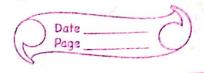
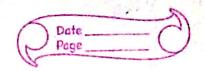


	BOOK Office and a frequency of the contract of					
(0)	What is algorithmo write an algorithm for fabonacci					
	Seriej.					
→	An algorithm is a set of algorithm- in step					
	by Step process for solving any kind of					
	by step process for solving any kind of problem. Basically it is the finite list of					
	instructions used to perform a task.					
100						
	Algorithm for fabonacci series:					
	Step 1: Start					
	Step 2: Declare the variables init, x, y, temp.					
	Step 3: Initialize the variables @ X=0, y=1 &					
	temp 20.					
	Step 4: Enfor the number of fabonacei Sovies					
	to be printed.					
	Step 5: print first two terms of Series.					
	Step 6: Use loop for following steps:					
	· temp = x+y					
	• N=y					
	· j= temp					
	• j+=1					
	· print value of temp.					
The Control						
	step + : End					



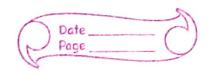
-		
(D . 11 Te.)	Difference between	Ourue & Stack data structure
	in detail with fig	The Exbrain gifferent obserquent
0 012	Envolved on those	data structure and
		1189 22.209 41.012 per
		1; janques incleson
①	Stack is an abstract	(3) Queue is also an abstract
	datatypes represented	datatypes similar to stack
	by the physical stock	severes itigis obn. and
	at the sontities	both ends. Truta
10 mil	where insertion for	11/0-11 sir/s(1:27)12
# 1=1	deletion takes place	de estimiting : Frote
1	from some end.	
		nin 311 10/13: 1- 1/012
(<u>a</u>)	It is based on	3.94 32 based on first
	last in first out	in first out (FIFO) -
	(LIFO) principale.	principle 19 7 921
	120112 1-6-11-	of good astrony to make
(3)	Push adds items	3 Enqueure adds items
	to the stack of	to queue & to rear f
	bob romard item	dequeré removes : tem from
	from the stack.	front of the queue.
		W Sair .
9	only one bourton	@ Two pointons are
	is used in stack.	used in a quence

Scanned with CamScanne



3	ч			6 front	year
	3	2	907 3	Ĩ	J
	2	9	<u> </u>	7 2	(9 1)
	,	6	- A-1		
	٥	5		Queu	C
	A.AX	stac	k		. —
4	Example: Stack of the			@ Example:	line of the
	books where we can				in food count.
	remove top book only.			distributi	07.
	2 February Cine and I was a man on the second				
	Operations performed in Stack.				
	O Push: 1 It adds an items in the stack. 2f				
	Stack is full it is said to be overflow				
	Condition.				
	2) pop: It removes items from the stack. The				
					reversed order
	in which they are pushed. If stack is				
	in which they are pushed. If stack is Null it is sald to be Underflow condition				
	(3) Per	21=	e peturns top	element of	the Stack.
	(u) 20 i	's En	opty: Returns	true if stack	is empty, else
			False.		is empty, else
			Returns true false		

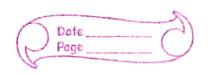
•	Operations on Queue
0	Enquere: Adds an item to the quere. if quere
	is full. then it is said to be -overflow
	Condition.
	91119
(2)	Dequeue: Removy an item from the queue. If
	Queue is Null than it is said to
	be underflow andition.
Alexan !	end of facture for our simple adjoint
3	peek: Returs top element.
©	7 470.00
<u>(3)</u>	Is Empty: Returns true it Queue is Null & vicevous.
17 -3	india and an about the same of
Contagor	and all him at the them of streets.
	. * * (1.1) * * * * * * * * * * * * * * * * * * *
027 -	duren in and month woman is and a
	reading get in the myer, governous
i die	to de la desta con aposte de pro- est
I' have	introduction of all the strate miles
	2 cl' de l'amorto nel morton : el m
Ka alfal	to of the fact and monthly in the first
	The state of the s



Q.3 Hsymptotic Notations - Heymptotic Notations are the mathematical notations used to describe the running time of an algorithm when the Enput tends towards a particular value or a limiting value. tor example: In bubble sort, when the input array Is already sorted, the time taken by the algorithm is linear i.e the boot case. But open the subat arread it is leverse congistion the algorithm takes the maximum time to sort the elements i.e the worst case. When the input array is neither sorted nor in reverse order than it to takes average time. These durations are denoted using Haymptotic Usually the time taken by an algorithm falls rugar 2 Abra. O Best case - Minimum time required for program exe-@ Woist are - Marimum time required for budran excorpion. (3) Average case - Average time required for biodiow exconfigur.



	Following are the commonly used asymptotic
100/10/20	notations to calculate the running time comple.
Allasper ri	xity of an algorithm.
the large to be	and the state of t
0	O notation
3	or notation
(3)	O. notation
millinge	come and restable and for the property of
But I have	Big on notation o
e." 1	The notation O(1) is the formal way to express
	the upper bond of an algorithms running time
100	It measures the worst are time complexity
	can possibly take to an algorithm
200	can possibly take to complete.
	- La Contraction
211-4	Liente the Later to the second to
	(10)
The works	7 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1
100 100 100 22	the state of the s
And the second	(-17) your production of the state of the state of
	the second are programmed to the second



	for example, for a function f(n)
	O(f(n)) = {g(n): there exists cro & no such that
	f(n) < c.g(n) for all n>no.}
	and the state of t
	Omega Notation (S):
7	The notation I(n) is the formal way to express
	the lower bound of an algorithms are running. time.
	It measures the best case of time complexity or
	the best amount of time an algorithm can possibly
	take to complete.
	A STATE OF THE STA
	Eco)
3	1/13/10 = (10) 1 di 100 600 17 300 1 37 300 1 37 300 1
	19 10 10 10 101 (cont) of alofe
	The state of the s
	for example: for a function f(n)
	1 (fens) 2 { g(n): there exists cro and no sach that
	g(n) = c. f(n) for all n>nois



للسلسا		
	Theta notation, o	
fort !	The notation O(n) is the formal way to express be	1
	the lower & upper bound of algorithms -running times	
	9+ is represented as follows:	
	f(n)	
spare was	- 1 year 4 month and of 500 (/ cm) / 2 g(n) - 100	
om U. L.	was in the color of the format winds the second	
ar k	Manne will a surfair of the window with	
A Sugar	assimply the ac spall of land to speed the	
	1 1 in stands	
		f
7		
	Ofen) = { g(n) of and only if g(n) = 0(f(n)) }	
	$ \frac{\partial f(n)}{\partial f(n)} = \partial$	