**RECURSION**

1. Consider the following Pascal function where A and B are non-zero positive integers. What is the value of GET(3,2)? **(GATE-1995)**

function GET(A,B:integer);integer;

begin

if B=0 then

GET:=1

else if A<B then

GET:=0

else

GET:=GET(A-1,B)+GET(A-1,B-1)

end;

1. What value would the following function return for the input x = 95? **(GATE-1998)**

function fun (x:integer):integer;

begin

if x > 100 then

fun : x – 10

else

fun : fun(fun (x + 11))

end;

a) 89 b) 90 c) 91 d) 92

1. Consider the following C function definition: **(GATE-1999)**  
    int Trial (int a, int b, int c)   
   {   
      if ((a > = b) && (c < b))

return b;   
    else if (a > = b)

return Trial (a,c,b);   
    else

return Trial (b,a,c);   
}  
The function Trial:    
a) Finds the maximum of a, b, and c  b) Finds the minimum of a, b and c   
c) Finds the middle number of a, b, c  d) None of the above

1. Consider the following C function: Find f(1)? **(GATE-2004)**

int f(int n)

{staticinti=1;

if(n>=5)

return n;

n=n+i;

i++;

return f(n);

}

a)5 b)6 c)7 d)8

1. Consider the following C-program: **(GATE-2005)**

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,", k);

}

int**main** ()

{

 int a = 2048, sum = 0;

  foo (a, sum);

  printf ("%d\n", sum);

  getchar();

}

What 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

1. Consider the following C function: **(GATE-2007)**  
   int f(int n)  
   {  
    static int r = 0;  
    if (n <= 0)

return 1;  
 if (n > 3)  
 {  
  r = n;  
 return f(n-2)+2;  
 }  
 return f(n-1)+r;  
}  
What is the value of f(5) ?  
a)5 b)7 c)9 d)18

**COMMON DATA QUESTIONS (GATE-2011)**  
Consider the following recursive C function that takes two arguments  
unsigned int foo(unsigned int n, unsigned int r)

{  
  if (n  > 0)

return (n%r +  foo (n/r, r ));  
  else

return 0;  
}

1. What is the return value of the function foo when it is called as foo(345, 10) ?  
   a) 345 b) 12 c) 5 d) 3
2. What is the return value of the function foo when it is called as foo(513, 2)?  
   a) 9 b) 8 c) 5 d) 2
3. What is the value printed by the following C program? **(GATE-2010)**

#include stdio.h  
intf(int \*a,int n)  
{  
if (n<= 0)

return 0;   
else if(\*a% 2==0)

return \*a+f(a+1,n-1);   
else

return \*a - f(a+1,n-1);   
}

int main( )  
{  
int a[ ]={12,7,13,4,11,6};  
printf("%d",f(a,6));  
return 0;  
}

a) -9 b)5 c)15 d)19

1. Consider the C function given below. **(GATE 2014)**

int f(int j)

{

staticinti = 50;

int k;

if (i == j)

{

printf(“something”);

k = f(i);

return 0;

}

else return 0;

}

Which one of the following is TRUE?

(A) The function returns 0 for all values of j.

(B) The function prints the string something for all values of j.

(C) The function returns 0 when j = 50.

(D) The function will exhaust the runtime stack or run into an infinite loop when j=50

1. Consider the following C function. **(GATE 2015)**

int fun (int n)

{

int x = 1, k;

if (n==1) return x;

for (k = 1; k < n; ++k )

x= x+fun( k ) \* fun (n-k) ;

return x;

}

The return value of fun (5) is \_\_\_\_\_\_\_

1. Consider the C functions foo and bar given below: **(GATE 2017)**

int foo(intval) {

int x=0;

while(val> 0) {

x = x + foo(val--);

}

returnval;

}

int bar(intval) {

int x = 0;

while(val> 0) {

x= x + bar(val-1);

}

returnval;

}

Invocations of $foo(3)$ and $bar(3)$ will result in:

A) Return of 6 and 6 respectively.

B) Infinite loop and abnormal termination respectively.

C) Abnormal termination and infinite loop respectively.

D) Both terminating abnormally.

1. Consider the following two functions **(GATE 2017)**

void fun1(int n){

if(n == 0) return;

printf(“%d”, n);

fun2(n-2);

printf(“%d”, n);

}

void fun2(int n){

if(n == 0) return;

printf(“%d”, n);

fun1(++n);

printf(“%d”, n);

}

The output printed when fun1 (5) is called is

(A) 53423122233445 (B) 53423120112233

(C) 53423122132435 (D) 53423120213243

**STORAGE CLASSES**

1. The value of j at the end of the execution of the following C program. **(GATE-2000)**

intincr (inti)  
{  
   static int count = 0;  
   count = count + i;  
   return (count);  
}

main ()  
{  
   inti,j;  
   for (i = 0; i<=4; i++)  
        j = incr(i);

}

a) 10 b) 4 c) 6 d) 7

**COMMON DATA QUESTIONS**

Consider the following C code segment

voidprtFun (void)

{

staticint a = 2; /\* line 2 \*/

int b = 1;

    a += ++b;

    printf (" \n %d %d " , a, b);

}

int a, b, c = 0;

voidprtFun (void);

int main ()

{

 staticint a = 1; /\* line 1 \*/

    prtFun();

    a += 1;

    prtFun();

printf ( "\n %d %d " , a, b) ;

}

1. What output will be generated by the given code segment? **(GATE-2013)**



1. What output will be generated by the given code d\segment if: **(GATE-2013)**  
   Line 1 is replaced by “auto int a = 1;”  
   Line 2 is replaced by “register int a = 2;”  
   
2. What will be the output of the following C program? **(GATE 2016)**

void count(int n)

{

staticint d = 1;

printf("%d ", n);

printf("%d ", d);

d++;

if(n > 1) count(n-1);

printf("%d ", d);

}

int main()

{

count(3);

}

A)3 1 2 2 1 3 4 4 4 B)3 1 2 1 1 1 2 2 2

C)3 1 2 2 1 3 4 D)3 1 2 1 1 1 2

1. The output of executing the following C program is \_\_\_\_\_\_\_\_. **(GATE 2017)**

# include <stdio.h>

int total(int v)

{

staticint count = 0;

while (v) {

count += v & 1;

v >>= 1;

}

return count;

}

void main()

{

staticint x = 0;

inti = 5;

for (; i> 0; i--) {

x = x + total(i);

}

printf (“%d\n”, x) ;

}

A) 23 B) 24 C) 26 D) 27

1. Consider the following recursive C function. **(GATE 2015)**

void get (int n)

{

if (n<1) return;

get (n-1);

get(n- 3) ;

print f (" %d",n) ;

}

If get (6) function is being called in main () then how many times will the get () function be invoked before returning to the main ( ) ?

a) 15 b) 25 c) 35 d) 45

**STATIC & DYNAMIC SCOPE**

1. Study the following program written in a block structure language: **(GATE-1987)**

varx,y:integer;

procedure P(n:integer);

begin

x:=(n+2)/(n-3)

end;

procedure Q

varx,y:integer;

begin

x:=3;

y:=4;

P(y);

Write(x); ------------(1)

end;

begin

x:=7;

y:=8;

Q;

Write(x);-------------(2)

end;

What will be printed by the write statements marked (1),(2) in the program if variable are statically scoped?

a)3,6 b)6,7 c)3,7 d)None of these

1. For the program given above what will be printed by the write statements marked (1),(2) if the variables are dynamically scoped: **(GATE-1987)**

a)3,6 b)6,7 c)3,7 d)None of these

1. Indicate results of the following program if the language uses

i)Static scope rule and

ii) Dynamic scope rules **(GATE-1989)**

varx,y:integer;

procedure**A**(var z:integer);

var x:integer;

begin

x:=1;

B;

z:=x;

end;

procedure**B**;

begin

x:=x+1;

end;

begin

x:=5;

A(y);

write(y)

end;

1. For the following code indicate the output, if **(GATE-1991)**

I) Static scope rule

II)Dynamic scope rule

vara,b:integer;

procedure P;

a:=5;

b:=10

end {P};

procedure Q;

vara,b:integer;

P;

end {Q};

begin

a:=1;

b:=2;

Q;

write(‘a=’,a,’b=’,b)

end;

1. Consider the program below: **(GATE-1994)**

Program main;

var r:integer;

procedure two;

begin

write(r) ;

end;

procedure one;

var r: integer;

begin

r:=5;

two;

end;

begin

r:=2;

two;

one;

two;

end;

What is printed by the above program if

1. Static scoping is assumed for all variables
2. Dynamic scoping is assumed for all variables
3. Given the following Pascal-like program segment **(GATE-1997)**  
   Procedure A,  
   x, y : integer ;  
   Procedure B;  
   x, z : real ;  
   S1  
   end B;  
   Procedure C;  
   i : integer ;  
   S2  
   end C  
   end A;  
   The variables accessible in S1 and S2 are

a) x of A, y, x of B and z in S1 and x of B, y and i in S2

b) x of B, y and z in S1 and x of B, i and z in S2

c) x of B, z and y in S1 and x of A, i and y in S2

d)None of these

1. Consider the following program in a language that has dynamic scooping  
   var x: real; **(GATE-1999)**  
   procedure show:   
   begin

print(x);

end;   
procedure small;   
var x: real;   
begin

x: = 0.125;

show;

end;   
begin

x:=0.25;   
show;

small ;  
end.;  
Then the output of the program is:   
a) 0.125 0.125 b) 0.25 0.25 c) 0.25 0.125 d) 0.125 0.25

1. Consider the following C program. **(GATE 2015)**

#include <stdio.h>

int f1(void) ;

int f2(void) ;

int f3(void);

int x = 10;

int main()

{

int x = 1;

x += f1() + f2() + f3() + f2();

printf("%d", x) ;

return 0;

}

int f1()

{

int x = 25;

x++ ;

return x;

}

int f2()

{

staticint x = 50;

x++ ;

return x;

}

int f3()

{

x\*=10;

return x ;

}

The output of the program is\_\_\_\_\_\_\_\_

**PARAMETER PASSING**

1. Consider the pseudo code (all items are of integer type) **(GATE-1991)**

Procedure P(a,b,c);

a:=2;

c:=a+b;

end P;

begin

x:=1

y:=5;

z:=100;

P(x,x\*y,z);

write(‘x=’,x,’z=’,z);

end;

Determine its output if the parameters are passed to the procedure P by

(i) value (ii) reference (iii)name

1. Consider the following program in pseudo pascal syntax. What is printed by the program if parameter a in procedure test 1 is passed as **(GATE-1996)**

I)call by reference parameter

ii)call by value result parameter

program Example(input,output)

var b:integer;

procedure test 2;

begin

b:=10

end;

procedure test 1(a:integer);

begin

a:=5;

writeln(‘point 1:’,a,b);

test 2;

writeln(‘point2 :’,a,b);

end;

begin

b:=3; test1(b);

writeln(‘point3:’,b);

end;

1. What will be the output of the following program assuming that parameter passing is

(i)call by value (ii)call by reference (iii)call by copy restore

Procedure P{x,y,z}; **(GATE-1999)**

begin

y:=y+1;

z:=z+x;

end;

begin

a:=2;

b:=3;

P(a+b,a,a);

print (a)

end;

1. Consider the following program is pseudo pascal syntax **(GATE-2000)**

Program main;

var x:integer;

Procedure Q(z:integer);

begin

z:=z+x;

writeln(z);

end;

Procedure P(y:integer)

var x:integer;

begin

x:=y+2;

Q(x);

Writeln(x);

end;

begin

x:=5;

P(x);

Q(x);

Writeln(x);

end;

What is the output of the program, when

a)Parameter passing is call by value and the scope rule is static scope

b)parameter passing is call by reference and the scope rule is dynamic scope

1. What is printed by print statement in the program P1 assuming call by reference parameter passing. **(GATE-2001)**

Program P1()

{ x=10;

y=3;

func1(y,x,x);

print x;

print y;

}

func1(x,y,z)

{ y=y+4;

z=x+y+z;

}

a)10,3 b)31,3 c)27,7 d)None of the above

1. Consider the following program **(GATE-2001)**

Program P2

var n:int;

Procedure W(var x:int)

begin

x=x+1;

Print x;

end;

procedure D

begin

var n:int;

n=3;

W(n);

end;

begin

n=10;

D;

end;

If the language has dynamic scoping and parameters are passed by reference, what will be printed by the program?

a)10 b)11 c)3 d)None of these

1. Consider the following program **(GATE-2003)**

globalinti = 100, j = 5;  
void P(x)  
{  
   inti = 100;

 print(x + 10);  
   i = 200;  
   j = 20;  
   print(x);  
}  
  
main()  
{  
   P(i + j);  
}

If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are  
a. 115, 220 b. 25, 220 c. 25, 15 d. 115, 105

1. Consider the following program in Psuedopascal syntax; **(GATE-1997)**

Program what:

var z:integer

Procedure recur(x);

begin

if x<=40 then

begin

x:=x+z;

recur(x);

z:=x+10;

end;

end;

begin

z=10;

recur(z);

writeln(z);

end;

a)Suppose the parameter to the procedure ‘recur’ is passed by value.

i)what value is printed by the program?

ii)How many times is ‘recur’ called?

b)What value is printed if the parameter is passed by reference?

1. What does the following code do? **(GATE-1993)**

vara,b:integer;

begin

a:=a+b;

b:=a-b;

a:=a-b;

end;

a)exchanges a and b

b)doubles a and stores in b

c)doubles b and stores in a

c)leaves a and b unchanged

e)none of the above

1. What is the result of the following program? **(GATE-1996)**

program side-effect(intput,output);

varx,result:integer;

function f(var x:integer):integer;

begin

x:=x+1;

f:=x;

end;

begin

x:=5;

result:=f(x)\*f(x);

writeln(result);

end;

a)5 b)25 c)36 d)42

1. What is the return value of f(p,p), if the value of p is initialized to 5 before the call? Note that the first parameter is passed by reference, whereas the second parameter is passed by value. **(GATE-2013)**

int f (int&x, int c)

{

c = c - 1;

if (c==0)

return 1;

x = x + 1;

return f(x,c) \* x;

}

a) 3024 b) 6561 c) 55440 d) 161051

1. What is the value of X printed by the following program ? **(GATE-1995)**  
   program COMPUTE ( input, output );  
   var X : integer ;  
   procedure FIND ( X: real ) ;  
   begin  
   X : = sqrt ( X ) ;  
   end ;

begin  
X : = 2  
FIND ( X )  
writeln ( X )

end;

a)2 b)sqare root 2 3)run time error d)None of these

**COMMON DATA QUESTIONS (GATE-1993)**

Consider the block of code given below

Program PARAM(input,output);

varm,n:integer;

procedure P(varx,y:integer);

var m:integer;

begin

m:=1;

x:=y+1;

end;

procedure Q(x:integer; var y:integer);

begin

x:=y+1;

end;

begin

m:=0; P(m,m);write(m);

n=0; Q(n\*1,n);write(n)

end;

1. The value of m, output by the program PARAM is

a)1, because m is a local variable in P

b)0, because m is the actual parameter that corresponds to the formal parameter in P

c) 0, because both x and y are just reference to m and y has the value 0

d)1, because both x and y are just references to m which gets modified in procedure p

1. The value of n, output by the program PARAM is

a)0, because n is the actual parameter corresponding to x in procedure Q

b)0, because n is the actual parameter to y in procedure Q

c)1, because n is the actual parameter corresponding to x in procedure Q

d)1, because n is the actual parameter corresponding to y in procedure Q

e)None of the above

1. What is the scope of m declared in the main program?

a)PARAM,P,Q b)PARAM,P c)PARAM,Q d)P,Q

1. What will be the output of the following pseudo-code when parameters are passed by reference and dynamic scoping is assumed? **(GATE 2016)**

a=3;

void n(x)

{

x = x \* a;

print(x);

}

void m(y)

{

a = 1;

a = y - a;

n(a);

print(a);

}

void main()

{

m(a);

}

A)6, 2 B)6, 6 C)4, 2 D)4, 4

**GENERAL**

1. Match the pairs in the following **(GATE-1990)**

A)Pointer data type (1)Type conversion

B)Activation Record (2)Dynamic data structure

C)Repeat-until (3)Recursion

D)Coercion (4)Nondeterministic loop

1. In some programming languages, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denote the sets of letters and digits respectively, which of the following expressions defines an identifier? **(GATE-1995)**

a) (L U D)+ b) L ( L U D )+ c) (LD)+ d) L. (L.D)+

1. The number of tokens in the following C statement. **(GATE-2000)**  
           printf("i=%d,&i=%x",i,&i); is  
   a) 3 b) 26 c) 10 d) 21
2. Consider the following C function **(GATE-2003)**

float f(float x,int y)

{

floatp,s; inti;

for(s=1,p=1,i=1;i<y;i++)

{

p \*=x/i;

S +=p;

}

return s;

}

For large values of y, the return value of the function f best approximates  
a) xy b) ex c) ln(1 + x) d) xx

1. Consider the following C program **(GATE-2004)**

main()

{

   int x, y, m, n;

   scanf ("%d %d", &x, &y);

   /\* x > 0 and y > 0 \*/

   m = x; n = y;

   while (m != n)

   {

      if(m>n)

         m = m - n;

      else

         n = n - m;

   }

   printf("%d", n);

}

The program computes   
a) x + y using repeated subtraction b) x mod y using repeated subtraction

c) the greatest common divisor of x and yd) the least common multiple of x and y

1. Consider the following program fragment for reversing the digits in a given integer to obtain a new integer. Let n = D1D2…Dm **(GATE-2004)**

|  |
| --- |
| int n, rev;  rev = 0;  while (n > 0)  {     rev = rev\*10 + n%10;     n = n/10;  } |

The loop invariant condition at the end of the ith iteration is

a) n = D1D2….Dm-i and rev = DmDm-1…Dm-i+1  
b) n = Dm-i+1…Dm-1Dm and rev = Dm-1….D2D1  
c) n \not =rev  
d) n = D1D2….Dm and rev = DmDm-1…D2D1

1. Which combination of the integer variables x, y and z makes the variable a get the value 4 in the following expression? **(GATE-2008)**

|  |
| --- |
| **a = ( x> y ) ? (( x > z ) ? x : z) : (( y > z ) ? y : z )** |

a) x = 3, y = 4, z = 2 b) x = 6, y = 5, z = 3  
c) x = 6, y = 3, z = 5 d) x = 5, y = 4, z = 5

1. The following program is to be tested for statement coverage: **(GATE-2010)**

begin

if (a== b)

{

S1;

exit;

}

else if (c== d)

{

S2;

}

else

{

S3;

exit;

}

S4;

end

The test cases T1, T2, T3 and T4 given below are expressed in terms of the properties satisfied by the values of variables a, b, c and d. The exact values are not given.

T1 : a, b, c and d are all equal  
T2 : a, b, c and d are all distinct  
T3 : a = b and c != d  
T4 : a != b and c = d  
Which of the test suites given below ensures coverage of statements S1, S2, S3 & S4?  
a) T1, T2, T3 b) T2, T4 c) T3, T4 d) T1, T2, T4

1. What will be the output of the following C program segment? **(GATE-2012)**

charinchar = 'A';

switch (inchar)

{

case 'A' :    printf ("choice A \n") ;

case 'B' :   printf ("choice B ") ;

case 'C' :

case 'D' :

case 'E' :

default:    printf ("No Choice") ;

}

a)No choice

b)Choice A

c)Choice A

Choice B No Choice

d)Program gives no output as it is erroneous

|  |
| --- |
|  |
|  |
|  |

1. Consider the following pseudo code. What is the total number of multiplications to be performed? **(GATE 2014)**

D = 2

fori = 1 to n do

for j = i to n do

for k = j + 1 to n do

D = D \*3

1. Half of the product of 3 consecutive integers
2. One third of the product of 3 consecutive integers
3. One-sixth of the product of 3 consecutive integers
4. None of the above
5. Consider the function func shown below: **(GATE 2014)**

intfunc(intnum) {

int count = 0;

while (num) {

count++;

num>>= 1;

}

return (count);

}

The value returned by func(435) is ………………

1. Suppose n and p are unsigned int variables in a C program. We wish to set p to nC3.

If n is large, which one of the following statements is most likely to set p correctly? **(GATE 2014)**

A) p = n \* (n-1) \* (n-2) / 6;

B) p = n \* (n-1) / 2 \* (n-2) / 3;

C) p = n \* (n-1) / 3 \* (n-2) / 2;

D) p = n \* (n-1) \* (n-2) / 6.0;

1. Consider the following function **(GATE 2014)**

double f (double x)

{

if ( abs( x\*x – 3) < 0.01) return x;

else return f (x/2+1.5/x);

}

Give a value q(to 2 decimal) such that f(q) will return q: ………………..

1. Consider the following C program **(GATE 2015)**

#inclue<stdio.h>

int main()

{

inti, j, k=0;

j = 2\*3 / 4 + 2.0 / 5 + 8 / 5;

k - = --j;

for (i=0; i<5; i++)

{

switch(i+k)

{

case1:

case 2 : printf("\ n %d", i+k) ;

case 3: printf("\ n %d", i+k);

default :printf("\ n%d", i+k) ;

}

}

return 0;

}

The number of times printf statement is executed is \_\_\_\_\_\_\_\_\_

1. The attributes of three arithmetic operators in some programming language are given below. **(GATE 2016)**

Operator Precedence Associativity Arity

+ High Left Binary

- Medium Right Binary

\* Low Left Binary

The value of the expression 2 - 5 + 1 - 7 \* 3 in this language is \_\_\_\_

1. Consider the following C program. **(GATE 2017)**

#include<stdio.h>

#include<string.h>

voidprintlength(char \*s, char \*t)

{

unsignedint c=0;

intlen = ((strlen(s) - strlen(t)) > c) ? strlen(s) :strlen(t);

printf("%d\n", len);

}

void main()

{

char \*x = "abc";

char \*y = "defgh";

printlength(x,y);

}

Recall that strlen is defined in string.h as returning a value of type size\_t, which is an unsigned int. The output of the program is \_\_\_\_\_\_\_\_\_\_

1. Consider the C program fragment below which is meant to divide x by y using repeated subtractions. The variable x, y, q and r are all unsigned int.

while(r >= y) **(GATE 2017)**

{

r = r - y;

q = q + 1;

}

Which of the following conditions on the variables x, y, q and r before the execution of the fragment will ensure that the loop terminates in a state satisfying the condition x == (y\*q + r)?

(A) ( q == r ) && ( r == 0)

(B) ( x> 0 ) && ( r == x ) && ( y > 0 )

(C) ( q == 0 ) && ( r == x ) && ( y > 0 )

(D) ( q == 0 ) && ( y > 0 )

1. Consider the following C program: **(GATE 2017)**

#include <stdio.h>

int main()

{

int m = 10;

int n, n1;

n = ++m;

n1 = m++;

n--;

--n1;

n -= n1;

printf("%d",n);

return 0;

}

The output of the program is \_\_\_\_\_

1. The following function computes XY for positive integers X and Y. **(GATE 2016)**

intexp(int X, int Y)

{

int res = 1, a = X, b = Y;

while ( b != 0 )

{

if ( b%2 == 0)

{

a = a\*a;

b = b/2;

}

else

{

res = res\*a;

b = b-1;

}

}

return res;

}

Which one of the following conditions is TRUE before every iteration of the loop

A) XY = ab B) (res\*a)Y = (res\*X)b C) XY = res\*ab D) XY = (res\*a)b

**Arrays, Pointers, Structures**

1. Consider the following declaration of a two-dimensional array in C **(GATE: 2002)**

char a[100][100];

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0, the address of a [40][50] is

a)4040 b)4050 c)5040 d)5050

1. struct node

{

inti;

float j;

};

struct node \*s[10];

The above C declaration define 's' to be **(GATE 2000)**

1. An array, each element of which is a pointer to a structure of type node
2. A structure of 2 fields, each field being a pointer to an array of 10 elements
3. A structure of 3 fields: an integer, a float, and an array of 10 elements
4. An array, each element of which is a structure of type node.
5. The most appropriate matching for the following pairs **(GATE 2000)**

X) m=malloc(5);m=NULL; 1) using dangling pointers

Y) free(n);n->value=5; 2) using uninitialized pointers

Z) char \*p;=’a’; 3) lost memory

a)X-1 Y-3 Z-2 b)X-2 Y-1 Z-3 c)X-3 Y-2 Z-1 d)X-3 Y-1 Z-2

1. Aliasing in the context of programming language refers to **(GATE 2000)**

a)Multiple variables having in the same memory location

b)Multiple variables having the same value

c)Multiple variables having the same identifier

d)Multiple uses of the same variable

1. Consider the following C declaration **(GATE 2000)**

struct

{

short s[5];

union

{

float y;

long z;

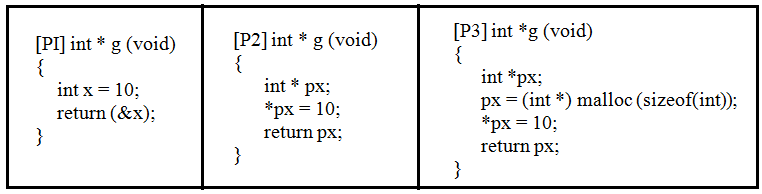
}u;

}t;

Assume that objects of the type short, float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively. The memory requirement for variable t, ignoring alignment considerations, is

a)22 bytes b)14 bytes c)18 bytes d)10 bytes

1. Consider the following three C functions: **(GATE 2001)**



|  |
| --- |
|  |

Which of the above three functions is likely to cause problems with pointers?   
a) Only P3 b) Only P1 and P3 c) Only P1 and P2 d) P1, P2 and P3

1. #include<stdio.h>

voidabc(char \*s)

{

    if(s[0] == '\0')

        return;

    abc(s + 1);

    abc(s + 1);

    printf("%c", s[0]);

}

main()

{

abc(“123”);

}

1. What will be the output of the program? **(GATE 2001)**
2. If abc(s) is called with a null terminating string s of length n characters (not counting the null character, how many characters will be printed by abc(s)?
3. Assume the following C variable declaration **(GATE 2003)**

int \*A [10], B[10][10];

Of the following expressions

I A[2] II A[2][3] III B[1] IV B[2][3]

Which will not give compile-time errors if used as left hand sides of assignment statements in a C program

a) I, II, and IV only b) II, III, and IV only

c) II and IV only d) IV only

1. Consider the following C program shown below: **(GATE 2003)**

#include<stdio.h>

#define print(x) printf(“%d”, x)

int x;

void Q(int z)

{

z+=x; print(z);

}

void P(int \*y)

{

int x=\*y+2;

Q(x); \*y=x-1;

print(x);

}

main(void)

{

x=5;

P(&x);

print(x);

}

The output of the program is

a)12 7 6 b)22 12 11 c)14 6 6 d)7 6 6

1. Consider the following C function **(GATE 2004)**

|  |
| --- |
| void swap (int a, int b)  {     int temp;     temp = a;     a = b;     b = temp;  } |

In order to exchange the values of two variables x and y.

1. call swap (x, y)
2. call swap (&x, &y)
3. swap (x,y) cannot be used as it does not return any value
4. swap (x,y) cannot be used as the parameters are passed by value
5. Consider the following C program segment **(GATE 2004)**

|  |
| --- |
| char p[20];  char \*s = "string";  int length = strlen(s);  inti;  for (i = 0; i< length; i++)       p[i] = s[length — i];  printf("%s",p); |

The output of the program is   
a) gnirts b) gnirt c) string d) no output is printed

1. What does the following C-statement declare? **(GATE 2005)**

**int (\*f) (int \*);**

a)A function that takes an integer pointer as argument and returns an integer

b)A function that takes an integer as argument and returns an integer pointer

c)A pointer to a function that takes an integer pointer as argument and returns as integer

d)A function that takes an integer pointer as argument and returns a function pointer

1. What does the following C program print? **(GATE 2011)**

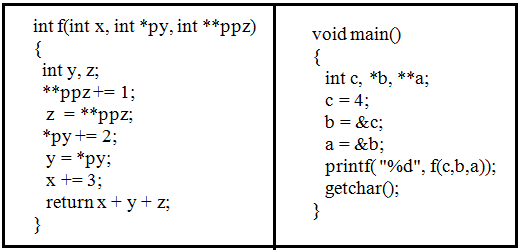
char c[] = "GATE2011";

char \*p =c;

printf("%s", p + p[3] - p[1]) ;

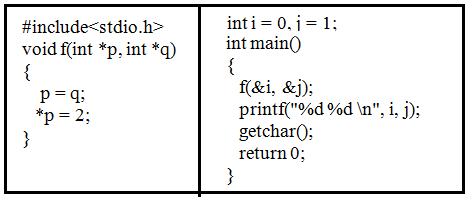
a) GATE2011 b) E2011 c) 2011 d) 011

1. What is printed by printed by following C program **(GATE 2008)**



a) 18 b) 19 c) 21 d) 22

1. What does the following program print? **(GATE 2010)**



a) 2 2 b) 2 1 c) 0 1 d) 0 2

1. Let A be a two-dimensional array declared as follows:  **(GATE 1998)** **A: array [1 …. 10] [1 …… 15] of integer;**

Assuming that each integer takes one memory location, the array is stored in row-major order and the first element of the array is stored at location 100, what is the address of the element A[i][j]?

a) 15i + j + 84           b) 15j + i + 84   c) 10i + j + 89           d) 10j + i + 89

1. An n x n array v is defined as follows **(GATE 2000)**

v**[i,j]=i-j for all i,j, 1<=i<=n, 1<=j<=n**

The sum of the elements of the array v is

a)0 b)n-1 c)n2-3n+2 d)n2 (n+1)/2

1. Suppose you are given an array s[1...n] and a procedure reverse (s,i,j) which reverse the order of elements in s between positions i and j (both inclusive). What does the following sequence do, where 1 < k <= n: **(GATE 2000)**

reverse (s, 1, k);  
 reverse (s, k + 1, n);  
 reverse (s, 1, n);  
a) Rotates s left by k positions b) Leaves s unchanged  
c) Reverses all elements of s d) None of the above

1. A single array A[1..MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (topl< top 2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for “stack full” is **(GATE 2004)**

a) (top1 = MAXSIZE/2) and (top2 = MAXSIZE/2+1)

b) top1 + top2 = MAXSIZE

c) (top1= MAXSIZE/2) or (top2 = MAXSIZE)

d) top1= top2 -1

1. A program P reads in 500 integers in the range [0,100] representing the scores of 500 students. It then prints the frequency of each score above 50. what would be the best way for P to store the frequencies? **(GATE 2005)**

a) An array of 50 numbers 

b) An array of 100 numbers

c) An array of 500 numbers

d) A dynamically allocated array of 550 numbers

1. Consider the following C program that attempts to locate an element x in an array Y[] using binary search. The program is erroneous. **(GATE 2008)**

f(int Y[10], int x)

{

     inti, j, k;

     i = 0; j = 9;

     do {

             k =  (i + j) /2;

             if( Y[k] < x)  i = k; else j = k;

         } while(Y[k] != x &&i< j);

     if(Y[k] == x) printf ("x is in the array ") ;

     elseprintf (" x is not in the array ") ;

}

On which of the following contents of Y and x does the program fail?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | a)Y is [1 2 3 4 5 6 7 8 9 10] and x < 10 | | |
|  | | b)Y is [1 3 5 7 9 11 13 15 17 19] and x < 1 | | |
|  | | c)Y is [2 2 2 2 2 2 2 2 2 2] and x > 2 | | |
|  | | d)Y is [2 4 6 8 10 12 14 16 18 20] and 2 < x < 20 and x is even | | |
|  | |  | | |
|  | |  | | |
|  | |  |
|  | |  |

1. Consider the following program in C language **(GATE 2014)**

#include<stdio.h>

void main()

{

inti;

int \*pi = &i;

scanf(“%d”,pi);

printf(“%d\n”,i+5);

}

Which of the following is true?

1. Compilation fails
2. Execution results in a run time error
3. On execution, the value printed is 5 more than the address of variable i
4. On execution, the value printed is 5 more than the integer value entered.
5. Consider the following C function in which size is the number of elements in the array E: **(GATE 2014)**

intMyX(int \*E, unsigned int size)

{

int Y = 0;

int Z;

inti, j, k;

for(i = 0; i< size; i++)

Y = Y + E[i];

for(i = 0; i< size; i++)

for(j = i; j < size; j++)

{

Z = 0;

for(k = i; k <= j; k++)

Z = Z + E[k];

if (Z > Y)

Y = Z;

}

return Y;

}

The value returned by the function MyXis the

1. Maximum possible sum of element in sub-array of array E
2. Maximum element in any sub-array of array E
3. Sum of maximum elements in all possible sub-arrays of array E
4. Sum of all the elements in the array E
5. Let A be a sqare matrix of size n X n. Consider the following pseudo code. What is the expected output? **(GATE 2014)**

C = 100 ;

fori =1 to n do

for j = 1 to n do

{

Temp = A[i][j] + C;

A[i][j] = A[j][i];

A[j][i] = Temp – C;

}

fori = 1 to n do

for j = 1 to n do

Output (A[i][j]);

1. The matrix A itself
2. Transpose of matrix A
3. Adding 100 to the upper diagonal elements and subtracting 100 from lower diagonal elements of A
4. None of the above
5. Consider the C function given below. Assume that the array ListA contains n (>0) elements, sorted in ascending order. **(GATE 2014)**

intprocessArray(int \*listA, int x, int n)

{

inti, j, k;

i = 0; j = n-1;

do {

k = ( i + j ) / 2;

if ( x <=listA[k])

j = k-1;

if ( listA[k] <= x)

i = k+1;

} while (i<=j);

if ( listA[k] == x)

return k;

else

return -1;

}

Which of the following statements about the function processArray is CORRECT?

1. It will run into an infinite loop when x is not in listA
2. It is an implementation of binary search
3. It will always find the maximum element in listA
4. It will return -1 even when x is present in listA.
5. The output of the following C program is \_\_\_\_\_\_\_\_\_\_. **(GATE 2015)**

void f1 (int a, int b)

{ int c;

c=a;

a=b;

b=c;

}

void f2 (int \*a, int \*b)

{

int c;

c=\*a; \*a=\*b;\*b=c;

}

main( )

{

int a=4, b=5, c=6;

f1 (a, b);

f2 (&b, &c);

printf (“%d”, c-a-b);

}

1. Consider the following function written the C programming language. **(GATE 2015)**

void foo (char \*a)

{

if (\*a && \*a != ' ')

{ foo(a+1);

putchar \*a ;

}

}

The output of the above function on input “ABCD EFGH” is

a) ABCD EFGH b) ABCD c) HGFE DCBA d) DCBA

1. What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory. **(GATE 2016)**

int main()

{

unsignedint x[4][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};

printf("%u, %u, %u", x+3, \*(x+3), \*(x+2)+3);

}

(A) 2036, 2036, 2036 (B) 2012, 4, 2204

(C) 2036, 10, 10 (D) 2012, 4, 6

1. Consider the following C program segment. **(GATE 2015)**

#include<stdio.h>

int main()

{

char s1[7]=”1234”, \*p;

p = s1+2;

\*p = ‘0’;

printf(“%s”s1);

}

What will be printed by the program?

a) 12 b) 120400 c) 1204 d) 1034

1. Consider the following C program **(GATE 2015)**

#include <stdio.h>

int main ()

{

staticint a[] = { 10, 20, 30 40, 50} ;

staticint \*p[] ={ a, a+3, a+4, a+1, a+2} ;

int \*\*ptr = p;

ptr++ ;

printf ("%d%d", ptr - p, \*\*ptr) ;

} The output of the program is \_\_\_\_\_\_\_\_\_\_

1. Consider the following C program. **(GATE 2016)**

void f(int, short);

void main()

{

inti = 100;

short s = 12;

short \*p = &s;

\_\_\_\_\_\_\_\_\_\_ ; // call to f()

}

Which one of the following expressions, when placed in the blank above, will NOT result in a type checking error?

A) f(s, \*s) B) i = f(i,s) C) f(i,\*s) D) f(i,\*p)

1. Consider the following C program. **(GATE 2016)**

#include<stdio.h>

void mystery(int \*ptra, int \*ptrb)

{

int \*temp;

temp = ptrb;

ptrb = ptra;

ptra = temp;

}

int main()

{

int a=2016, b=0, c=4, d=42;

mystery(&a, &b);

if (a < c)

mystery(&c, &a);

mystery(&a, &d);

printf("%d\n", a);

}

The output of the program \_\_\_\_\_\_\_\_\_\_\_\_

1. The following function computes the maximum value contained in an integer array p[] of size n (n >= 1) **(GATE 2016)**

int max(int \*p, int n)

{

int a=0, b=n-1;

while (\_\_\_\_\_\_\_\_\_\_)

{

if (p[a] <= p[b])

{

a = a+1;

}

else

{

b = b-1;

}

}

return p[a];

}

The missing loop condition is

A)a != n B)b != 0 C)b > (a + 1) D)b != a

1. The value printed by the following program is **(GATE 2016)**

void f(int\* p, int m)

{

m = m + 5;

\*p = \*p + m;

return;

}

void main()

{

inti=5, j=10;

f(&i, j);

printf("%d", i+j);

}

1. Consider the following program: **(GATE 2016)**

int f(int \*p, int n)

{

if (n <= 1)

return 0;

else

return max(f(p+1,n-1),p[0]-p[1]);

}

int main()

{

int a[] = {3,5,2,6,4};

printf("%d", f(a,5));

}

Note: max(x,y) returns the maximum of x and y. The value printed by this program is

1. #include<stdio.h> **(GATE 2017)**

int \*assignval (int \*x, intval)

{

\*x = val;

return x;

}

void main ()

{

int \*x = malloc(sizeof(int));

if (NULL == x) return;

x = assignval (x,0);

if (x) {

x = (int \*)malloc(sizeof(int));

if (NULL == x) return;

x = assignval (x,10);

}

printf("%d\n", \*x);

free(x);

}

The code suffers from which one of the following problems:

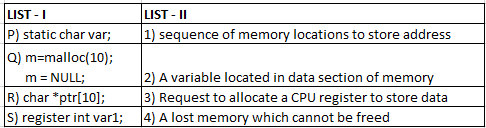
(A) compiler error as the return of malloc is not typecast appropriately.

(B) compiler error because the comparison should be made as x==NULL and not as shown.

(C) compiles successfully but execution may result in dangling pointer.

(D) compiles successfully but execution may result in memory leak.

1. Match the following **(GATE 2017)**



1. P-2, Q-4, R-1, S- 3 B) P-2, Q-1, R-4, S-3
2. P-2, Q-4, R-3, S-1 D) P-3, Q-4, R-1, S-2
3. Consider the following function implemented in C: **(GATE 2017)**

voidprintxy(int x, int y)

{

int \*ptr;

x = 0;

ptr = &x;

y = \*ptr;

\*ptr = 1;

printf("%d,%d", x, y);

}

The output of the printxy(1,1) is

(A) 0,0 (B) 0,1 (C) 1,0 (D) 1,1

1. Consider the following program **(GATE 2017)**

#include<stdio.h>

#include<string.h>

int main()

{

char \* c = "GATECSIT2017";

char \*p = c;

printf("%d", (int)strlen(c+2[p]-6[p]-1));

return 0;

}

The Output of the following program is \_\_\_\_\_\_\_\_

**Stacks & Queues**

1. The following sequence of operations is performed on stack: **(GATE 1991)**

PUSH(10),PUSH(20),POP,PUSH(10),PUSH(20),POP,POP,POP,PUSH(20),POP

The sequence of values popped out is:

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

1. Which of the following permutations can be obtained in the output (in the same order) using a stack assuming that the input is the sequence 1,2,3,4,5 in that order? **(GATE 1994)**

a)3,4,5,1,2 b)3,4,5,2,1 c)1,5,2,3,4 d)5,4,3,1,2

1. The postfix expression for the infix expression A+B\*(C+D)/F+D\*E is **(GATE 1995)**

a)AB+CD+\*F/D+E\*

b)ABCD+\*F/+DE\*+

c)A+B+CD/F\*DE++

d)A+\*BCD/F\*DE++

1. Which of the following statements are correct? **(GATE 1996)**
2. FIFO types of computations are efficiently supported by STACKS.
3. Implementing LISTS on linked lists is more efficient than implementing LISTS on arrays for almost all the basic LIST operations.
4. Implementing QUEUES on circular arrays is more efficient than implementing QUEUES on linear array with two indices.
5. LIFO type of computations are are efficiently supported by QUEUES.

Which of the following is correct?

a) ii and iii are true b) i and ii are true

c) iii and iv are true d) ii and iv are true

1. The best data structure to check whether an arithmetic expression has balanced parentheses is **(GATE 2004)**

a)queue b)stack c)tree d)list

1. Compute the post fix equivalent of the following infix expression **(GATE 1998)**

**3\* log(x+1) – a/2**

1. Suppose a stack implementation supports, in addition to PUSH,POP, an operation REVERSE, which reverses the order of the elements on the stack **(GATE 2000)**

A)To implement a queue using the above stack implementation, show how to implement ENQUEUE using a single operation and DEQUEUE using a sequence of 3 operations.

B) The following postfix expression containing single digit operands and arithmetic operators + and \*, is evaluated using a stack.

5 2 \* 3 4 + 5 2 \* \* +

Show the contents of the stack

i. After evaluating 5 2 \* 3 4 +

ii.After evaluating 5 2 \* 3 4 + 5 2

iii.After the end of evaluation

1. Let S be a stack of size n >= 1. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operation take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For m >= 1, define the stack-life of m as the time elapsed from the end of Push(m) to the start of the pop operation that removes m from S. The average stack-life of an element of this stack is

a) n(X+ Y) b) 3Y + 2X c) n(X + Y)-X d) Y + 2X **(GATE 2003)**

1. Assume that the operators +,-,\* are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, \*,+,-. The post fix expression corresponding to the infix expression a+b\*c-d^e^f is  **(GATE 2004)**

a)abc\*+def^^- b)abc\*+de^f^- c)ab+c\*d-e^f^ d)-+a\*bc^^def

1. A program attempts to generate as many permutations as possible of the string "abcd" by pushing the character a b c d in the same order onto stack, but it may pop off the top character at any time. Which cannot be generated using this program? **(GATE 2004)**a)abcd b)dcba c)cbad d)cabd
2. A function f defined on stacks of integers satisfies the following properties. f(@)=0; /\*@ denotes 'phi'\*/ **(GATE 2005)**  
    f(push(S,i)) = max(f(S),0)+i for all stacks and integers i.  
   If a stack contains the integers 2 ,-3 , 2 ,-1 , 2 in order from top to bottom, what is f(S)?  
   a) 6 b) 4 c) 3 d) 2
3. The following function computes the value of (mcn) correctly for all legal values m and n (m>=1,n>=0 and m>n) **(GATE 2006)**

intfunc(intm,int n)

{ if (E) return 1;

else

return (func(m-1,n)+func(m-1,n-1));

}

In the above function, which of the following is the correct expression for E?

a)(n==0) || (m==1) b)(n==0) && (m==1)

c)(n==0) || (m==n) d)(n==0) && (m=n)

1. An implementation of a queue Q, using two stacks S1 and S2, is given below: **(GATE 2006)**

|  |
| --- |
| void insert(Q, x)  {     push (S1, x);  }   void delete(Q)  {     if(stack-empty(S2)) then        if(stack-empty(S1)) then  {            print(“Q is empty”);            return;        }        else  while (!(stack-empty(S1)))  {            x=pop(S1);            push(S2,x);        }     x=pop(S2);  } |

Let n insert and m (<=n) delete operations be performed in an arbitrary order on an empty queue Q. Let x and y be the number of push and pop operations performed respectively in the process. Which one of the following is true for all m and n?  
a) n+m<= x < 2n and 2m <= y <= n+m  
b) n+m<= x < 2n and 2m<= y <= 2n  
c) 2m <= x < 2n and 2m <= y <= n+m  
d) 2m <= x <2n and 2m <= y <= 2n

1. The following postfix expression with single digit operands is evaluated using a stack:

**8 2 3 ^ / 2 3 \* + 5 1 \* - (GATE 2007)**

Note that ^ is the exponentiation operator. The top two elements of the stack after the first \* is evaluated are:

a)6,1 b)5,7 c)3,2 d)1,5

1. Suppose a stack implementation supports an instruction REVERSE, which reverses the order of elements on the stack, in addition to the PUSH and POP instructions. Which one of the following statements is TRUE with respect to this modified stack? **(GATE 2014)**

a) A queue cannot be implemented using this stack.

b) A queue can be implemented where ENQUEUE takes a single instruction and DEQUEUE takesa sequence of two instructions.

c) A queue can be implemented where ENQUEUE takes a sequence of three instructions and DEQUEUE takes a single instruction.

d) A queue can be implemented where both ENQUEUE and DEQUEUE take a single instructioneach.

1. Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:  **(GATE 2007)**

i. isEmpty(Q) – returns true if the queue is empty, false otherwise

ii.delete(Q) – deletes the elements at the front of the queue and returns its value

iii.insert(Q,i) – inserts the integer I at the rear of the queue.

Consider the following function:

void f(queue Q)

{

inti;

if(! isEmpty(Q))

{

i=delete(Q);

f(Q);

insert(Q,i);

}

}

What operation is performed by the above function?

a)Leave the queue Q unchanged

b)Reverse the order of the elements in the queue Q

c)Deletes the element at the front of the queue Q and insert it at the rear keeping the other elements in the same order

d)Empties the queue Q

1. Consider the following C program **(GATE 2007)**

#include<stdio.h>

#define EOF -1

void push(int);

int pop(void);

voidflagError();

int main()

{

intc,m,n,r;

while( (c=getchar() ) != EOF )

{

if (isdigit(c ) )

push(c );

else if ( (c==’+’) || (c==’\*’) )

{

m=pop();

n=pop();

r=(c==’+’) ?n+m: n\*m;

push(r );

}

else if( c!= ‘ ‘)

flagError();

}

printf(“%c”,pop() );

}

What is the output of the program for the following input?**5 2 \* 3 3 2 + \* +**

a)15 b)25 c)30 d)150

1. Consider the C program below. **(GATE 2015)**

#include<stdio.h>

int \*A, stkTop;

intstkFunc (intopcode, intval)

{

staticint size =0, stkTop=0;

switch (opcode)

{

case -1 : size = val;

break;

case 0 : if (stkTop< size)

A[stktop++] = val;

break;

default :

if (stktop)

return A [--stkTop];

}

return -1;

}

int main ( )

{ int B[20] ; A = B; stkTop = -1;

stkFunc (-1, 10);

stkFunc (0, 5);

stkFunc (0, 10);

printf (“%d\n”, stkFunc(1, 0) + stkfunc(1, 0) );

} The value printed by the above program is \_\_\_\_\_\_\_\_\_\_\_\_\_.

1. The result evaluating the postfix expression 10 5 + 60 6 / \* 8 − is \_**(GATE 2015)**

a) 284 b) 213 c) 142 d) 71

1. A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is CORRECT (n refers to the number of items in the queue)? **(GATE 2016)**

A) Both operations can be performed in O(1) time

B) At most one operation can be performed in O(1) time but the worst case time for the other operation will be Ω(n)

C) The worst case time complexity for both operations will be Ω(n)

D) Worst case time complexity for both operations will be Ω(log n)

1. Let Q denote a queue containing sixteen numbers and S be an empty stack. Head(Q) returns the element at the head of the queue Q without removing it from Q. Similarly Top(S) returns the element at the top of S without removing it from S. Consider the algorithm given below. **(GATE 2016)**

While Q is not Empty

do

if S is empty OR Top(S) <= Head(Q) then

x:= Dequeue(Q)

Push(S,x);

else

x:= Pop(S);

Enqueue(Q,x);

end

end

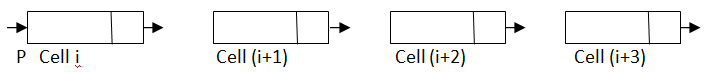
The maximum possible number of iterations of the while loop in the algorithm is\_\_\_\_\_

**LINKED LISTS**

1. In a circular linked list organization, insertion of a record involves modification of **(GATE 1987)**

a)One pointer b)Two pointers c)Multiple pointers d)No pointer

1. Let P be a pointer as shown in the figure in a singly linked list.  **(GATE 1998)**



What do the following assignment statements achieve?

i.q:=p🡪next ii.p🡪next := q🡪next

iii. q🡪next := (p🡪next)🡪next iv. (p🡪next)🡪next := q

1. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is **(GATE 2002)**

a)log2n b)n/2 c)log2n – 1 d)n

1. Let P be a single linked list, Let Q be the pointer to an intermediate node x in the list.What is the worst-case time complexity of the best-known algorithm to delete the node x from the list? **(GATE 2004)**

a)O(n) b)O(log2n) c)O(log n) d)O(1)

1. The following C function takes a singly linked list of integers as a parameter and rearranges the elements of the list. The list is represented as pointer to a structure. The function is called with the list containing the integers 1,2,3,4,5,6,7 in the given order. What will be contents of the list after the function completes execution?**(GATE 2005)**

struct node { int value; struct node \*next};

void rearrange (struct node \*list)

{

struct node \*p,\*q;

int temp;

if (! list || ! list🡪next)

return;

p=list;

q=list🡪next;

while(q)

{

temp=p🡪value;

p🡪value=q🡪value;

q🡪value=temp;

p=q🡪next;

q = p ?p🡪next : 0;

}

a)1,2,3,4,5,6,7 b)2,1,4,3,6,5,7 c)1,3,2,5,4,7,6 d)2,3,4,5,6,7,1

1. The following C function takes a singly linked list as input argument. It modified the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank. **(GATE 2010)**

typedefstruct node

{

int value;

struct node \*next;

} NODE;

NODE \*mode\_to\_front(NODE \*head)

{

NODE \*p,\*q;

if ( (head==NULL) || (head🡪next ==NULL) )

return head;

q=NULL; p=head;

while (p🡪next !=NULL)

{

q=p;

p=p🡪next;

}

………………………………………

return head;

}

Choose the correct alternative to replace the blank line.

a) q=NULL;p🡪next=head; head=p;

b) q🡪next =NULL;head=p;p🡪next=head;

c) head=p;p🡪next=q;q🡪next=NULL;

d) q🡪next=NULL;p🡪next=head;head=p;

1. N items are stored in a sorted doubly linked list. For a delete operation, a pointer is provided to the record to be deleted. For a decrease-key operation, a pointer is provided to the record on which the operation is to be performed. An algorithm performs the following operations on the list in this order: Θ(N) delete, O(log N) insert, O(log N) find, and Θ(N) decrease-key What is the time complexity of all these operations put together **(GATE 2016)**

A)O(Log2N) B) O(N) C) O(N2) D) Θ(N2 Log N)

1. Consider the following C code fragment **(GATE 2017)**

typedefstruct node

{

int data;

node\* next ;

} node;

void join(node\* m, node\* n)

{

node\* p = n;

while (p->next != NULL)

p = p->next;

p–>next = m;

}

Assuming that m and n point to valid NULL- terminated linked lists, invocation of join will

(A) append list m to the end of list n for all inputs

(B) either cause a null pointer dereference or append list m to the end of list n

(C) cause a null pointer dereference for all inputs.

(D) append list n to the end of list m for all inputs.

1. A Circular queue has been implemented using singly linked list where each node consists of a value and a pointer to next node. We maintain exactly two pointers FRONT and REAR pointing to the front node and rear node of queue. Which of the following statements is/are correct for circular queue so that insertion and deletion operations can be performed in O(1) i.e. constant time. **(GATE 2017)**

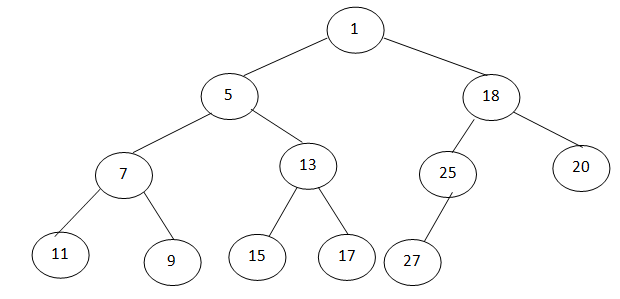
I. Next pointer of front node points to the rear node.

II. Next pointer of rear node points to the front node.

(A) I only (B) II only (C) Both I and II (D) Neither I nor II

**TREES**

1. Is it possible to construct a binary tree uniquely whose pre-order and post-order traversals are given?  **(GATE 1987)**
2. If the number of leaves in a tree is not a power of 2, then the tree is not a binary tree. (True/ False)  **(GATE 1987)**
3. Consider the binary tree as shown below **(GATE 1991)**
4. What structure is represented by the binary tree?



1. Give the steps for deleting the node with key 5 so that the structure is preserved.
2. A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is

a) log2n b) n-1 c) n d) 2n **(GATE 1995)**

1. What is the number of binary trees with 3 nodes which when traversed in post order give the sequence A,B,C? Draw all such possible binary trees **(GATE 1995)**
2. The binary search tree is generated by inserting in order the following integers:

**50, 15, 62, 5, 20, 58, 91,3, 8,37, 60, 60, 24 (GATE 1996)**

The number of nodes in the left subtree and right subtree of the root respectively is

a) (4,7) b)(7,4) c) (8,3) d) (3,8)

1. A binary search tree contains values 1,2,3,4,5,6,7,8. The tree is traversed in preorder and the values are printed out. Which of the following sequences is a valid output? **(GATE 1997)**

a)5 3 1 2 4 7 8 6 b)5 3 1 2 6 4 8 7 c)5 3 2 4 1 6 7 8 d)none

1. A complete n-ary tree is one which every node has 0 or n sons. If x is the number of internal nodes of a complete n-ary tree, the number of leaves in it is given by **(GATE 1998)**

a) x(n-1)+1 b) xn-1 c) xn+1 d) x(n+1)

1. Draw the min-heap that results from insertion of the following elements in order into an initially empty min-heap: 7,6,5,4,2,3,1. Show the result after the deletion of this heap. **(GATE 1999)**
2. Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right sub trees respectively of node X. Note that Y and Z may be NULL, or further nested. Which of the following represents a valid binary tree? **(GATE 2000)**
3. (1 2 (4 5 6 7)) b)(1 ((2 3 4) 5 6) 7)

c) (1 (2 3 4) (5 6 7)) d) (1 (2 3 null) (4 5))

1. Let LASTPOST, LASTIN, LASTPRE denote the last vertex visited in a post order, inorder, preorder traversals respectively, of a complete binary tree. Which of the following is always true? **(GATE 2000)**

a)LASTIN = LASTPOST b)LASTIN = LASTPRE

c)LASTPRE = LASTPOST d) None of these

1. The no ofinternal nodes in a rooted tree of n nodes, each node having 0 or 3 children is **(GATE 2002)**

a) n/2 b) (n-1)/3 c) (n-1)/2 d) (2n-1)/3

1. The following number are inserted into an empty binary search tree in the given order: 10,1,3,15,12,16. What is the height of the binary search tree? **(GATE 2004)**

a) 2 b) 3 c) 4 d) 6

1. Level order traversal of a rooted tree can be done by starting from the root and performing **(GATE 2004)**

a) Preorder traversal b) inorder traversal

c) depth first search d) breadth first search

1. How many distinct binary search trees can be created out of 4 distinct keys? **(GATE 2005)**

a) 5 b) 14 c) 24 d) 42

1. Consider the following C program segment **(GATE 2004)**

structCellNode

{

  structCelINode \*leftchild;

  int element;

  structCelINode \*rightChild;

}

intDosomething(structCelINode \*ptr)

{

    int value = 0;

    if (ptr != NULL)

    {

      if (ptr->leftChild != NULL)

        value = 1 + DoSomething(ptr->leftChild);

      if (ptr->rightChild != NULL)

        value = max(value, 1 + DoSomething(ptr->rightChild));

    }

    return (value);

}

The value returned by the function DoSomething when a pointer to the root of a non-empty tree is passed as argument is

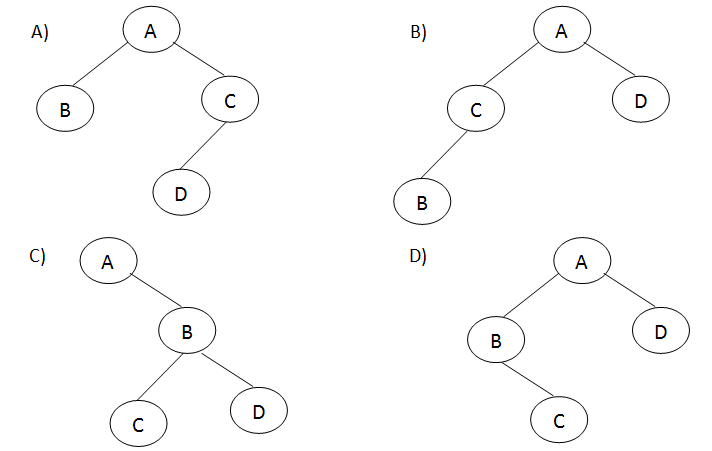
a) The number of leaf nodes in the tree

b) The number of nodes in the tree

c) The number of internal nodes in the tree

d) The height of the tree

1. Which one of the following binary trees has its inorder and preorder traversal as BCAD and ABCD respectively?  **(GATE 2004)**



1. Post order traversal of a given binary search tree. T produces the following sequence of keys

**10,9,23,22,27,25,15,50,95,60,40,29 (GATE 2005)**

Which of the following sequences of keys can be the result of an **inorder** traversal of the tree T?

1. 9,10,15,22,23,25,27,29,40,50,60,95
2. 9,10,15,22,40,50,60,95,23,25,27,29
3. 29,15,9,10,25,22,23,27,40,60,50,95
4. 95,50,60,40,27,23,22,25,10,9,15,29
5. In a complete k-ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal nodes is:  **(GATE 2005)**

a) nk b) (n-1) k +1 c) n(k-1) +1 d) n(k-1)

1. The numbers 1,2,---,n are inserted in a binary search tree in some order. In the resulting tree, the right sub tree of the root contains p nodes. The first number to be inserted in the tree much be **(GATE 2005)**
2. p b) p+1 c) n-p d) n-p+1
3. A binary search tree contains the numbers 1,2,3,4,5,6,7,8. When the tree is traversed in pre-order and the values in each node printed out, the sequence of values obtained is 5,3,1,2,4,6,8,7. If the tree is traversed in post-order, the sequence obtained would be **(GATE2005)**
4. 8,7,6,5,4,3,2,1 b) 1,2,3,4,8,7,6,5
5. c) 2,1,4,3,6,7,8,5 d) 2,1,4,3,7,8,6,5
6. In a binary tree, the number of internal nodes of degree 1 is 5, and the number of internal nodes of degree 2 is 10. The number of leaf nodes in the binary tree is **(GATE 2006)**
7. 10 b) 11 c) 12 d) 15
8. Which of the following sequences of array elements forms a heap? **(GATE 2006)**
9. {23,17,14,6,13,10,1,12,7,5}
10. {23,17,14,6,13,10,1,5,7,12}
11. {23,17,14,7,13,10,1,5,6,12}
12. {23,17,14,7,13,10,1,12,5,7}
13. The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is: **(GATE 2007)**

a)2h – 1 b) 2h-1 – 1 c) 2h+1 – 1 d) 2h+1

1. Suppose that we have number between 1 and 100 in a binary search tree and want to search for the number 55. Which of the following sequences CANNOT be the sequence of nodes examined **(GATE 2006)**
2. {10,75,64,43,60,57,55}
3. {90,12,68,34,62,45,55}
4. {9,85,47,68,43,57,55}
5. {79,14,72,56,16,53,55}
6. The maximum number of binary trees that can be formed with three unlabeled nodes is **(GATE 2007)**
7. 1 b) 5 c)4 d) 3
8. The inorder and preorder traversal of a binary tree are “d b e a f c g” and“a b d e c f g” respectively. The post order traversal of the binary tree is: **(GATE 2007)**
9. d e b f g c a b) e d b g f c a c) e d b f g c a d) d e f g b c a
10. A complete n- aryis a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L=41, and I=10, what is the value of n? **(GATE 2007)**
11. 3 b) 4 c) 5 d) 6
12. Which of the following is TRUE? **(GATE 2008)**
13. The cost of searching an AVL tree is O(log n) but that of a binary search tree is O(n)
14. The cost of searching an AVL tree is O(log n) but that of a complete binary tree is O(n logn)
15. The cost of searching a binary search tree is O(log n) but that of an AVL tree is O(n)

d) The cost of searching an AVL tree is O(n log n) but that of a binary search tree is O(n)

1. The following three are known to be the preorder, inorder and postorder sequences of a binary tree. But it is not known which is which. **(GATE 2008)**
2. MBCAFHPYK
3. KAMCBYPFH
4. MABCKYFPH

Pick the true statement from the following.

a) I and II are preorder and inorder sequences, respectively

b) I and III are preorder and post order sequences, respectively

c) II is the inorder sequence. But nothing more can be said about the other two sequences

d) II and III are the preorder and inorder sequences, respectively.

1. Consider the following C program segment where CellNode represents a node in a binary tree: **(GATE 2007)**

structCellNode

{

  structCellNOde \*leftChild;

  int element;

  structCellNode \*rightChild;

};

intGetValue(structCellNode \*ptr)

{

  int value = 0;

  if (ptr != NULL)

  {

   if ((ptr->leftChild == NULL) && (ptr->rightChild == NULL))

      value = 1;

   else

      value = value + GetValue(ptr->leftChild) + GetValue(ptr->rightChild);

  }

  return(value);

}

|  |
| --- |
|  |

The value returned by GetValue() when a pointer to the root of a binary tree is passed as its argument is:

a) the number of nodes in the tree  
b) the number of internal nodes in the tree  
c) the number of leaf nodes in the tree  
d) the height of the tree

**COMMON DATA QUESTION**

A binary search tree stores values in the range 37 to 573. Consider the following sequences of keys.

1. 81,537,102,439,285,376,305
2. 52,97,121,195,242,381,472
3. 142,248,520,386,345,270,307
4. 550,149,507,395,463,402,270
5. Suppose the BST has been unsuccessfully searched for key 273. Which all of the above sequences list nodes in the order in which we could have encountered them in the search?: **(GATE 2008)**
6. II and III only B) I and III only C)III and IV only D) III only
7. Which of the following statements is TRUE? **(GATE 2008)**
8. I,II and IV are inorder sequences of three different BSTs
9. I is a preorder sequence of some BST with 439 as the root
10. II is an inorder sequence of some BST where 121 is the root and 52 is a leaf
11. IV is a postorder sequence of some BST with 149 as the root

**COMMON DATA QUESTION**

A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, a[0],nodes in the next level, from left to right, is stored from a[1] to a[3]. The nodes from the second level of the tree from left to right are stored from a[4] location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location a[n] and pushing it up the tree to satisfy the heap property. **(GATE 2006)**

1. Which one of the following is a valid sequence of elements in an array representing 3-ary max heap?

a) 1, 3, 5, 6, 8, 9                  b) 9, 6, 3, 1, 8, 5

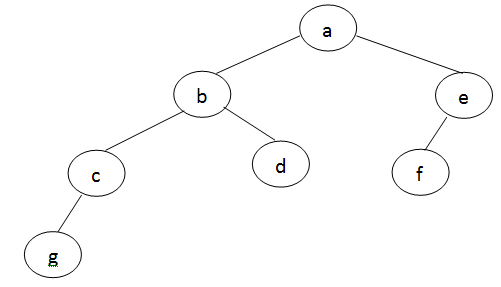
c) 9, 3, 6, 8, 5, 1             d) 9, 5, 6, 8, 3, 1

1. Suppose the elements 7, 2, 10 and 4 are inserted, in that order, into the valid 3-ary max heap found in the above question, which one of the following is the sequence of items in the array representing the resultant heap?

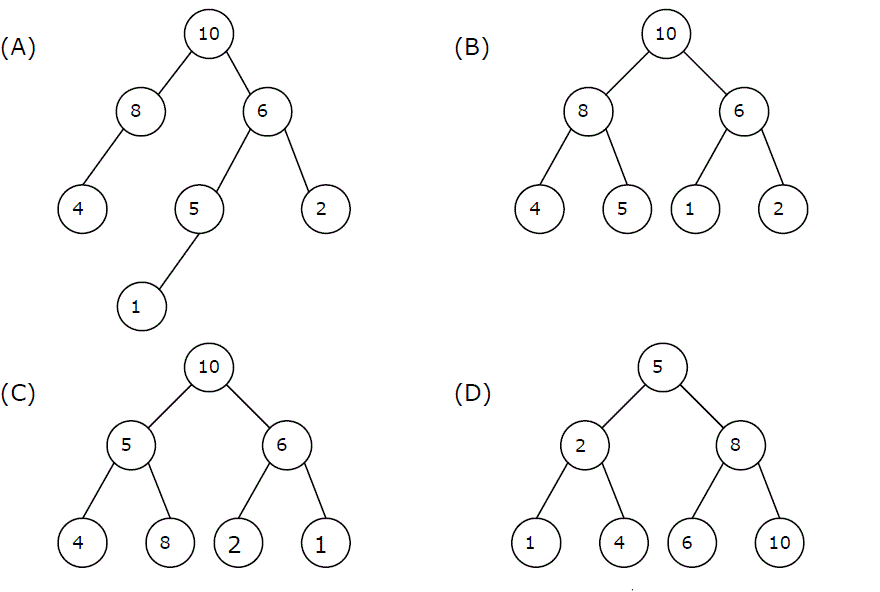
a) 10, 7, 9, 8, 3, 1, 5, 2, 6, 4 b) 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

c) 10, 9, 4, 5, 7, 6, 8, 2, 1, 3                     d) 10, 8, 6, 9, 7, 2, 3, 4, 1, 5

1. How many distinct BSTs can be constructed with 3 distinct keys? **(GATE 2008)**
2. 4 b) 5 c) 6 d) 9
3. What is the maximum height of AVL tree with 7 nodes? Assume that the height of a tree with a single node is 0 **(GATE 2009)**
4. 2 b) 3 c) 4 d) 5
5. We are given a set of n distinct elements and an unlabeled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree? **(GATE 2011)**
6. 0 b) 1 c) n! d) 2n Cn . 1 / (n+1)
7. In the balanced binary tree in figure given below, how many nodes will become unbalanced when a node is inserted as a child of the node “g”? **(GATE 1996)**



1. 1 b) 3 c) 7 d) 8
2. **A max-heap is a heap where the value of each parent is greater than or equal to the values of its children. Which of the following is a max-heap? (GATE 2011)**

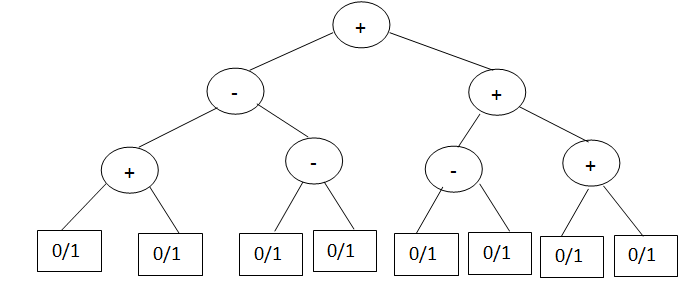
**[](http://geeksforgeeks.org/wp-content/uploads/gate_2011_2.gif)**

1. The preorder traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the postorder traversal sequence of the same tree? **(GATE 2013)**a) 10, 20, 15, 23, 25, 35, 42, 39, 30  
   b) 15, 10, 25, 23, 20, 42, 35, 39, 30  
   c) 15, 20, 10, 23, 25, 42, 35, 39, 30  
   d) 15, 10, 23, 25, 20, 35, 42, 39, 30
2. A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is: **(GATE 2014)**

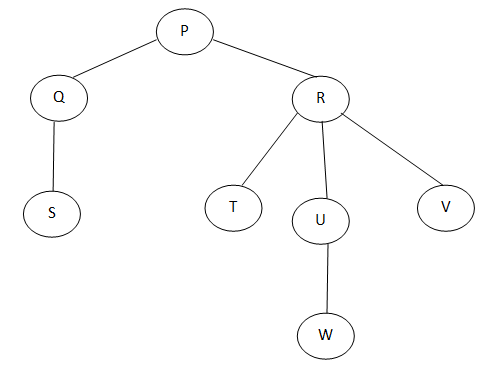
a) 10, 8, 7, 3, 2, 1, 5 b) 10, 8, 7, 2, 3, 1, 5

c) 10, 8, 7, 1, 2, 3, 5 d) 10, 8, 7, 5, 3, 2, 1

1. Consider the expression tree shown. Each leaf represents a numerical value, which can either be 0 or 1. Over all possible choices of the values at the leaves, the maximum possible value of the expression represented by the tree is \_\_\_. **(GATE 2014)**



1. Consider the following rooted tree with the vertex labeled P as root:**(GATE 2014)**



The order in which the nodes are visited during an in-order traversal of a tree is

1. SQPTRWUV b) SQPTUWRV

c) SQPTWUVR d) SQPTRUWV

1. What are the worst-case complexities of insertion and deletion of a key in a binary search tree? **(GATE 2015)**

a) θ(logn) for both insertion and deletion

b) θ(n) for both insertion and deletion

c) θ(n) for insertion and θ(logn) for deletion

d) θ(logn) for insertion and θ(n) for deletion

1. Which of the following is/are correct inorder traversal sequence(s) of binary search tree(s)? **(GATE 2015)**

1) 3,5,7,8,15,19,25 2) 5, 8, 9,12,10,15,25

3) 2,7,10,8,14,16,20 4) 4,6,7,9,18,20,25

a) 1 and 4 only b) 2 and 3 only c) 2 and 4 only d) 2 only

1. The height of a tree is the length of the longest root-to-leaf path in it. The maximum and minimum number of nodes in a binary tree of height 5 are **(GATE 2015)**

a) 63 and 6 b) 64 and 5 c) 32 and 6 d) 31 and 5

1. Consider a max heap, represented by the array: 40, 30, 20, 10, 15, 16, 17, 8, 4. **(GATE 2015)**

Array Index 1 2 3 4 5 6 7 8 9

Value 40 30 20 10 15 16 17 8 4

Now consider that a value 35 is inserted into this heap. After insertion, the new heap is

a) 40, 30, 20, 10, 15, 16, 17, 8, 4, 35 b) 40, 35, 20, 10, 30, 16, 17, 8, 4, 15

c) 40, 30, 20, 10, 35, 16, 17, 8, 4, 15 d) 40, 35, 20, 10, 15, 16, 17, 8, 4, 30

1. A binary tree T has 20 leaves. The number of nodes in T having two children is \_\_\_\_\_\_\_. **(GATE 2015)**
2. While inserting the elements 71,65,84,69,67,83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is **(GATE 2015)**

a) 65 b) 67 c) 69 d) 83

1. Consider a binary tree T that has 200 leaf nodes. Then, the number of nodes in T that have exactly two children are \_\_\_\_ **(GATE 2015)**
2. An operator delete(i) for a binary heap data structure is to be designed to delete the item in the i-th node. Assume that the heap is implemented in an array and i refers to the i-th index of the array. If the heap tree has depth d (number of edges on the path from the root to the farthest leaf), then what is the time complexity to re-fix the heap efficiently after the removal of the element? **(GATE 2016)**

A) O(1) B) O(d) but not O(1)

C) O(2d) but not O(d) D) O(d2d) but not O(2d)

1. A complete binary min-heap is made by including each integer in [1, 1023] exactly once. The depth of a node in the heap is the length of the path from the root of the heap to that node. Thus, the root is at depth 0. The maximum depth at which integer 9 can appear is \_\_\_\_\_ **(GATE 2016)**
2. Consider the following New-order strategy for traversing a binary tree: Visit the root; Visit the right subtree using New-order Visit the left subtree using New-order The New-order traversal of the expression tree corresponding to the reverse polish expression 3 4 \* 5 - 2 ˆ 6 7 \* 1 + - is given by: **(GATE 2016)**

A) + - 1 6 7 \* 2 ˆ 5 - 3 4 \*

B) - + 1 \* 6 7 ˆ 2 - 5 \* 3 4

C) - + 1 \* 7 6 ˆ 2 - 5 \* 4 3

D) 1 7 6 \* + 2 5 4 3 \* - ˆ -

1. The number of ways in which the numbers 1, 2, 3, 4, 5, 6, 7 can be inserted in an empty binary search tree, such that the resulting tree has height 6, is \_\_\_\_\_\_\_\_\_\_\_\_\_ Note: The height of a tree with a single node is 0. **(GATE 2016)**
2. Let T be binary search tree with 15 nodes. The minimum and maximum possible heights of T respectively are: **(GATE 2017)**
3. 4 and 15 b) 3 and 14 c) 4 and 14 d) 3 and 15