



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 23ITT101 Problem solving and C programming

I YEAR /I SEMESTER

Unit 4- Functions and pointers

Topic 4:Recursion





Brain Storming



1. What is recursive function?



Recursion



It is a programming technique where a function calls itself repeatedly to solve a problem by breaking it down into smaller parts:

- ✓ Explanation

A function that calls itself is called a recursive function, and the process of calling itself is called a recursive call.

- ✓ How it works

The function performs part of the task and delegates the rest to another instance of itself. This process continues until a base case is reached, which stops the function from calling itself and resolves the problem.

- ✓ Benefits

Recursion can be an efficient way to solve complex problems by simplifying them into sub problems. It can also make code easier to read and write.

- ✓ Termination condition

It's important to include a termination condition, or base case, to prevent the function from calling itself forever and getting stuck in a loop



Example Program



```
#include<stdio.h>
void binary_search(int a[], int low, int high, int key)
{ int mid; mid = (low + high) / 2;
if (low <= high)
  { if (a[mid] == key)
printf("Element found at index: %d\n", mid);
  else if(key < a[mid])
binary_search(a, low, mid-1, key);
  else if (key>a[mid])
  binary_search(a, mid+1, high, key); }
else if (low > high)
printf("Unsuccessful Search\n"); }

void main()
{ int i, n, low, high, key;
n = 5; low = 0; high = n-1;
int a[10] = {12, 14, 18, 22, 39};
key = 22;
binary_search(a, low, high, key);
}
```

Output:
Element found at index: 3



Assessment 1



1. Write about application of recursive function?

Ans : _____





References



1. Reema Thareja, “Programming in C”, Oxford University Press, Second Edition, 2016

Thank You