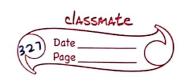
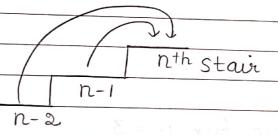
| | 11/03/2023 |
|---------|--|
| | |
| | Questions on Recursion |
| | |
| QI | Climb Stairs destination |
| | J |
| 4 . 4 . | 2 nth Stavi |
| | |
| | |
|)-21 | CHILL STATE OF THE |
| -1-5 | Loowice |
| | Steps allowed > move I stair at a time |
| | move a stair at a time |
| | A DOMESTIC TO THE STATE OF THE |
| | Find the minimum number of steps to reach 1th stair. |
| | reach nth stair. |
| | |

Scarineu with Cam



We are cur rently on the oth stair. We just have to solve one case & rest recursion will handle.



To reach the nth stair, we can reach from (n-1)th Stair or (n-2)th Stair

nth stair =
$$(n-1)$$
th stair + $(n-2)$ th stair $f(n)$ = $f(n-1)$ + $f(n-2)$

5) As we have tofind total ways

The recursive relation is some as that of the fibonacci servies.

(ode

Int climb Stairs (int n) {

// Base Case

if (n = = 0 || n = = 1)

return 1;

// Recursive relation

return climb Stairs (n-1) + climb Stairs (n-2);

3

Understanding base case

| > | (only for n = 0 case) |
|---------|--|
| MADLE | 1 way way |
| | |
| | 1st stavi |
| | 0th stair |
| | n=0, way $n=1$, 1 way |
| | 0 |
| | Dry run for $n = 3$ |
| | The answer for no of ways to reach the 3rd stair is 3 |
| | 3rd stair is 3 |
| 40 | Try party was pipte still and a war T |
| | 1st solution is a second secon |
| | |
| الاطالة | Ist way |
| | (1, (1, 1, 1) |
| | |
| 911 1 | |
| 5.7 | 2nd Solution same en mortolar avisco de |
| | |
| | 2nd way |
| () | 2nd way (1,2) |
| | |
| | fin ani sante Januari |
| | 3rd solution |
| | 3rd way (2,1) |
| | 3rd Way |
| | (2,1) |
| J 1 1 | 11 (1-10) 2 min 12 min 12 min |
| (2) | |
| N 12 | The most confusing thing in the question is the base case for 0th stair, that is only I way is there to reach 0th stair. |
| | the base case for 1th stains that |
| | I wou is there to reach oth ctain |
| | Truly. |

Scarineu with cam

classmate Time Limit Exceeded Vote There will many TLEs in case of recursion which will further be optimized. f(5) = f(4) + f(3)f(3) + f(2) $\rightarrow f(3) + f(1)$ $\hookrightarrow f(1) + f(0)$ Similary for f (3) in 1st line, calls will be sent See the recursion tree below. 5+3=8f(5) f(4) $\rightarrow f(3)$ 175(2) f(3) ← f(1)R 1 f(1) f(1)f(0) f(1)f(0)f(1) f(0) Hence to reach the 5th Stair, there are 8 possible ways. Note - f(0) = 0 is not selected as it means that we can't reach the oth Step/Stair but we are already on oth Stair & hence we take it as I. 2 Print the elements of the away. We will simply take the base case as if index becomes equal to size of arriay, we need to return. After that simply print Scarineu with CaM

| 4:19 | the element & rest recursion will handle. |
|--------|---|
| | · Lastant Di sa ridin di la |
| | Code |
| | |
| | void print (int aver [], int i's int size){ |
| | //Base Case |
| | if (i = = size) |
| 33 | retwin |
| | 1/ Processing |
| | cout << au [i] << " "; |
| | // Recursive Relation |
| | print (arr, i+1, size); |
| ± 7 | 3 |
| (| index |
| (4) | Tree 1:00 (2) To the board of the |
| | |
| | f (our, o, size) |
| | J cout << aur [0]; |
| | f(our, I, size) |
| | J. cout << aux [1]; |
| 2 0+ | f(avr, 2, size) |
| | L cout << avr [2]; |
| | |
| Jr. A | goes on until base |
| 3 1/ | i J condition is achieved |
| #1 5 | Contract & Ranks of Conglished |
| | Alternative code |
| | 11 1 1 1 (1011 011 57 |
| | void print (int aur [], int size) { // Base (ase |
| | $ \frac{1}{ f(size = 0) } $ |
| ≥ \/ √ | return; |
| dais. | i etwin , |
| | Scarneu with G |

ocanneu wini dam

ocanneu with carn

| | classma | Ł _a |
|-----|--------------|----------------|
| 332 | Date Page | |

| 1 % | Page Page |
|-------------|--|
| | |
| | retwini |
| 5 | // Processing |
| | if (mini > aur [i]) |
| | mini = our [i]; |
| | // Recursive Relation |
| | find Min (aur, i+1, size, mini) |
| <u> </u> | 31 of the traverse senon MIMITINE IN THE |
| - E 11 | Les and Le to di goods Will state promise |
| 100 | The approach is same as that of Q3 but |
| | the if condition has changed. |
| 0- | |
| $-\omega_5$ | Find whether key is present in string or |
| | noten Dan Dan Indiana |
| | Juxom seni |
| | 1/p→ love babbar, Rey = ey, |
| | υ/ρ ¬ 1 ω |
| | Will be done by recursion |
| | $\parallel \backslash \land \land \lor \lor$ |
| | 4 Checking this only (1st case solve) |
| | |
| | Base condition => 9f we have traversed |
| | St whole string, then simply return false. If while traversing, we have traversed |
| | If while traversing, we found the key then |
| | simply return true otherwise we move on to the next index. |
| J* 1 | |
| - 601r. | the set of some one of the set |
| | |
| | bool checkkey (string strain) |
| | int key) s |
| 4 | bool checkkey (string stroint i, int no int key) { // Base condition |
| | int key) { // Base condition if (i = = n) return false; |
| | return falso: |
| | 19:000 |
| | II |

Scarineu with cam

// I case solve

if (sty[i] = = key)

return true;

// Recursive relation

return checkkey (str, i+1, n, Fey);

3 not mandatory

4) move to next

index.

Note -> By doing &, TLE can get eliminated. So here string can be passed as reference.

Here we can return index of the key in String. Just change bool type to int, false to -1 and true to i. 9fthere are multiple occurrences of the key in the string, then we can store those indexes in vector.

Q6 Find the digits of the number.

1/b → 647

0/p - digits

6

4

7

647%10 = 7 g 1st digit

647/10 = 64

64 % 10 = 4 3 2nd digit

64/10 = 6

6%.10 = 6 3 3rd digit

6/10 = 0 → stop

| Base condition > When n becomes 0, We need to Stop. Code Void print Digits (int n) { // Base case if (n = 0) | | |
|---|------------|---|
| Void print Digits (int n) { // Base case if (n = 0) Yetwin; // Recounsive Relation print Digits (n/10); // Processing int digit = n % 10; cout << digit << " "); 3 Tree PD (647) PD (64) Print PD (0) Base case order 6 4 7 Will be printed which are | | Base condition => When n becomes 0, |
| Code Void print Digits (int n) { // Base case if (n = 0) | | We need to stop. |
| Void print Digits (int n) { // Base case if (n = 0) | | September 19 Comment of the Comment |
| void print Digits (int n) { // Base case if (n = = 0) | ζ. | Coden III (1929) judgament muli |
| // Base case If (n = = 0) Yetwin; // Recursive Relation print Digits (n/lo); // Processing Int digit = n % 10; cout << digit << ""; 3 Tree PD (64) 4 PD (6) Frint PD(0) Base case order 6 4 7 Will be printed which are | | |
| // Base case If (n = = 0) Yetwin; // Recursive Relation print Digits (n/lo); // Processing Int digit = n % 10; Cout << digit << ""; 3 Tree PD (64) 4 PD (6) Frint PD(0) Base case order 6 4 7 Will be printed which are | | void print Digits (int n)? |
| return; // Recursive Relation print Digits (n/10); // Processing int digit = n % 10; cout << digit << ""; } Tree PD (64) 4 () Print PD (0) Base case order 6 4 7 Will be printed which are | 4 · j. | aly doing & . The no ret eliminates |
| return; // Recursive Relation print Digits (n/10); // Processing int digit = n % 10; cout << digit << " "); 3 Tree PD (647) PD (64) 4 | | |
| // Recursive Relation print Digits (n/10); // Processing int digit = n % 10; cout << digit << ""; 3 Tree PD (647) PD (64) 4 PD (6) 6 Print PD (0) Base case order 6 4 7 Will be printed which are | | |
| print Digits (n/10); // Processing Int digit = n % 10; cout << digit << " "); 3 Tree PD (64) 4 PD (6) Print PD (0) Base case order 6 4 7 Will be printed which are | | nesses and de a returnion and on the |
| // Processing Int digit = n % 10; Cout << digit << " "; 3 Tree PD (647) PD (64) Print PD (0) Base case Order 6 4 7 Will be printed which are | | |
| // Processing Int digit = n % 10; cout << digit << " "; 3 Tree PD (64) PD (64) Print PD (0) Base case order 6 4 7 Will be printed which are | <i>3</i> 0 | print Digits (n/10); |
| Tree 7 PD (647) PD (64) 4 PD (6) Print PD (0) Base case order 6 4 7 Will be printed which are | . po | 1/Processing |
| Tree 7 PD (647) 4 PD (64) 4 PD (6) 6 Print PD (0) Base case order 6 4 7 Will be printed which are | ·VE | int digit = n % 10 i |
| Tree PD (647) PD (64) PD (6) Print PD (0) Base case order 6 4 7 Will be printed which are | | cout << digit << " ") |
| PD (647) PD (64) PD (64) PD (6) Print PD (0) Base case order 6 4 7 Will be printed which are | | 3 Rudmit adt la existe adt hu |
| PD (647) PD (64) PD (6) Print PD (0) Base case order 6 4 7 Will be printed which are | | Tree |
| PD (64) 4 () PD (6) Print PD (0) Base case order 6 4 7 Will be printed which are | | 1 + 2 + 4 |
| Print PD(0) Base case order 6 4 7 Will be printed which are | | 7 7 PD (647) |
| Print PD(0) Base case order 6 4 7 Will be printed which are | | |
| Print PD(0) Base case order 6 4 7 Will be printed which are | | |
| Print PD(0) Base case order 6 4 7 Will be printed which are | | |
| Print PD(0) Base case order 6 4 7 Will be printed which are | | |
| order 6 4 7 Will be printed which are | | |
| 6 4 7 Will be printed which are | | |
| 6 4 7 will be printed which are the digits of number 647. | | OTAGE |
| the digits of number 647. | | (4 7) |
| The digits of number 647. | | 6 9 Will be printed which are |
| | | the digits of number 647. |
| | | 0 = 01.10 |
| | | C = 01 \s |
| | V | Scarineu wit |

Scarineu with dam