CS & IT ENGINEERING

'C' Programming

Functions

Lecture No.- 04

By- Satya sir

Recap of Previous Lecture

Minimum IQ in GATE

- Types of **Recursion**
  - Direct Recursion
    - Head Recursion
    - Tail "
    - Tree "
    - Nested Recursion
  - Indirect Recursion

Topics to be Covered

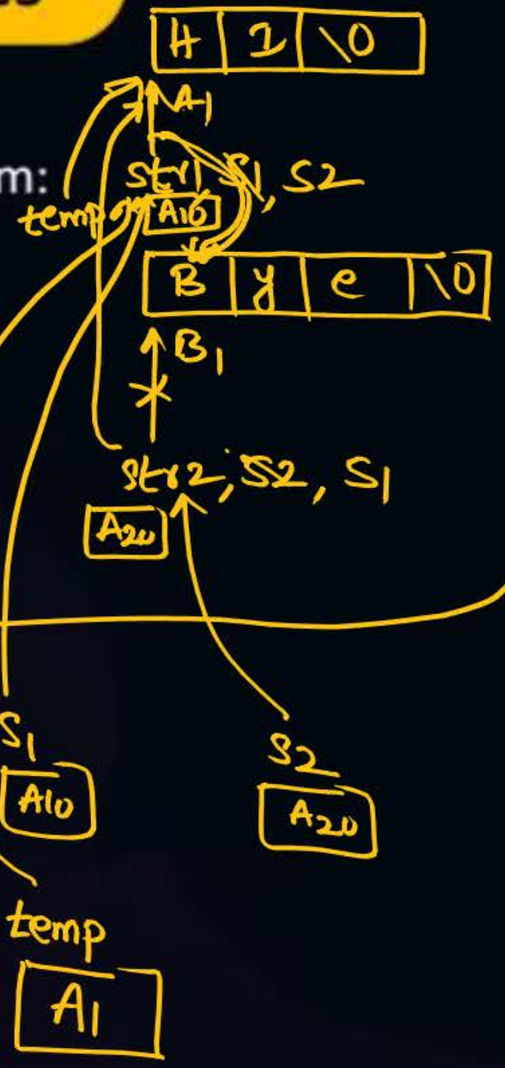
- PYQ Practice on Recursion
- Storage classes
- Static & Dynamic Scoping

Topic : Storage Classes

GATE 2018

#Q. Consider the following C program:

```
#include <stdio.h>
void fun1(char *s1, char *s2) {
char *temp;
temp = s1;
s1 = s2;
s2 = temp;
}
void fun2(char **s1, char **s2) {
char *temp;
temp = *s1;
*s1 = *s2;
*s2 = temp;
}
```




```
int main() {
char *str1 = "Hi", *str2 = "Bye";
fun1(str1, str2);
printf("%s %s", str1, str2);
fun2(&str1, &str2);
printf("%s %s", str1, str2);
return 0;
}
```

Hi, Bye

Bye Hi

The output of the program above is

A. Hi Bye Bye Hi ✓

B. Hi Bye Hi Bye

C. Bye Hi Hi Bye

D. Bye Hi Bye Hi

Topic : Storage Classes

GATE 1998

#Q. Consider the function

```
int fun(x: integer)
{
If x > 100 then fun = x – 10;
else
fun = fun(fun(x + 11));
}
```

For the input x = 95, the function will return ___

ANS: 91

```
int fun(int x)
{
  if (x>100) { return (x-10); }
  else return fun(fun(x+11));
}
```

return 91

x>100
= fun(101)
fun(fun(111))

fun(95) x=95
↳ fun(fun(106))
= fun(96) x=96
↳ fun(fun(107))
⇒ fun(97) x=97
↳ fun(fun(108))
= fun(98)
↳ fun(fun(109))
= fun(99)
↳ fun(fun(110))
= fun(100)

fun(106) x=106 return 96	fun(109) return 99
fun(107) x=107 return 97	fun(110) return 100
fun(108) x=108 return 98	fun(111) return 101

Topic : Storage Classes

GATE 2021

(NAT)

#Q. Consider the following ANSI C program

```
int foo(int x, int y, int q)    {
if ((x<=0) && (y<=0))
return q;
if (x<=0)
return foo(x, y-q, q);
if (y<=0)
return foo(x-q, y, q);
return foo(x, y-q, q) + foo(x-q, y, q);
}
```

```
int main( )
{
   int r = foo(15, 15, 10);
   printf("%d", r);
   return 0;
}
```

The output of the program upon execution is

_____

Topic : Storage Classes

15 15 10

```
int foo(int x, int y, int q)    {
if ((x<=0) && (y<=0))
return q;
if (x<=0)
return foo(x, y-q, q);
if (y<=0)
return foo(x-q, y, q);
return foo(x, y-q, q) + foo(x-q, y, q);
}
```

x y q

foo(15, 15, 10)

foo(15,5,10) + foo(5,15,10)

foo(15,-5,10) + foo(5,5,10)

foo(5,-5,10)

foo(-5,-5,10)

return 10

foo(5,5,10) + foo(-5,15,10)

foo(5,-5,10) + foo(-5,5,10)

foo(-5,-5,10) return 10

foo(-5,-5,10) return 10

foo(-5,5,10)

foo(-5,-5,10)

return 10

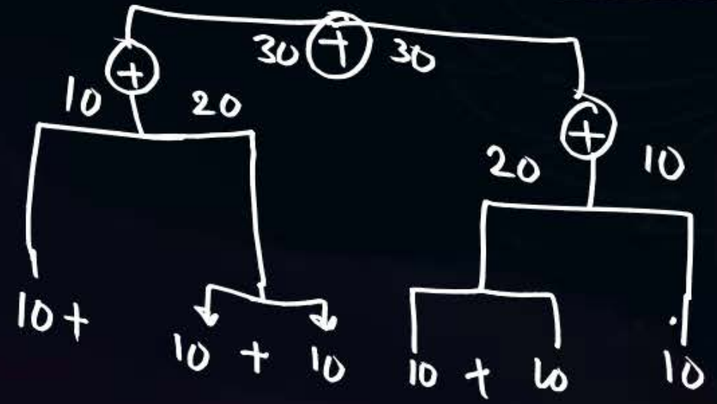
foo(5,-5,10) + foo(-5,5,10)

foo(-5,-5,10)

foo(-5,-5,10)

return 10

return 10



Return value = = 60

Topic : Storage Classes

GATE 2005

MCQ

#Q. Consider the following C-program :

```
void foo (int n, int sum){
int k = 0, j = 0;
if (n==0) return;
k = n % 10; j = n /10;
sum = sum + k;
foo (j, sum);
printf ("%d\t", k);
}
int main() {
int a = 2048, sum = 0;
foo (a, sum);
printf ("%d\t", sum);
}
```

Which does the above program print ?

A. 8, 4, 0, 2, 14
B. 8, 4, 0, 2, 0
C. 2, 0, 4, 8, 14
D. 2, 0, 4, 8, 0 ✓

foo(2048, 0) — n, Sum
k=8, j=204
Sum = 0+8=8
foo(204, 8) — n, Sum
Print 8
k=4 j=20
Sum=8+4=12
foo(20, 12) — n, Sum
Print 4
k=0, j=2
Sum=12+0=12
foo(2, 12) — n, Sum
Print 0
k=2, j=0
Sum=12+2=14
foo(0, 14) — n, Sum
Print 2
return

a: 2048
Sum: 0

o/p: 2 0 4 8 0

Topic : Storage Classes

GATE 2019

#Q. Consider the following C program :

```
#include<stdio.h>
int jumble(int x, int y){
   x = 2*x+y;
   return x; }
int main(){
   int x=2, y=5;
   y=jumble(y,x);
   x=jumble(y,x);
   printf("%d \n",x);
   return 0;
}
```

jumble (5, 2)
x = 2×5 + 2
= 12

jumble (12, 2)
x = 2×12 + 2
= 24 + 2
= 26

The value printed by the program is 26.

Topic : Storage Classes

GATE 2005

NAT

#Q. What is the output printed by the following program?

```
#include<stdio.h>
int f(int n, int k)
{
if (n == 0)
return 0;
else if (n % 2)
return f(n/2, 2*k) + k;
else return f(n/2, 2*k) - k;
}
int main ()
{
printf("%d", f(20, 1));
return 0;
}
```

if (n % 2) == if(1) for odd 'n' value
= if(0) for even 'n'

$f(n=20, k=1)$
↳ if(0) False
$f(n=10, k=2) - 1$
↳ $f(n=5, k=4) - 2$
↳ $f(n=2, k=8) + 4$
↳ $f(n=1, k=16) - 8$
↳ $f(n=0, k=32) + 16$
return 0

$0 + 16 - 8 + 4 - 2 - 1$
$= 20 - 11$
$= 9$

Topic : Storage Classes

- Storage class defines
  1) Default value
  2) Memory Location
  3) Scope : The extent (limit) up to which, a variable can be accessed
  4) Lifetime of a variable.

Ex: ① int i;
Printf("%d", i); // what will be default value?

② int i; // Let 1 int = 4 Bytes
char j; // 1 Byte
float k; // 4 Bytes

} In which memory (RAM, ROM, HD, Cache Memory, CPU Registers) Space allocated?

```
int k;
{
   int i;
   // i accessible?
   // k accessible?
   // j accessible?
}
{  int j;
   // i, j, k accessible?
}
i, j, k accessible?
```

Topic : Storage Classes

Scope : Access limits

a) Local scope (or) Block scope ⇒ with in the current block, in which it is declared

b) Global scope (or) File scope ⇒ Variable can be accessed, through out Program (In Any Block, outside blocks)

```
int j=10; // Global variable ⇒ A Variable, declared outside all blocks; Also called as External Variable.
void main( )
{
    int i=20; // Local variable ⇒ A Variable, Declared with in Some block, Also called as Internal Variable.
    Printf("%d %d", i, j); // Valid
    f( );
}
void f( )
{
    int k=30;
    Printf("%d %d", k, j); // Valid
    Printf("%d", i); // Invalid
}
```

Topic : Storage Classes

If Local and Global variables have same name, Then always Local variable > Global variable.

Scoping can be implemented in 2 ways :

① Static scoping : Local variable, Global variable →

② Dynamic scoping : Local variable, Recent invoked function local variable ----, Global variable →

Static Scoping Vs Dynamic Scoping

Example

```
int a=10, b=30;
void f( )
{   int a=20;
    b= b+a;
    Printf("%d %d \n", a,b);
    g( ); Printf("%d %d \n", a,b);
    h( );
    Printf("%d %d \n", a,b);
}
void g( )
{   int b=40;
    a= a+b;
    Printf("%d %d \n", a,b);
    h( ); Printf("%d %d \n", a,b);
}
void h( )
{
    Printf("%d %d \n", a,b);
}
void main( )
{
    int a=5, b=7;
    Printf("%d %d \n", a,b);
    f( );
    Printf("%d %d \n", a,b);
}
```

a: 10 → 50 ; b: 30 → 50

a: 20 ; b = 30+20 = 50

a: 20 → 60 ; b = 7+20 = 27

b: 40 ; a = 10+40 = 50 ; a = 20+40 = 60

a: 5 ; b: 7 → 27

```
// 20 50   // 20 27
// 20 50   // 60 27
// 20 50   // 60 27
// 50 40   // 60 40
// 50 40   // 60 40
// 50 50   // 50 50
```

f() g() h() f() main()

o/p:

5	7
20	27
60	40
60	40
60	40
60	27
60	27
60	27
5	27

o/p:

5	7
20	50
50	40
50	50
50	40
20	50
50	50
20	50
5	7

Topic : Storage Classes

ISRO 2017

To be Contd ...

#Q. What is the output of the following program?

```
#include<stdio.h>
int tmp=20;
main()
{
  printf("%d", tmp);
  func();
  printf("%d", tmp);

}
func()
{
 static int tmp=10;
 printf("%d", tmp);
}
```

a) 20 10 10
b) 20 10 20
c) 20 20 20
d) 10 10 10

Topic : Storage Classes

GATE 2019

#Q. Consider the following C program:

```
#include <stdio.h>
int r() {
      static int num=7;
      return num--;
}
int main() {
      for (r();r();r())
            printf("%d",r());
      return 0;
}
```

Which one of the following values will be displayed on execution of the programs?

a) 41 b) 52 c) 63 d) 630

Topic : Storage Classes

GATE 2020

#Q. Consider the following C functions.

The return value of fun2(5) is _____

```
int fun1(int n) {
   static int i= 0;
   if (n > 0) {
      ++i;
     fun1(n-1);
  }
 return (i);          }
int fun2(int n) {
  static int i= 0;
  if (n>0) {
     i = i+ fun1 (n) ;
     fun2(n-1) ;
 }
return (i);          }
```

2 mins Summary

- PYQ Practice on Recursion
- Storage classes
- Static Vs Dynamic Scoping

THANK - YOU