5 Marks
O ciolonos	
10101103	
	© CHO10TIC1 10101101

QUESTION 3		Marks Gorest:+I Democi Annual:+I Marks
Let $T(n)$ be a function defined by the recurrence $T(n) = 2T(n/2) + 2n$ for $n \ge 2$ and $T(1) = 1$ Which of the following statements is TRUE?		
○ T(n) = 6(log n)	$\bigcirc T(n) = \theta(\sqrt{n})$	
▼ T(n) - θ(n)	$O = T(n) - \theta(n \log n)$	
EXPLANATION $n^{(\log_2 n)} = n \text{ which is } = n^*(1-5) + O(\operatorname{sqrt.n}) \text{ then by applying case } 1 \text{ of master method we get.} T(n) = O(n) \text{ Please refer http://www.geeksforgeeks.org/analysis-algorithm-set-4-master-method-solving-recurrences/ for more details.}$		



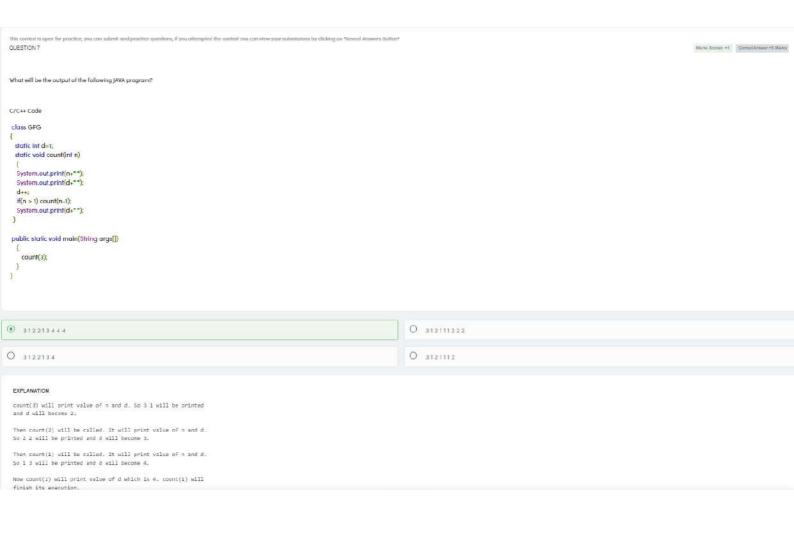
sl is open for proctice, you can submit and proctice questions, if you affempted the carriest you can view your submissions by clicking on "Seved Answers Button" QUESTION 5 Morte Somet +1 Cornet Answer +5 Marks Consider the following C function. C/C++ Code int fun (int n)(int x = 1, k; if (n --1) return x; for (k = 1; k < n; ++k) x = x + fun(k) * fun(n - k); return x; The return value of fun(5) is _____ O 0 O 26 ⑤ 51 0 71 EXPLANATION
$$\begin{split} & \text{fun}(S) = 1 + \text{fun}(1) * \text{fun}(4) + \text{fun}(2) * \text{fun}(3) + \\ & \text{fun}(3) * \text{fun}(2) + \text{fun}(4) * \text{fun}(1) \\ & = 1 + 2^*[\text{fun}(1)^*\text{fun}(4) + \text{fun}(2)^*\text{fun}(3)] \end{split}$$
Substituting fun(1) = 1 = 1 + 2*[fun(4) + fun(2)*fun(3)]Calculating fun(2), fun(3) and fun(4) fun(2) = $1 + fun(2)^{-1}fun(1) = 1 + 1^{-1} = 2$ fun(3) = $1 + 2^{-6}fun(2)^{-6}fun(2) = 1 + 2^{-1}2^{-2} = 5$ fun(4) = $1 + 2^{-6}fun(2)^{-6}fun(2$ Substituting values of fun(2), fun(3) and fun(4) fun(5) = 1 + 2*[15 + 2*5] = 51 Hence Option(C) is the correct answer.

Predict the Output: C/C++ Code class GFG static int f(int a[], int i, int n) Siture ...

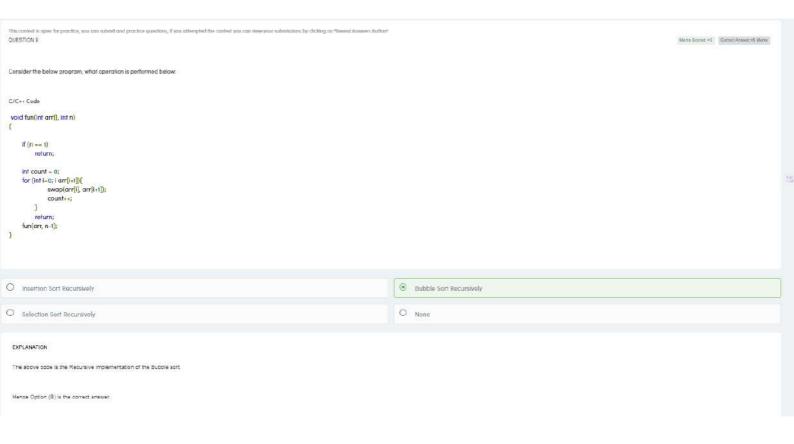
(
if(n < -0) return 0;
else if(C[i] % 2 = 0) return C[i] + f(a, i+1, n-1);
else return C[i] - f(a, i+1, n-1), public static void main(String args[]) (int a[] = (12, 7, 12, 4, 11, 6); System.out.print(f(a,0,6));) O -9 © 5 15 O 19 f() is a recursive function which adds f(a+1, n-1) to "a if "a is even. If "a is odd then it) subtracts f(a+1, n-1) from "a. See below recursion tree for execution of f(a, 6). F(add(12), 6) /*Since 12 is first element. a contains address of 12 */ 12 + f(add(7), 5) /* Since 7 is the next element, a+1 contains address of 7 $^{\prime\prime}$ / 7 - f(add(13), 4) 13 - f(add(4), 3)

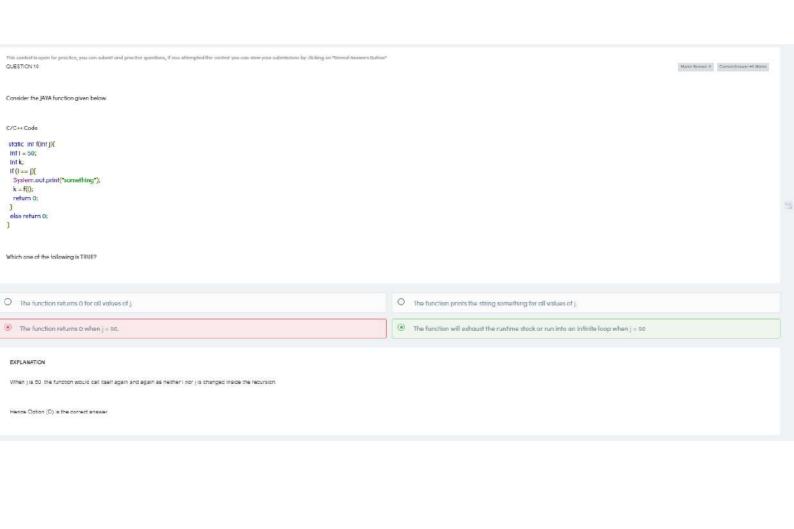
Marks Scored 4 Corned Arever:+1 Norte

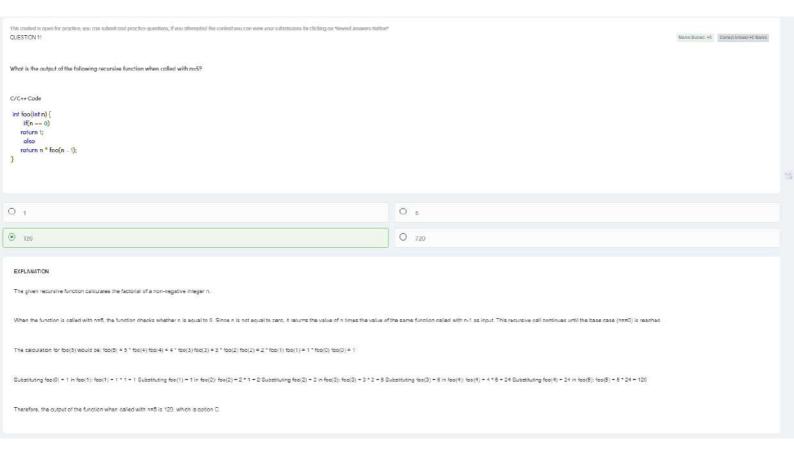
QUESTION 6



This contest is open for practice, you can submit and practice questions, if you alteropted the contest you can view your submissions by dicking on "ineveal Accesses's button' QUESTION 8	Matrix Scinics (S. CompatiAnness +5 Naints
$ \text{ tet } T(n) \text{ be defined by } T(1) = 10 \text{ and } T(n+1) = 2n+T(n) \text{ and for all integers } n \geq 1. \text{ Which of the following represents the order of growth of } T(n) \text{ as a function of the following represents the order of growth of } T(n) \text{ as a function of the following represents the order of growth of } T(n) \text{ as a function of the following represents the order of growth of } T(n) \text{ and } T(n+1) = 2n+T(n) \text{ and } T(n) = 2n+T(n) \text{ and } T(n+1) = 2n+T(n) \text{ and } T(n) = 2n+T(n) an$	tion of
• o(n)	O O(n log n)
⊙ (n²)	O O(n²)
EXPLANATION $T(n+1) = 2n + T(n)$ By substitution mathod: $T(n+1) = 2n + (2(n-1) + T(n-1))$ $T(n+1) = 2n + (2(n-1) + (2(n-2) + T(n-2)))$ $T(n+3) = 2n + (2(n-1) + (2(n-2) + T(n-3))))$ $T(n+3) = 2n + (2(n-1) + 2(n-2) + 2(n-3) + T(n-3))))$ $T(n+2) = 2n + 2(n-1) + 2(n-2) + 2(n-3) + \dots + (2(n-2) + T(n-3)))$ $T(n+1) = 2n + 2n + 2n + 2n + 4 + 2n + 6 + \dots + 18$ $T(n+1) = 2(n+n) + 2(n+n) + 2(n+n) + 2(1+2+3+\dots)$ $T(n+1) = 2(n+n) + 2(n$	







his contect is open for procisce, you can submit and practice questions, if you afterspiled the contect you can view your submissions by clicking on "General Answers Buffor" IUESTICN 12.	Marks Scored: +5 Correct Arasmer +5 Maria	
et $f(n)$ and $g(n)$ be asymptotically non-negative functions. Which of the following is correct?		
$\theta (f(n)^*g(n)) = \min (f(n), g(n))$	$O = (f(n)^*g(n)) = \max (f(n), g(n))$	
0 a(f(n) + g(n)) = min(f(n), g(n))		

Case-1: When none of the fin) and g(n) are constant functions - In this case maxif(n), g(n)| <= fin) * g(n) so max(f(n), g(n)) can not provide a upper bound for fin) * g(n)|.
 Case-2: When both of the f(n) & g(n) are constant functions or when any one of the f(n) and g(n) is a non-zero constant function, in this case f(n) * g(n) = theta(maxif(n), g(n))|.
 Case-3: When at least any one of the f(n) and g(n) is 0. In this case f(n) * g(n) = fleta(maxif(n), g(n))|. Since maxif(n), g(n)|. COULD BE unable to give a lower bound.

EXPLANATION

Option (D) is correct.

This contest is open for practice, you can submit and practice questions, if you attempted the contest you can view your submissions by dicting an "Reyed Asswers Button" QUESTION 13	Marks Scoop: -5 Corest Armest + Warks
Consider the same recursive C++ function that takes two arguments	
C/C++ Code unsigned int foo(unsigned int n, unsigned int r) { if (n > 0)	
O 9	O 8
O 5	
EXPLANATION Soci513, 2) will return 1 + foo(256, 2). All subsequent recursive calls (including foo(258, 2)) will return 0 + foo(n/2, 2) except the last call foo(1, 2). The last call foo(1, 2) returns 1. So, the value returned by foo(512, 2) is 1 + 0 + 0 + 0 + 1. The function foo(n, 2) basically returns sum of bits (or count of set bits) in the number n. Hence (D) is the correct Answer.	

