C-Programming and DS

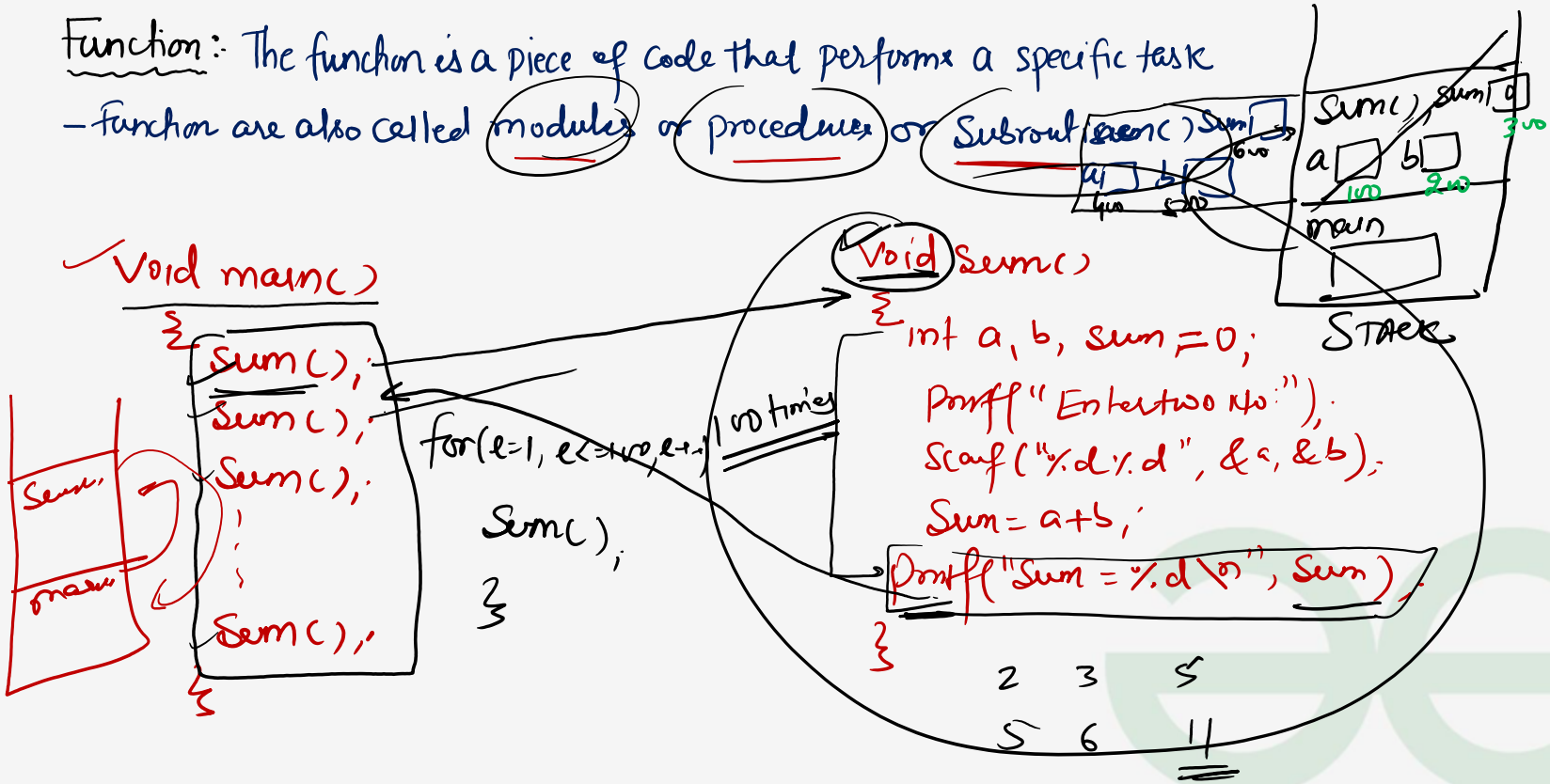
Today Class Topics

-Function — Recursion
-Storage classes
  -Auto
  -Register
  -Static
  -Extern

C-Programming and DS

Function :- The function is a piece of code that performs a specific task

- Function are also called modules or procedures or Subroutines

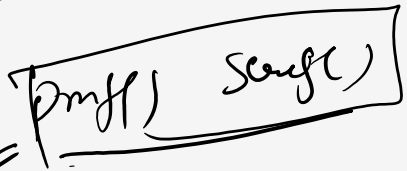


Function :-

1) The Purpose of the function is code re-utilization

2) The Purpose of the loop is code repetition

3) The Purpose of header file is function reutilization



4) If we want to reuse the function then we need to create header file of that particular function and include that in the required Program

C-Programming and DS

Function having following Parameter:

Return type | funchnname (set of inputs, ...);

int funname(set of inpets);

1) Prototype / Declaration → void swap (int, int);

2) Function Calling → swap (10, 20); ← Actual parameter

funchnname (set of inpets);

3) Function definition → void swap (int i, int j)

{

}

formal param.

Prototype Contains:-

i) Name of function ✓

ii) return type of function ✓

iii) no. of Argument in function ✓

iv) type of Arguments in function ✓

C-Programming and DS

Question:

```
int add(int, int); /* Declaration */
void main()
{ int x=10, y=20, z;
  z = add(x,y); /* Calling */
  Printf("%d", z);
}
int add(int a, int b) /* Definition */
  { int c;
    c=a+b;
    return c;
  }
```

Actual parameter

formal parameter

z | 30

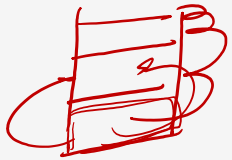
30

a | 10

b | 20

c | 30

C-Programming and DS



DS

Recursion :-

1) Function calling itself is called recursion — STACK

2) Recursion should definitely have termination condition otherwise it leads to stack overflow

3) Recursion internally makes use of STACK

4) Everytime when the recursive call is made then activation record will be created in the stack

Activation record :-

1) The stack frame allocated to the recursive function call is called as activation record

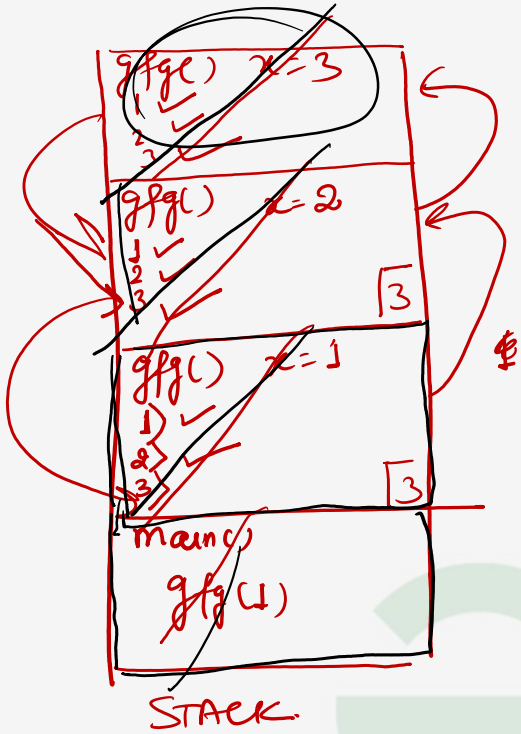
2) The creation and deletion of activation records depends on actual function calling sequence

C-Programming and DS

Question: What is the output Printed?

```
Void gfg(int x)
{
1) printf("%d", x);
2) if (x <= 2)
       gfg(x+1);
3) printf("%d", x);
}
Void main()
{
   gfg(1);
}
```

123321



C-Programming and DS

GATE Question:

```
void foo(int n, int sum)
{
1) int k=0, j=0;
2) if (n==0) return;
3) k = n%10; j = n/10;
4) sum = sum+k;
5) foo(j, sum);
6) printf("%d", k);
}
```

```
int main()
{
1) int a=2048, sum=0;
2) foo(a, sum);
3) printf("%d", sum);
}
```

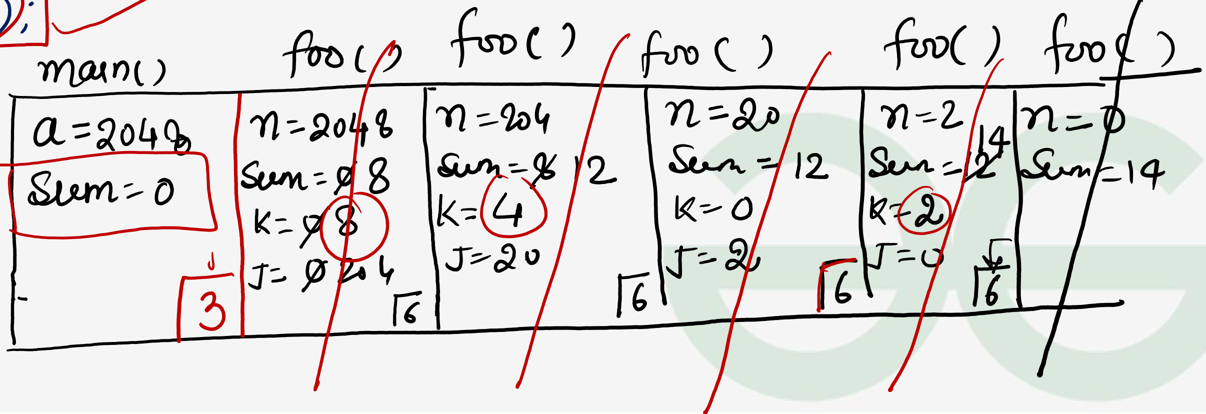
Ans ⇒ 20480

2%10 = 2

2048 % 10

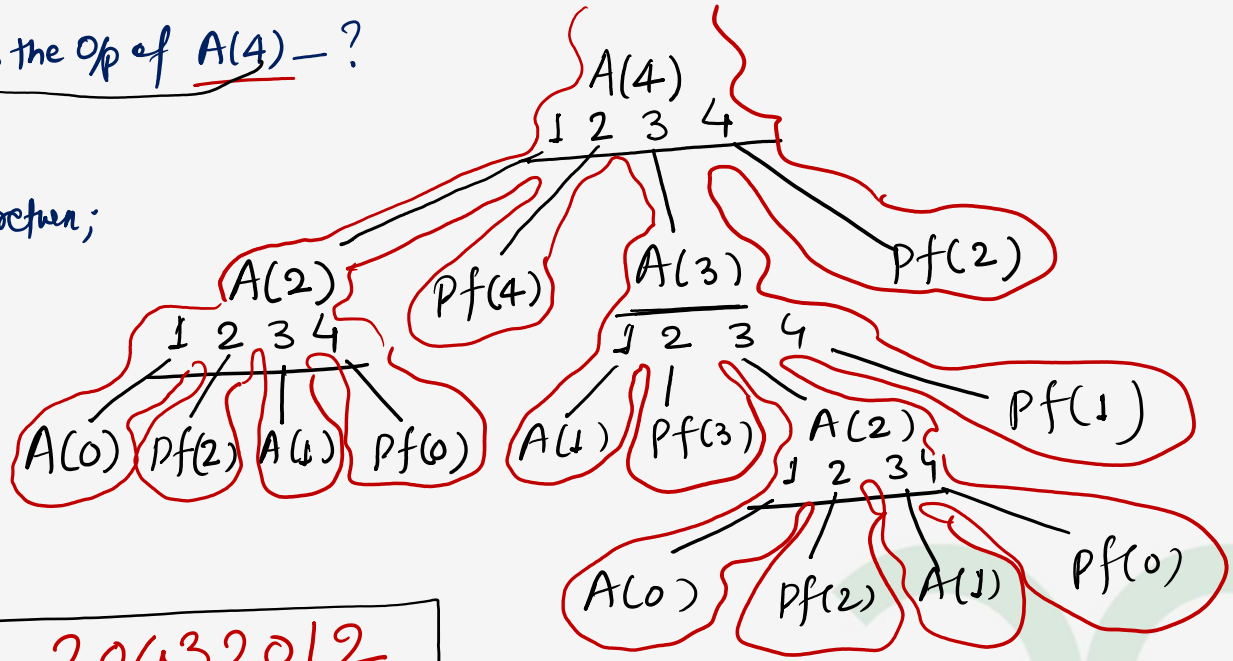
$J = \frac{n}{10} = \frac{2048}{10} = 204$

STACK



Question: What is the O/p of A(4) – ?

```
A(int n)
{
if (n<=1) return;
else
{
1)- A(n-2);
2)- Pf(n);
3) A(n-1);
4) Pf(n-2);
}
}
```



O/p = 20432012

C-Programming and DS

Points to Remember :-

1) A function can't be defined inside another function
2) Every function can call itself (including main) is called recursion
3) Default return type is int not void
4) A function can take any no. of arguments fun(int x, ...)
5) function need not return a value but if it returns that it can be of any type except function or Array
6) function can't return more than one value
7) Redeclaration of function is not an error whereas redefinition of function is an error
8) function arguments can't Static or Extern

C-Programming and DS

GATE

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

P 106

106-10 ⇒ 96

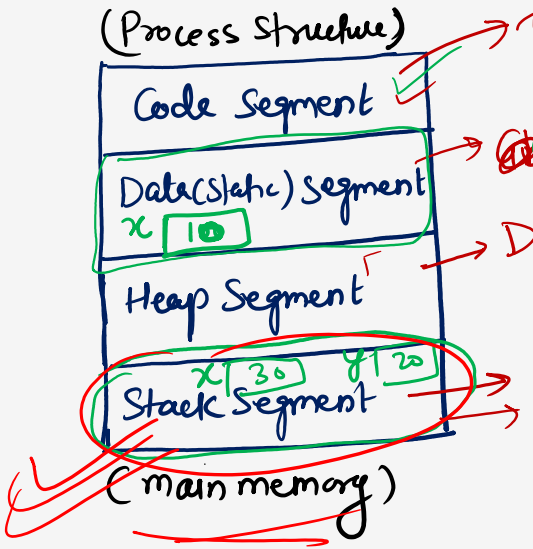
What is the return value of fun(95);

f(95)
f(f(106))
f(96)
f(f(107))
f(97)
f(f(108))
f(98)
f(f(109))
f(99)
f(f(110))
f(100)
f(f(111)) → f(101)

91 Ans

C-Programming and DS

Memory Segment :-



Text Code readonly

Global or Static

Dynamic memory Allocat
malloc
calloc

Return
local variable &
formal parameter

5 mins

Ex:

```
int x=10;
main()
{ int y=20;
  {
    int x=30;
    printf("%d",x);   // 30
  }
  printf("%d%d",x,y);  // 10, 20
}
```

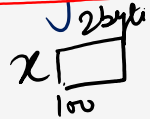
C-Programming and DS

NOTE:

Data segment

1) Local variable is having higher Precedence than global variable
2) If global variable not initialized, then default it Contain 0
3) for the global variable memory will be allocated at compile time if it is not initialized default Contains 0
4) For the local variable memory allocated at runtime (execution time) if it is not initialized default Contains garbage value
5) Local variable is having higher Precedence Compare to global variable
6) The Purpose of declaring variable is allocating memory

Ex:- int x;



C-Programming and DS

Datatype
Scope
Life
Default value

Storage classes:-

1) In the C-language every variable is having characterstics like data type, Scope, lifetime and default value

2) Storage classes mainly deals with Scope and lifetime & Default of a variable

Scope:- The region in which the variable is available and accessible is called as Scope of a variable

Extent: Life time of a variable

C-Programming and DS

Imp

Type	Scope	Extent (Lifetime)	Default value	Storage	Memory segment
Auto	Body (Local)	Body (Local)	Garbage	RAM	Stack segment
Register	Body (Local)	Body (Local)	Garbage	Register	CPU Register
Static	Body (Local)	Program (Global)	0	RAM	Data segment
Extern	Program (Global)	Program (Global)	0	RAM	Data segment

100, 90%

C-Programming and DS

Auto :- Syntax:- auto data_type variable name; (or) data_type variable name;

Ex:- auto int x; (or) int x;

1) By default all the local variable are auto variable, if auto variable is not initialized then default contain garbage value

2) for every function call auto variable is recreated and reinitialized (Imp)

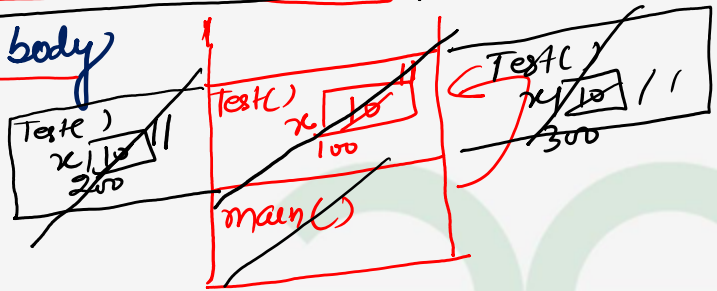
3) It has local scope and life time is within the body

Ex:-

```
main()
{ test();
  test();
  test();
}

test()
{ auto int x=10;
  printf("%d", x);
  x++;
}
```

o/p = 10, 10, 10



C-Programming and DS

90%

Register :-

Syntax :- register datatype variablename;

Ex :- register int i; /*

printf(&i); / Error

10 ∞ =>

1) Keyword register is not a command but its just a request by the Compiler to allocate the memory in CPU register

2) If the free registers are available then it will be allocated in register, otherwise they will be Allocated in the RAM only

3) Register variable are faster as compared to auto variable hence they will be used in loops and functions

4) By default register variable behave like auto variable only

```
register int i;
for(i=1; i<=10^5; i++)
    Pf("%d", i);
```

C-Programming and DS

VImp

Static :- Syntax:- Static, data type, variablename;

Ex:- Static int x;

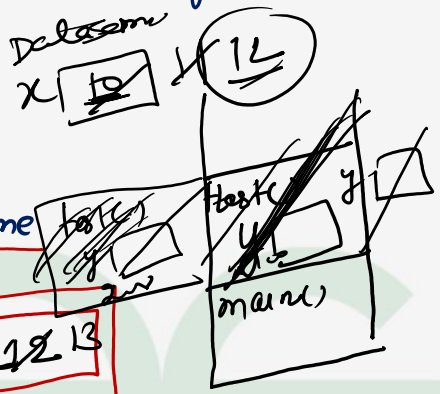
1) The variable Persisting the Previous state value even though the destruction of several functions calls is called as Static variable.

2) Static variable is created only once, memory will allocated only once in data segment at Compile time ⇒ (static)

3) if it is not initalized default it contain 0

4) it has local Scope but lifetime will end after completion of program.

5) Static variable can be local variable but memory allocated at Compile time

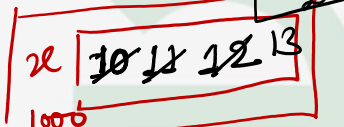


Ex:-

```
main()
{ test();
  test();
  test();
}
```

```
test()
{ int y;
  static int x=10;
  printf("%d", x);
  x++;
}
```

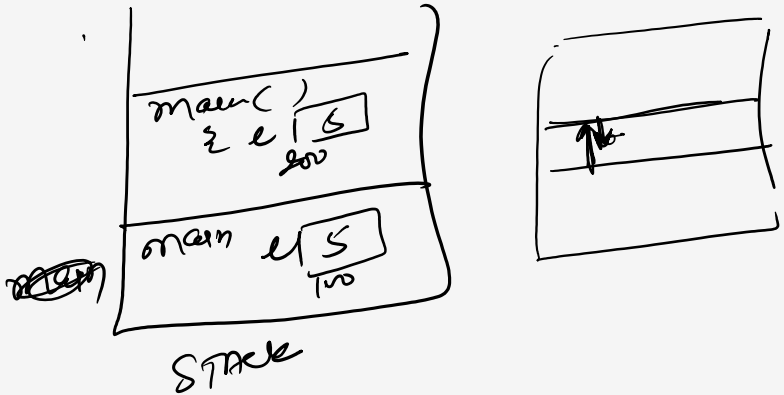
o/p = 10, 11, 12 ✓



C-Programming and DS

Q-1)

```
main()
{ int i=5;
  Pf("%d",--i); //4
  if(i)
    main();
}
```



o/p = ?

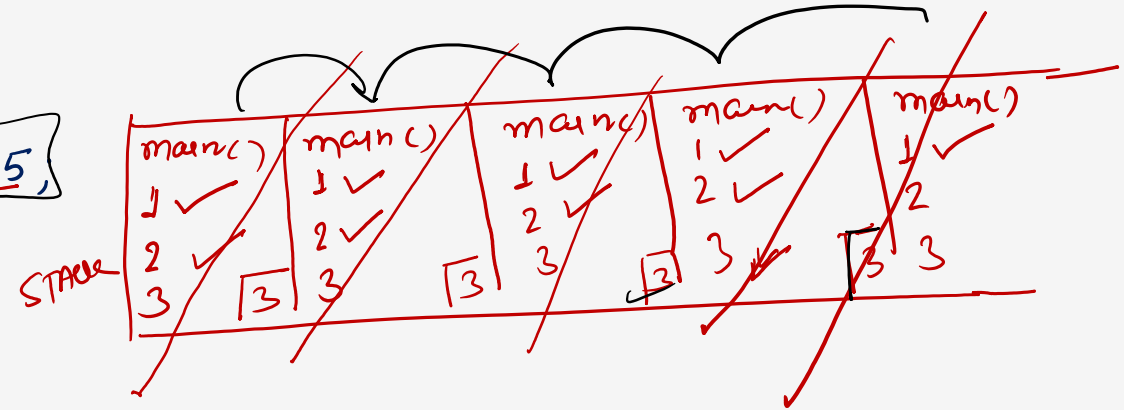
⇒ 4 4 4 4 4 4 ——— upto Stackoverflow message

But not infinite loop

C-Programming and DS

Q-2)

```
main()
{ static int i=5;
1) if(--i)
2) {main();
3) printf(i);
}
}
```

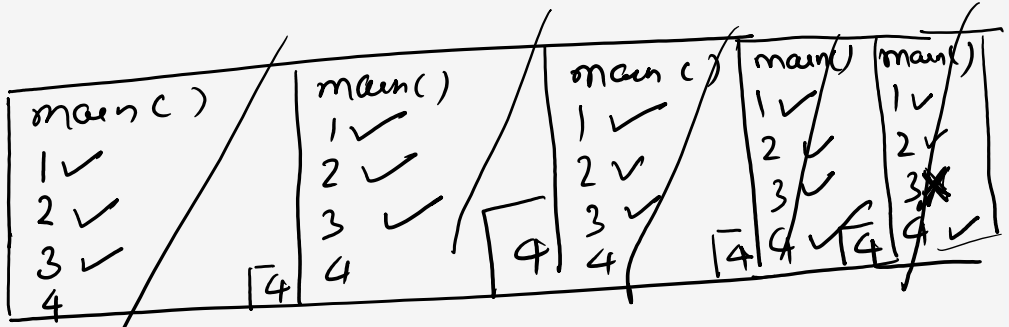


C-Programming and DS

Q3)

```
main()
{
  static int i = 5
1) if(--i),
2) if(i)
3) main()
4) Pf("Hi")
}
```

STACK



Data segment

5	4	3

i

2 X 0

if(i) if(4)

4 3 2 1 0 Hi Hi Hi Hi Hi

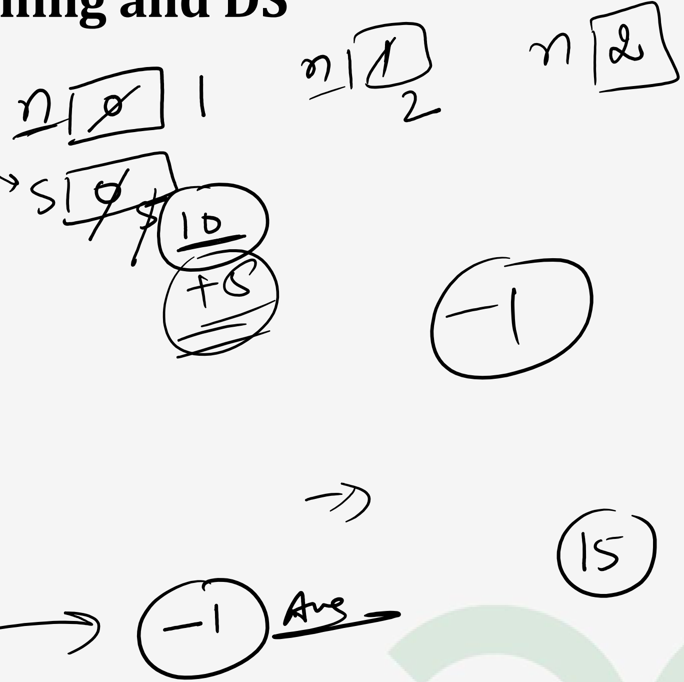
C-Programming and DS

GATE

Q-4)

```
int Sample(int n)
{
  Static int S;
  if(n==2) return S;
  S+=5;
  n++;
  return Sample(n);
}
main()
{
  pf("%d", Sample(?));
}
```

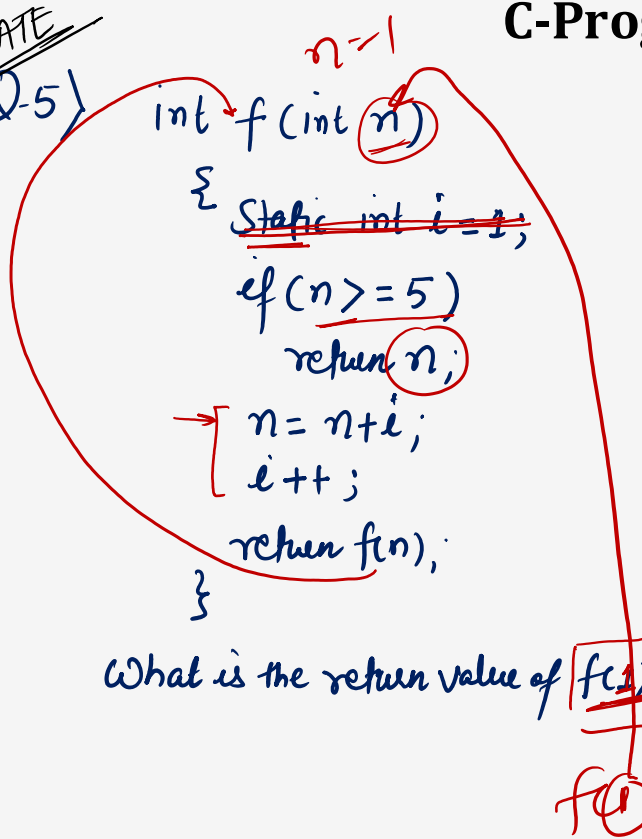
What would be here to print 15



C-Programming and DS

GATE

Q-5)



n | 1 | 2 | 4 | 7 | 7

i | 1 | 2 | 3 | 4

Data segment

What is the return value of f(1) ? 7 Ans

C-Programming and DS

GATE Q-6)

```
int f(int n)
{
  static int r;
  if(n<=0) return 1;
  if(n>3)
  { r=n;
    return f(n-2)+2;
  }
  return f(n-1)+r;
}
```

n | 5 | 4 | 3 | 2 | 1 | 0

r | 0 | 5

What is the return value of f(5) ___ ?

f(5) = 18 Ans

f(3)+2 = 18

f(2)+5 = 16

f(1)+5 = 11

f(0)+5 = 6

1

C-Programming and DS

GATE

Q-7)

```
int incr(int i)
{ static int count;
  count = count + i;
  return (count);
}
⇒ void main()
{ int i, j;
→ for(i=0; i<=4; i++)
    j = incr(i);
  printf("%d", j);
}
```

What is the output printed 10 ?

i	j	static count
0	0	0
1	1	1
2	3	3
3	6	6
4	10	10
5		

C-Programming and DS

GATE

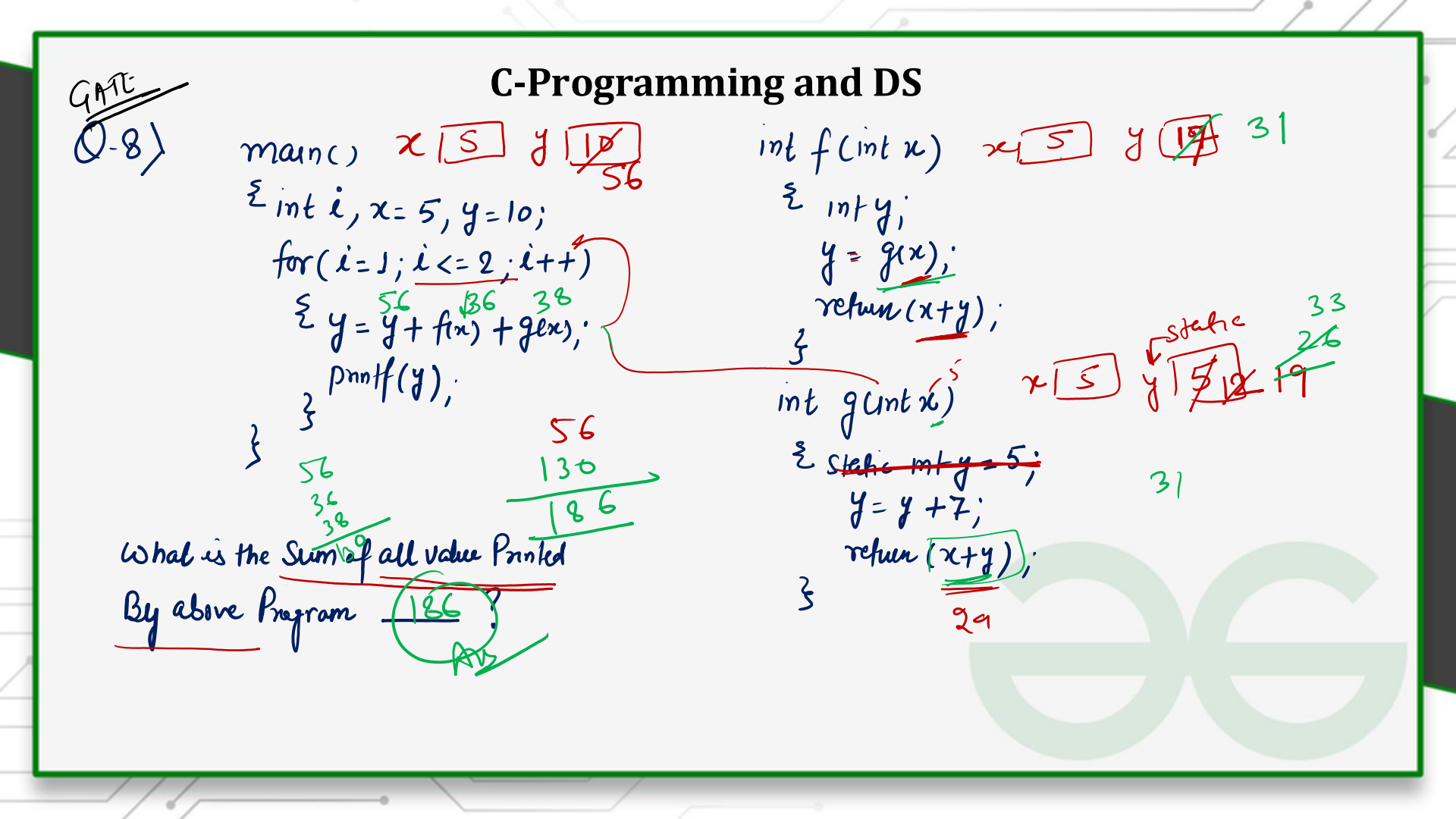
Q-8)

```
main()
{ int i, x=5, y=10;
  for(i=1; i<=2; i++)
  { y = y + f(x) + g(x);
    printf(y);
  }
}
```

```
int f(int x)
{ int y;
  y = g(x);
  return(x+y);
}
```

```
int g(int x)
{ static int y = 5;
  y = y + 7;
  return(x+y);
}
```

What is the Sum of all value Printed By above Program ? 186



C-Programming and DS

Q-9)

```
int r()
{ static int num = 7;
  return num--;
}
main()
{
  for( r(); r(); r())
      printf("%d", r());
}
```

num | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0

3

7 6 4 1

52

52

What's output printed ?

a) 41 c) 63

b) 630 d) 52

5 2

C-Programming and DS

Q-10)

```
Void Count(int n)
{
1) static int d=1;
2) printf("%d", n);
3) printf("%d", d);
4) d++;
5) if(n>1)
6) Count(n-1);
7) printf("%d", d);
}
Void main()
{
Count(3);
}
```

What o/p printed ?

a) 31 22 13 444

b) 31 21 11 222

c) 31 22 13 4

d) 31 21 11 2

31 22 13 444

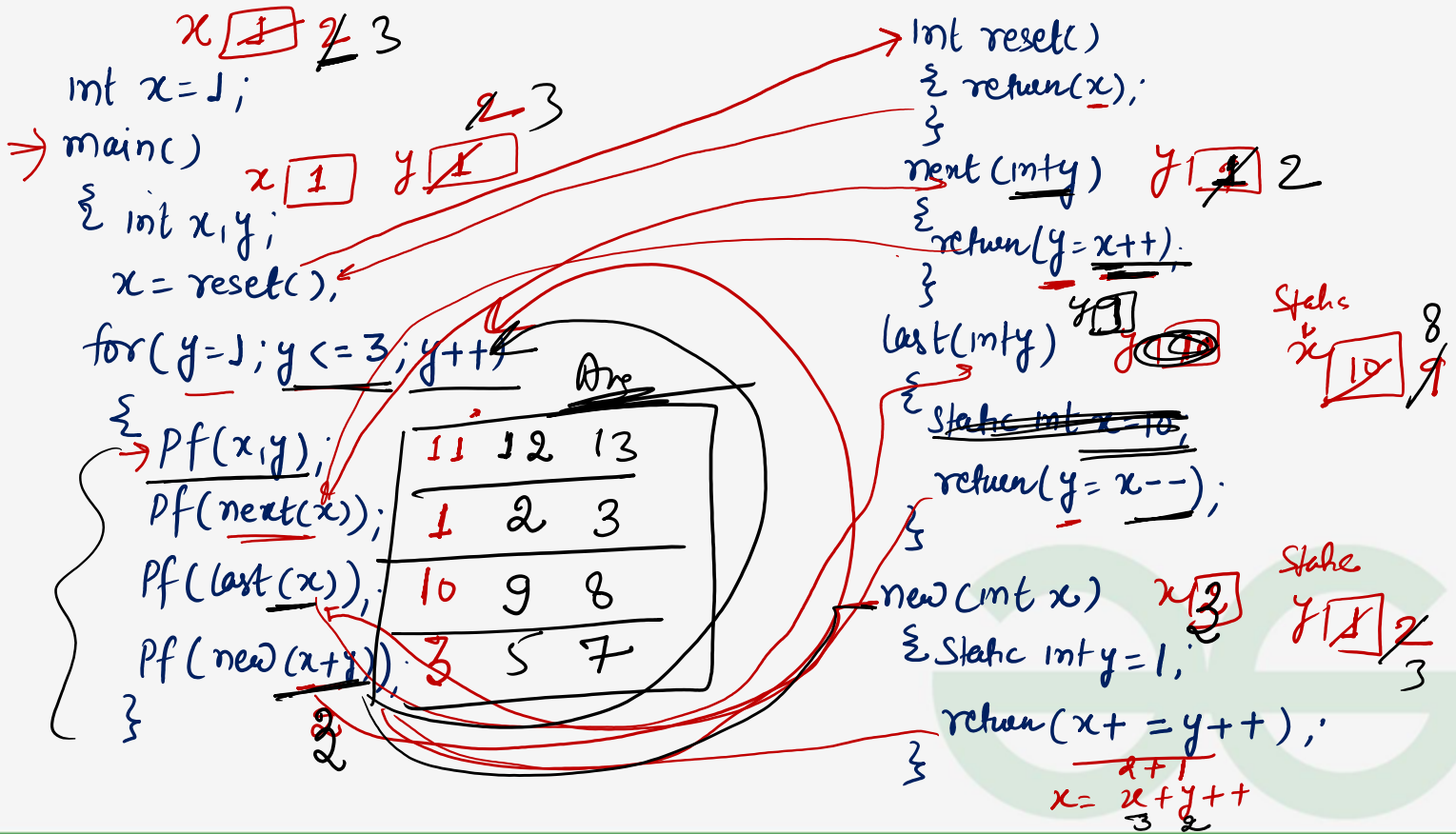
n 3 2 1

static → d 1 2 3 4

C-Programming and DS

GATE

Q-11)



C-Programming and DS

Last Ques GATE

Q.12)

```
funcf(int x)
{ int y;
  y = funcg(x);
  return y;
}

funcg(int x)
{ static int y = 10;
  y++;
  return (x+y);
}

main()
{ static int y=10, x=5, c;
  for(c=1; c<=2; c++)
  { y += funcf(x) + funcg(x);
    Pf(y);
  }
}
```

x 5, y 15

x 5

y 10 → 11 → 12 → 13 → 14

16

43

y 10 x 5, c 1

43 + 18 + 19

43 80 Ans

Thank You !