IV

Programming & Data Structures

Contents

1. Programming 125

2. Linked List, Stack, Queue and Hashing 139

3. Trees 150

Description Sheet

PROGRAMMING AND DATA STRUCTURES

Chapter 1 : Programming

- Basics of C programming
- Memory segments
- Storage classes
- Scope of variable
- Recursion
- Pointers
  - Pointer to pointer
  - Array of pointers
  - Multidimensional arrays
- Strings
- Arrays base address calculation
- Union and structures
- Dynamic memory allocation

Chapter 2 : Linked List, Stack, Queue and Hashing

- Singly linked list
- Operations on single linked list
- Double linked list
- Circular single and double linked list
- Stack implementation using array and linked list
- Prefix, infix and postfix
- Postfix evaluation
- Queue implementation using array and linked list
- Double ended queue
- Implementing stack using queues
- Implementing queue using stacks
- Collisions
- Hash functions
- Collision resolution techniques

Chapter 3 : Trees

- Tree programming
- Constructing unique binary tree
- Tree traversals
- Binary search trees
- AVL Tree
- Rotation techniques, insertions and deletions

■■■■

1 Programming

Multiple Choice Questions

Q.1 The following C declarations:

```
struct node
{   int i:
    float j;
};
struct node *s[10];
```

define *s* to be

(a) an array, each element of which is a pointer to a structure of type node

(b) a structure of 2 fields, each field being a pointer to an array of 10 elements

(c) a structure of 3 fields: an integer, a float, and an array of 10 elements

(d) an array, each element of which is a structure of type node

[GATE-2000]

Q.2 The goal of structured programming is to

(a) have well indented programs

(b) be able to infer the flow of control from the compiled code

(c) be able to infer the flow of control from the program text

(d) avoid the use of GOTO statements

[GATE-2004]

Q.3 An Abstract Data Type (ADT) is

(a) same as an abstract class

(b) a data type that cannot be instantiated

(c) a data type for which only the operations defined on it can be used, but none else

(d) all of the above

[GATE-2005]

Q.4 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?

(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

[GATE-2005]

Q.5 Consider the following C function.

```
float f (float x, int y)
{
    float p, s; int i;
    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) X^y (b) e^x

(c) $\ln(1 + x)$ (d) X^x

[GATE-2003]

Q.6 Assume the following C variable declaration

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

Of the following expressions

1. A[2]
2. A[2][3]
3. B[1]
4. B[2] [3]

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

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

[GATE-2003]

Q.7 The C language is:
(a) a context free language
(b) a context sensitive language
(c) a regular language
(d) parsable fully only by a Turing machine

[GATE-2002]

Q.8 What is printed by the print statements in the program P1 assuming call by reference parameter passing?

```
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 these

[GATE-2001]

Q.9 Consider the following program:

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      \\beginP2
        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) 4

[GATE-2001]

Q.10 In the C language
(a) at most one activation record exists between the current activation record and the activation record for the main
(b) the number of activation records between the current activation record and the activation record for the main depends on the actual function calling sequence
(c) the visibility of global variables depends on the actual function calling sequence.
(d) recursion requires the activation record for the recursive function to be saved on a different stack before the recursive fraction can be called

[GATE-2002]

Q.11 Consider the following C declaration

```
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

[GATE-2000]

Q.12 An unrestricted use of the "goto" statement is harmful because
(a) it makes it more difficult to verify programs
(b) it increases the running time of the programs
(c) it increases the memory required for the programs
(d) it results in the compiler generating longer machine code

[GATE-1994]

Q.13 Consider the following C function:

```
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

[GATE-2007]

Q.14 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", k);
}
    int main ( )
{
    int a = 2048, sum = 0;
          foo (a, sum);
    printf ("% d/n", sum);
}
```

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

[GATE-2005]

Q.15 Consider the following C program

```
main( )
{
    int  x, y, m, n;
    scanf("% d %d", &x, &y);
    / * Assume 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 y
(d) the least common multiple of x and y

[GATE-2004]

Q.16 Consider the following C function

```
int f(int n)
{   static int i = 1;
    if (n > = 5) return n;
    n = n + i;
    i++;
    return f(n) ;
}
```

The value returned by $f(1)$ is

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

[GATE-2004]

Q.17 Consider the following C-function:

```
int f (int a, int b)
{
    if (b > a) return (f(b, a));
    else if (b == 0) return (a);
         else return (f(b, a%b));
}
```

Which of the following will be returned by the function call $f(18, 30)$?

(a) 2 (b) 3
(c) 6 (d) 8

[DRDO-2009]

Q.18 In a C program how many bytes are required to store the string "john" in a character array"

(a) 2 (b) 3
(c) 4 (d) 5

[DRDO-2009]

Q.19 Consider a declaration

```
int a = 5, *b = & a;
printf ("%d", a*b);
```

The output is

(a) 25 (b) garbage
(c) 5 * address of *b* (d) error message

[JNUEE-2009]

Q.20

```
main ( )
{
    void *vp;
    char ch = 'g';
    char *cp = "goofy";
    int j = 20;
    vp = &ch;
    printf("%c", * (char *) vp);
    vp = &j;
    printf("%d", *(int *) vp);
    vp = cp;
    printf("%s", (char *) vp + 3);
}
```

(a) g20fy (b) goofy
(c) go2fy (d) None of these

Q.21 For the following program:

```
main( ) {
    inc( ); inc( ); inc( );
}
inc( ) {
    static int x;
    printf("%d", ++x);
}
```

the output

(a) prints 0, 1, 2
(b) prints 1, 2, 3
(c) prints 3 consecutive but unpredictable numbers
(d) prints 111

[JNUEE-2009]

Q.22 Which combination of the integer variables x, y and z makes the variable a get the value 4 in the following expression?

$a = (x > y)\ ?\ ((x > z)\ ?\ x : z) : ((y > z)\ ?\ y : z)$

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

[GATE-2008]

Q.23

```
main ( ) {
    int i = 300;
    char *ptr = (char*)&i;
    *++ptr = 2;
    printf("%d", i);
}
```

(a) 556 (b) 300
(c) 655 (d) 003

Q.24 What does the following program print?

```
#include < stdio.h>
void f(int *p, int *q) {
        p = q;
        *p = 2;
}
int i = 0, j = 1;
int main ( ) {
        f(&i, &j);
        printf("%d %d\n", i, j);
        return 0;
}
```

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

[GATE-2010]

Q.25 The following program is to be tested for statement coverage:

```
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 and S4?

(a) T1, T2, T3 (b) T2, T4
(c) T3, T4 (d) T1, T2, T4

[GATE-2010]

Q.26 Consider the following code:

```
int x = 0, i;
for (i = 0; i < 10; i ++)
    if (i % 2 && x ++)
        x + = 2;
```

What will be the value of x?

(a) 11 (b) 13
(c) 15 (d) 17

Q.27 Consider the following function implemented in C:

```
void printxy (int x, int y)
{
    int *ptr;
    x = 0;
    ptr = &x;
    y = *ptr;
    *ptr = 1;
    printf ("%d, %d", x, y);
}
```

The output of invoking printxy (1, 1) is

(a) 0, 0 (b) 0, 1
(c) 1, 0 (d) 1, 1

[GATE-2017]

Q.28 Consider the following code:

```
int a[10], *p, *q;
p = &a[5];
q = &a[7];
```

Which of the following statement is incorrect with respect to pointers?

(a) $p + 1$ (b) $q - 3$
(c) $p + q$ (d) $q - p$

Q.29 Consider the following code:

```
int a = 32, b = 2, c = 3;
Switch (X)
{   Case 2: printf("%d", a);
    Case 4: printf("%d", b);
    Case 6: break;
    Case 8: printf("%d", c);
    default: printf("%d", b);
}
```

Find the missing statement X, if the above 'C' code prints the output as 32.

(a) $b \times c$ (b) $b \times c - 2$
(c) $b + c \times 2$ (d) None of these

Q.30 What is the output of the following program?

```
int f(int a)
{
    printf ("%d", a++);
    return (++a);
}
main( )
{
    int b = 1;
    b = f(b);
    b = f(b);
    b = f(1 + f(b));
}
```

(a) 1 3 3 5
(b) 1 3 5 8
(c) 2 3 3 5
(d) 2 4 6 8

Q.31 Consider the following C program:

```
#include <stdio. h>
void f(int x, int * p)
{   *p = x;
    x = 10;
}
int main ( )
{   int a = 5,  b = 6;
    int *p = &a, **q;
    *p = 20; q = &p;
    f(a, &b);
    *q = &b;
    *p = 30;
    printf("%d, %d", a, b);
}
```

What is the output product by above C program

(a) 10, 20 (b) 20, 30
(c) 30, 10 (d) 20, 20

Q.32 Consider the following function written in the C programming language.

```
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

Q.33 Let a be an array containing n integers in increasing order. The following algorithm determines whether there are two distinct numbers in the array whose difference is a specified number S > 0.

```
i = 0; j = 1;
while (j < n) {
    if (E) j++;
    else if (a[j] – a[i] = = S) break;
    else i++;
}
if (j < n) printf("yes");
else printf("no");
```

Choose the correct expression for E.

(a) $a[j] - a[i] > S$ (b) $a[j] - a[i] < S$
(c) $a[i] - a[j] < S$ (d) $a[i] - a[j] > S$

[GATE-2005]

Q.34 Which one of the choices given below would be printed when the following program is executed?

```
#include <stdio.h>
int a1[ ] = {6, 7, 8, 18, 34, 67};
int a2[ ]= {23, 56, 28, 29};
int a3[ ] = {–12, 27, –31};
int *x[ ] = {a1, a2, a3};
void print(int *a[ ])
{
    printf("%d,", a[0][2]);
    printf("%d,", *a[2]);
    printf("%d,", *++a[0]);
    printf("%d,", *(++a)[0]);
    printf("%d\n", a[–1][+1]);
}
main( )
{
    print(x);
}
```

(a) 8, –12, 7, 23, 8
(b) 8, 8, 7, 23, 7
(c) –12, –12, 27, –31, 23
(d) –12, –12, 27, –31, 56

[GATE-2006]

Q.35 Consider the following three C functions:

```
[P1] int *g (void)
     {
          int x = 10;
          return (&x);
     }
[P2] int *g (void)
     {
          int *px;
          *px = 10;
          return px;
     }
[P3] int *g(void)
     {
          int *px;
          px = (int *)malloc (size of (int));
          *px = 10;
          return px;
     }
```

Which of the above three functions are likely to cause problems with pointers?

(a) Only P3 (b) Only P1 and P3
(c) Only P1 and P2 (d) P1, P2 and P3

[GATE-2001]

Q.36 Consider the C program shown below.

```
# 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 this program is

(a) 12 7 6 (b) 22 12 11

(c) 14 6 6 (d) 7 6 6

[GATE-2003]

Q.37 Consider the following C code:

```
#include <stdio.h>
int *assignval (int *x, int val)
{   *x = val;
    return x;
}
void main( )
{   int *x = malloc (size of (int));
    if (NULL == x) return;
    x = assignval(x, 0);
    if(x)
    {  x = (int*) malloc (size of (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

[GATE-2017]

Q.38 Consider the following C program:

```
#include <stdio.h>
typedef struct
{
    char *a;
    char *b;
} t;
void f1(t s);
void f2(t *p);
main( )
{
    static t s = {"A", "B"};
    printf("%s %s\n", s.a, s.b);
    f1(s);
    printf("%s %s\n", s.a, s.b);
    f2(&s);
}
void f1(t s)
{
    s.a = "U";
    s.b = "V"
    printf(%s %s\n", s.a, s.b);
    return;
}
    void f2(t *p)
{
    p → a  = "V";
    p → b = "W"
    printf("%s %s\n", p → a, p → b);
    return;
}
```

What is the output generated by the program?

(a) A B
U V
V W
V W

(b) A B
U V
A B
V W

(c) A B
U V
U V
V W

(d) A B
U V
V W
U V

[GATE-2004]

Q.39 Consider the following C program which is supposed to compute the transpose of a given 4 × 4 matrix M. Note that, there is an X in the program which indicates some missing statements. Choose the correct option to replace X in the program.

```
#include <stdio.h>
#define ROW 4
#define COL 4
int M[ROW][COL] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
main( )
{
    int i, j, t;
    for (i = 0; i < 4; ++i)
    {
        X
    }
    for (i = 0; i < 4; ++i)
        for (j = 0; j < 4; ++j)
            printf("%d", M[i][j]);
}
```

(a)
```
for(j = 0; j < 4; ++j)
{
    t = M[i][j];
    M[i][j] = M[j][i];
    M[j][i] = t;
}
```

(b)
```
for(j = 0; j < 4; ++j)
{
    M[i][j] = t;
    t = M[j][i];
    M[j][i] = M[i][j];
}
```

(c)
```
for(j = i; j < 4; ++j)
{
    t = M[j][i];
    M[i][j] = M[j][i];
    M[j][i] = t;
}
```

(d)
```
for(j = i; j < 4; ++j)
{
    M[i][j] = t;
    t = M[j][i];
    M[j][i] = M[i][j];
}
```

[GATE-2004]

Q.40 Consider the following C program.

```
void f (int, short);
void main( )
{
    int i = 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) $f(i, *p)$

[GATE-2016]

Numerical Answer Type Questions

Q.41 The output of executing the following C program is _____.

```
#include <stdio.h>
int total (int v)
{   static int count = 0;
    while (v)
    {   count+ = v & 1;
        v >> = 1;
    }
    return count;
}
void main ( )
{   static int x = 0;
    int i = 5;
    for (; i > 0; i --) {
        x = x + total (i);
    }
    printf ("%d\n", x);
}
```

[GATE-2017]

Q.42 Consider the following C program:

```
void main( )
{
    int i = 1, j = 1;
    for (; ;)
    {
        if (i > 5)
            break;
        else
            j += i;
        printf("\n%d", j);
        i += j;
    }
}
```

The output of the program is _____.

Q.43 What is printed by the following C program?

```
void main( )
{   int c, *b, **a;
    c = 4;
    b = &c;
    a = &b;
    printf("% d", f(c, b, a));
}
int f ( int x, int *py, int ** ppz)
{   int y, z;
    **ppz + = 1;
    z = **ppz;
    *py + = 2;
    y = *py;
    x + = 3;
    return (x + y + z);
}
```

Q.44 The output of the following C program is _____.

```
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;
}
int main( )
{   int a = 4, b = 5, c = 6;
    f1 (a, b);
    f2 (&b, &c);
    printf ("%d", c – a – b);
}
```

Q.45 What is the value printed by the following C program?

```
#include < stdio.h>
int f(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};
    print f("%d", f(a, 6));
    return 0;
}
```

Q.46 Find the output of the following function call A(6)

```
A(n)
{   if (n < 1) return (1);
    else return A(n – 2) + B(n – 1);
}
B(n)
{   if (n ≤ 1) return (1);
    else return B(n – 1) + A(n – 2);
}
```

Q.47 Consider the following C program:

```
# include <stdio.h>
int main( ) {
    int i, j, k = 0;
    j = 2 * 3 / 4 + 2.0 / 5 + 8 / 5;
    k – = – –j;
    for(i = 0; i < 5; i++) {
        switch(i + k)
        {
            case 1:
            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 _____.

[GATE-2015]

Q.48 Consider the C program below.

```
# include <stdio.h>
int *A, stkTop;
int stkFunc (int opcode, int val)
{
    static int 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 _____.

[GATE-2015]

Q.49 Consider the following snippet of a C program. Assume that swap (&x, &y) exchanges the contents of x and y.

```
int main( ) {
    int array[ ] = {3, 5, 1, 4, 6, 2};
    int done = 0;
    int i;
    while {done == 0)
    {   done = 1;
        for (i = 0; i <= 4; i++)
        {
            if (array[i] < array[i + 1])
            {
                swap(&array[i], &array[i + 1]);
                done = 0;
            }
        }
        for (i = 5; i >= 1; i – –) {
            if (array[i] > array[i – 1])
            {
                swap(&array[i], &array[i – 1]);
                done = 0;
            }
        }
        printf("%d", array[3]);
}
```

The output of the program is _____.

[GATE-2017]

Multiple Select Questions

Q.50 Consider the following program:

```
#include <stdio.h>
int main( )
{
    char arr[6] = {10, 20, 30, 40, 50, 0};
    char *ptr = (char *)(& arr + 1);
    printf ("%d%d", *(arr + 1), *(ptr – 2));
}
```

Which of the below statements are TRUE?

(a) output printed is 20, 50
(b) output printed is 20 and garbage value
(c) ptr is pointing next continuous 1D array
(d) ptr is pointing element of an array after type casting (address after the last element of array)

Q.51 Let R denotes the class of recursive programs and I denotes the class of iterative programs. Which of the following correct?

(a) Some programs belonging to class R do not terminate sometimes.
(b) For every program belonging to class L, there exists an equivalent program belonging to class R.
(c) Every program in R uses strictly more stack space compare to its equivalent program in I.
(d) Every program belonging to class I, also uses stack.

Q.52 int a[10][10] = {0, 1, ..., 99}
In the above declaration, which of the following is TRUE?
(a) & a represents the base address of whole array.
(b) a represents the base address 0^{th} 1D-array.
(c) a[0] represents the base address of 0^{th} element of 0^{th} 1D array.
(d) a[0] represents the base address of 0^{th} 1D array.

Q.53 Consider the string str, declared as "Madeeasy". Assuming the size of a character is 1 byte. Choose the most appropriate choices.
(a) strlen (str) output the value 8
(b) size of (str) output the value 9
(c) Both strlen (str) and size of (str) output the value 8.
(d) Both strlen (str) and size of (str) output the value 9.

Try Yourself

T1. Find the time complexity of following function $A(n)$

```
{   for (i = 1 to n)
    {   if (n mod i = = 0)
        {   for (j = 1 to n)
                printf(j);
        }
    }
}
```

[Ans: $O(n^2)$]

T2. What is the output of following program?

```
main( )
{   int i = 3;
    switch (i)
    {   default : printf("zero");
        Case 1 : printf("one");
                 break;
        Case 2 : printf("two");
                 break;
        Case 3 : printf("three");
                 break;
    }
}
```

[Ans: O(1)]

T3. Write the meaning of following C declarations
1. const int *p;
2. int *const p;

T4. Write the meaning of following C declarations
1. char (*(*x( ))[ ])( );
2. char (*(*x[3])( ))[5];
3. void (*b(int, void (*f) (int)))(int);
4. void (*ptr)(int (*)[2], int (*) (void));

T5.

```
int main( )
{   int i;
    char a[ ] = "\0";
    if(printf(("%s", a))
    printf("string empty");
    else
    printf("string is not empty");
    return 0;
}
```

Output of the above program is
(a) string empty
(b) string is not empty
(c) no output
(d) 0

[Ans: (b)]

T6. What will be the output of the following C program?

```
void count(int n)
{static int d = 1;
printf("%d", n);
printf("%d", d);
        d++;
if (n > 1) count (n – 1);
printf("%d", d);
}
```

```
void 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

[GATE-2016, Ans: (a)]

T7. What will be the output of the following pseudo-code when parameters are passed by reference and dynamic scoping is assumed?

```
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

[GATE-2016, Ans: (d)]

T8. The following function computes X^Y for positive integers X and Y.

```
int exp (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) $X^Y = a^b$
(b) $(res * a)^Y = (res * X)^b$
(c) $X^Y = res * a^b$
(d) $X^Y = (res * a)^b$

[GATE-2016, Ans: (c)]

T9. Consider the following program:

```
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};
    print f("%d", f(a, 5));
}
```

Note: max (x, y) returns the maximum of x and y.

The value printed by this program is _____.

[GATE-2016, Ans: (3)]

T10. The following C function takes two ASCII strings and determines whether one is an anagram of the other. An anagram of a string s is a string obtained by permuting the letters in s.

```
int anagram (char *a, char *b)
{
    int count [128], j;
    for (j = 0; j < 128; j++)
        count[j] = 0;
    j = 0;
    while (a[j] && b[j])
    {
```

```
        A;
        B;
    }
    for (j = 0; j < 128; j++)
        if (count [j]) return 0;
    return 1;
}
```

Choose the correct alternative for statements A and B.

(a) A: count [a[j]]++ and B: count [b[j]]– –
(b) A: count [a[j]]++ and B: count [b[j]]++
(c) A: count [a[j++]]++ and B: count [b[j]]– –
(d) A: count [a[j]]++ and B: count [b[j++]]– –

[GATE-2005, Ans: (d)]

T11. Consider the C program below. What does it print?

```
#include <stdio.h>
#define swapl (a, b) tmp = a; a = b; b =tmp;
void swap2 (int a, int b)
{
    int tmp;
    tmp = a; a = b; b = tmp;
}
void swap3 (int*a, int*b)
{
    int tmp;
    tmp = *a; *a = *b; *b = tmp;
}
int main( )
{
    int num1 = 5, num2 = 4, tmp;
    if(num1 > num2)
    { swap1(num1, num2); }
    if (num1 < num2)
    { swap2(num1 + 1, num2); }
    if(num1 > = num2)
    { swap3 (&num1, &num2); }
    printf ("%d, %d", num1, num2);
}
```

(a) 5, 5 (b) 5, 4
(c) 4, 5 (d) 4, 4

[GATE-2008, Ans: (c)]

T12. Consider the program below:

```
# include <stdio.h>
int fun (int n, int *f_p) {
    int t, f;
    if (n < = 1) {
      *f_p = 1;
      return 1;
    }
    t = fun(n – 1, f_p);
    f = t + * f_p;
    *f_p = t;
    return f;
}
int main( )
{
    int x = 15 ;
    printf ("% d\n", fun (5, &x));
    return 0;
}
```

The value printed is

(a) 6 (b) 8
(c) 14 (d) 15

[GATE-2009, Ans: (b)]

T13. Consider the following C program.

```
# 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( );
      pirntf("%d", x);
      return 0;
}
int f1( ) {int x = 25; x++; return x;}
int f2( ) {static int x = 50; x++; return x;}
int f3( ) {x *= 10; return x;}
```

The output of the program is _____.

[GATE-2015, Ans: (230)]

T14. Consider the following C program:

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

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

[GATE-2018, Ans: (a)]

T15. Consider the following C code. Assume that unsigned long int type length is 64 bits.

```
unsigned long int fun (unsigned long int n) {
    unsigned long int i, j = 0, sum = 0;
    for (i = n; i > 1; i = i/2) j++;
    for (; j > 1; j = j/2) sum++;
    return (sum);
}
```

The value returned when we call fun with the input 2^{40} is

(a) 4 (b) 5

(c) 6 (d) 40

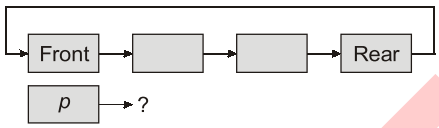
[GATE-2018, Ans: (b)]

■■■■

2 Linked List, Stack, Queue and Hashing

Multiple Choice Questions

Q.1 A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations EnQueue and DeQueue can be performed in constant time?



(a) rear node
(b) front node
(c) not possible with a single pointer
(d) node next to front

[GATE-2004]

Q.2 The following C function takes a singly-linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```
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

[GATE-2008]

Q.3 Let A be a two-dimensional array declared as follows:
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) $10j + i + 89$

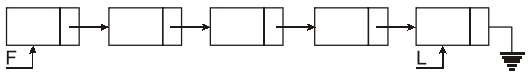
[GATE-1998]

Q.4 Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest?

(a) union only
(b) intersection, membership
(c) membership, cardinality
(d) union, intersection

[GATE-2004]

Q.5 Consider a singly linked list of the form as given in the figure below, where F is a pointer to the first element in the list and L is a pointer to the last element in the list. For which of the following operations, the corresponding time depends on the length of the list?



(a) Add an element after the last element of the list
(b) Delete the last element of the list
(c) Add an element before the first element
(d) Interchange the first two elements of the list

[JNUEE-2007]

Q.6 A data structure is comprised of nodes each of which has exactly two pointers to other nodes with no null pointers. The following C program is to be used to count the number of nodes accessible from a given node. It uses a mark field assumed to be initially zero for all nodes. There is a statement missing from this code.

```
struct test {int info, mark; struct test * p, *q}
int nodecount (struct test *a)
{
    if (a → mark) return 0;
    return nodecount (a → p) + nodecount
    (a → q) + 1;
}
```

Select from the following the change that should make the program work property

(a) Add a → mark = 1; as the first statement
(b) Add a → mark = 0; after the if-statement
(c) Add a → mark = 1; after the if-statement
(d) Add a → mark = 0; as the last statement

[JNUEE-2005]

Q.7 A doubly-linked list facilitates the determination of the predecessor of a given item. Which of the following operations utilizes this property of a doubly-linked list to its greatest advantage?

(a) accessing an item
(b) recovering a lost pointer
(c) copying a list
(d) merging two lists

[JNUEE-2004]

Q.8 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.

```
typedef struct node
{   int value;
    struct node *next;
}   Node;
Node *move_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;

[GATE-2010]

Q.9 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?

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

[GATE-1994]

Q.10 A priority queue Q is used to implement a stack that stores characters. PUSH (C) is implemented INSERT (Q, C, K) where K is an appropriate integer key chosen by the implementation. POP is implemented as DELETEMIN(Q). For a sequence of operations, the keys chosen are in

(a) non-increasing order
(b) non-decreasing order
(c) strictly increasing order
(d) strictly decreasing order

[GATE-1997]

Q.11 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.

```
int main( ) {
unsigned int 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

[GATE-2015]

Q.12 Consider the C code fragment given below.

```
typedef struct 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.

[GATE-2017]

Q.13 Which of the following is TRUE?

(a) The cost of searching an AVL tree is $\Theta(\log n)$ but that of a binary search tree is $O(n)$
(b) The cost of searching an AVL tree is $\Theta(\log n)$ but that of a complete binary tree is $\Theta(n \log n)$
(c) The cost of searching a binary search tree is $O(\log n)$ but that of an AVL tree is $\Theta(n)$
(d) The cost of searching an AVL tree is $\Theta(n \log n)$ but that of a binary search tree is $O(n)$

[GATE-2008]

Q.14 Let P be a singly 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?

(a) $O(n)$ (b) $O(\log^2 n)$
(c) $O(\log n)$ (d) $O(1)$

[GATE-2004]

Q.15 The concatenation of 2-lists is to be performed in 0(1) time. Which one of the following implements?

(a) Single linked list
(b) Doubly linked list
(c) Circular doubly linked list
(d) None of these

Q.16 Consider the following C program.

```
struct listnode
{   int data;
    struct listnode *next;
} ;
void fun (struct listnode *head)
{   if (head == NULL || head → next == NULL)
    return;
    struct listnode *tmp = head → next;
    head → next = tmp → next;
    free (tmp);
    fun (head → next);
}
```

What is the functionality of the above function?

(a) It reverses the linked list
(b) It deletes the linked list
(c) Alternate nodes will be deleted
(d) It reverses the linked list and delete alternate nodes

Q.17 Consider the following code.

```
int f(Struct Node *A, Struct Node *B)
{
    if (A == NULL && B == NULL) return 0;
    else if (A == NULL || B == NULL) return 1;
    else if (A → data != B → data) return 1;
    if (f(A → left, B → left) || f(A → right, B → right)) return 1;
    else return 0;
}
```

Assume Node is a structure type with 3 members as follows:

```
Struct Node
{   int data;
    Struct Node *left;
    Struct Node *right;
};
```

If two binary tree root pointers are passed to the function *f*( ), then which of the following statement is correct?

(a) It compares two given binary trees and returns 1 if two trees are different and it returns 0 otherwise.

(b) It compares two given binary trees and returns 0 if two trees are different and it returns 0 otherwise.

(c) It compares two given binary trees but return value can not be used to differentiate the trees.

(d) None of these

Q.18 Let $h(k) = k \bmod 7$. Calculate the number of collisions with linear probing for insertion of the following keys.

29 36 16 30

(a) 1 (b) 2

(c) 3 (d) 4

[GATE-2000]

Q.19 Which of the following is the best choice as m in the hash function, $h(k) = k \bmod m$?

(a) 61 (b) 701

(c) 81 (d) 1031

[GATE-2000]

Q.20 Consider the following function.

```
void madeeasy (int n)
{
    enqueue (Q, 0);
    enqueue (Q, 1);
    for (i = 0; i < n; i++)
    {
        x = dequeue (Q);
        y = dequeue (Q);
        enqueue (Q, y);
        enqueue (Q, x + y);
        print (x);
    }
}
```

What is the functionality of above function madeeasy?

(a) Prints numbers from 0 to $n - 1$

(b) Prints numbers from $n - 1$ to 0

(c) Prints first n Fibbonacci numbers

(d) Prints first n Fibbonacci numbers in reverse order

Q.21 Choosing the hash function randomly from a class of hash functions such that it is independent of the keys to be stored, is termed as:

(a) perfect hashing

(b) simple uniform hashing

(c) universal hashing

(d) none of the above **[GATE-2000]**

Q.22 Given the hash function $h(k, i) = (h'(k) + i + i^2) \bmod 11$ and $h'(k) = k \bmod 11$.

What is the number of collisions to store the following keys?

Following keys : 23 12 19 11 33 16 46 37

(a) 3 (b) 2

(c) 11 (d) None of these

[GATE-2000]

Common Data for Q.23 & Q.24

Q.23 Consider an open address hash table with a total of 10,000 slots, containing 9800 entries. What is the expected number of probes in a successful search?

(a) 2 (b) 3

(c) 4 (d) 4.5

[GATE-2000]

Q.24 In above problem, what is the expected number of probes in unsuccessful search?

(a) 4 (b) 10

(c) 20 (d) 50

[GATE-2000]

Q.25 The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function $h(k) = k \bmod 10$ and linear probing. What is the resultant hash table?

(a)

0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

(b)

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

(c)

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

(d)

0	
1	
2	12,2
3	13,3,23
4	
5	5,15
6	
7	
8	18
9	

[GATE-2009]

Q.26 Consider a hash table consisting of M = 11 slots (numbering of slots start from 0), and suppose integer key values are hashed into the table using the hash function $h1$:

```
int h1 (int key)
{   int x;
    x = (key + 5) * (key + 5);
    x = x/16;
    x = x + key;
    x = x % 11;
    return x;
}
```

Suppose that collisions are resolved using linear probing. The probe sequence is given therefore by $(h1(k) + i) \pmod{11}$ (i starts from 0). The integer key values listed below are to be inserted, in the order given. What are the final contents of the hash table after the following key values have been inserted in the given order:

43, 23, 1, 0, 15, 31, 4, 7, 11, 3

(a)

43	0	31	1		23	15	7	11	3	4

(b)

43	0	1	31		7	15	23	11	4	3

(c)

43	0	31	1		7	23	15	11	4	3

(d) None of the above

Linked Answer for Q.27 & Q.28

A has table of length 10 uses open addressing with hash function $h(k) = k \bmod 10$, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Q.27 Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

(a) 46, 42, 34, 52, 23, 33

(b) 34, 42, 23, 52, 33, 46

(c) 46, 34, 42, 23, 52, 33

(d) 42, 46, 33, 23, 34, 5

[GATE-2010]

Q.28 How many different insertion sequences of the key values using the same hash function and linear probing will result in the hash table shown above?

(a) 10 (b) 20

(c) 30 (d) 40

[GATE-2010]

Q.29 Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true?

1. 9679, 1989, 4199 hash to the same value
2. 1471, 6171 hash to the same value
3. All elements hash to the same value
4. Each element hashes to a different value

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

[GATE-2003]

Q.30 Consider the hash table of size 7, with starting index 0 and a has function $(3x + 4)$ mod 7. Assuming the hash table is initially empty, which of the following is the contents of the table, when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing (Linear probing)?

(a) 8, -, -, -, -,-, 10 (b) 1, 8, 10, -, -, -, 3
(c) 1, -, -, -, -, -, 3 (d) 1, 10, 8, -, -, -, 3

[GATE-2007]

Q.31 A one dimensional array A has indices 1... 75. Each element is a string and takes up three memory words. The array is stored at location 1120 decimal. The starting address of A[49] is

(a) 1267 (b) 1164
(c) 1264 (d) 1169

[ISRO-2009]

Q.32 What is the minimum number of stacks of size n required to implement a queue of size n?

(a) one (b) two
(c) three (d) four

[GATE-2001]

Q.33 Consider the following declaration of a two-dimensional array in C.

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

[GATE-2002]

Q.34 Suppose you are given an array $s[1...n]$ and a procedure reverse (s, i, j) which reverse the order of elements between positions i and j (both inclusive). What does the following sequence do, where $1 \le k \le n$.

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 **[GATE-2000]**

Q.35 Assume that there are two lower triangular matrices A and B of size $n \times n$. If matrix A and transpose of B are fit into a rectangular matrix C of size $n \times (n + 1)$, then

(a) $B[i, j] = C[i, j + 1]$ (b) $B[i, j] = C[j + 1, i]$
(c) $B[i, j] = C[j, i + 1]$ (d) None of these

[JNUEE-2009]

Q.36 Assume that a lower triangular matrix $A[0.. n - 1, 0.. n - 1]$ is stored in a linear array $B\left[0..\frac{1}{2}n(n+1)-1\right]$ in row-by-row order. For $n = 100$, if $A[0, 0]$ is stored in $B[0]$, where is $A[90, 80]$ stored in B array?

(a) 4175 (b) 0
(c) 4165 (d) 4160

[JNUEE-2009]

Q.37 To evaluate an expression without any embedded function calls

(a) one stack is enough
(b) two stacks are needed
(c) as many stacks as the height of the expression tree are needed
(d) a turning machine is needed in the general case

[GATE-2002]

Q.38 Let S be a stack of size $n \ge 1$. 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 operations take X seconds each, and Y seconds elapse

between the end of one such stack operation and the start of the next operation. For $m \geq 1$, 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]

Q.39 A single array A[1...MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top 1 and top 2 (top 1 < 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

(a) (top 1 = MAXSIZE/2) and (top 2 = MAXSIZE/2 + 1)
(b) top 1 + top 2 = MAXSIZE
(c) (top 1 = MAXSIZE/2) or (top 2 = MAXSIZE)
(d) top 1 = top 2 – 1

[GATE-2004]

Q.40 To remove recursion from a program we have to use the following data structures

(a) Array (b) Stack
(c) Queue (d) List

[DRDO-2008]

Q.41 The expression 1 * 2 ^ 3 * 4 ^ 5 * 6 will be evaluated as

(a) 32^{30} (b) 162^{30}
(c) 49152 (d) 173458

[ISRO-2009]

Q.42 What is the maximum size of the operator stack during the conversion of the infix expression A + B*C – D/E to postfix?

(a) 1 (b) 2
(c) 3 (d) 4

[DRDO-2008]

Q.43 What is the maximum size of the operand stack while evaluating the postfix expression 6 2 3 + –3 8 2/+ *?

(a) 1 (b) 2
(c) 3 (d) 4

[DRDO-2008]

Q.44 Assume that the operators x, –, + are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, x, + and –. The postfix expression corresponding to the infix expression $a + b \times c - d \wedge e \wedge f$ is

(a) $abc \times + def \wedge \wedge -$
(b) $abc \times + de \wedge f \wedge$
(c) $ab + c \times d - e \wedge f \wedge$
(d) $- + a \times bc \wedge \wedge def$

[ISRO-2009]

Q.45 The best data structure to check whether an arithmetic expression has balanced parentheses is a

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

[GATE-2004]

Q.46 A circular queue has been implemented using a singly linked list where each node consists of a value and a single pointer pointing to the next node. We maintain exactly two external pointers **FRONT** and **REAR** pointing to the front node and the rear node of the queue, respectively. Which of the following statements is/are CORRECT for such a circular queue, so that insertion and deletion operations can be performed in $O(1)$ time?

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

[GATE-2017]

Q.47 Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:

(i) isEmpty (Q) : returns true if the queue is empty, false otherwise.
(ii) delete (Q) : deletes the element 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)
{
    int i;
    if (! is Empty (Q))
    {
        i = delete (Q);
        f(Q);
        insert (Q, i);
    }
}
```

What operation is performed by the above function f?

(a) Leaves the queue Q unchanged
(b) Reverses the order of the elements in the queue Q
(c) Deletes the element at the front of the queue Q and inserts it at the rear keeping the other elements in the same order
(d) Empties the queue Q

[GATE-2007]

Q.48 Suppose a circular queue of capacity (n - 1) elements is implemented with an array of n elements. Assume that the insertion and deletion operations are carried out using REAR and FRONT as array indeed variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are

(a) full : (REAR + 1) mod n == FRONT
empty : REAR == FRONT
(b) full : (REAR + 1) mod n == FRONT
empty : (FRONT + 1) mod n == REAR
(c) full : REAR == FRONT
empty : (REAR + 1) mod n == FRONT
(d) full : (FRONT + 1) mod n == REAR
empty : REAR == FRONT

[GATE-2012]

Q.49 Consider the following operation along with Enqueue and Dequeue operations on queues, where k is a global parameter.

```
Multi-Dequeue (Q) {
    m = k;
    while (Q is not empty) and (m > 0) {
        Dequeue (Q);
        m = m – 1;
            }
}
```

What is the worst case time complexity of a sequence of n queue operations on an initially empty queue?

(a) $\Theta(n)$ (b) $\Theta(n + k)$
(c) $\Theta(nk)$ (d) $\Theta(n^2)$

[GATE-2013]

Q.50 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?

(a) A queue cannot be implemented using this stack.
(b) A queue can be implemented where ENQUEUE takes a single instruction and DEQUEUE takes a 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 instruction each.

[GATE-2014]

Numerical Answer Type Questions

Q.51 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.

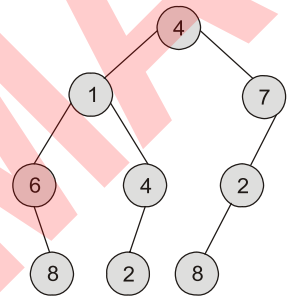
while *Q is not Empty* do
if *S* is *Empty OR Top* (*S*) ≤ Head(*Q*) then
x : = Dequeue (*Q*);
Push (*S*, x);
else
x : = Pop (*S*);
Enqueue (*Q*, x);
end
end
The maximum possible number of iterations of the **while** loop in the algorithm is _____.

[GATE-2016]

Q.52 Consider the following C program

```
struct treenode
{   int data;
    struct treenode *left;
    struct treenode *right;
}
int gate2015 (struct treenode *root)
{   int gate;
    if (root ==NULL) return 0;
    if (root → data % 2==1)
    {   root → data –=1;
        gate ++;
    }
    else
        root → data +=1;
return (root → data + gate2015 (root → left) +
gate2015 (root → right);
}
```

If the above code is executed on the following tree then what is the output?



Q.53 Consider the tree T in which left subtree contains half of the maximum number of nodes possible in the AVL tree of height 6 and right subtree consists of one 3rd of the minimum number of nodes possible in AVL tree of height '6'. Then what will be the total number of nodes in T.

Q.54 Suppose $c = \langle c[0], \ldots, c[k-1] \rangle$ is an array of length k, where all the entries are from the set {0, 1}. For any positive integers a and n, consider the following pseudocode.

```
DOSOMETHING (c, a, n)
z ← 1
for i ← 0 to k – 1
      do z ← z^2 mod n
      if c[i] = 1
            then z ← (z × a) mod n
return z
```

If $k = 4$, $c = \langle 1, 0, 1, 1 \rangle$, $a = 2$ and $n = 8$, then the output of DOSOMETHING (*c*, *a*, *n*) is _____.

[GATE-2015]

Common Data for Q.55 & Q.56

Consider the following keys that are hashed into the hash table in the order given using the hash function $h(i) = (2i + 5) \bmod 11$.

12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5.

Assume hash table has locations from 0 to 10.

Q.55 Find the location of an element 39, if hash table uses linear probing to hash the given elements.

Q.56 If hash table uses chaining to handle the collisions, how many locations are left without hashing any element into it?

Multiple Select Questions

Q.57 Consider the postfix expression given

25 * 6 + 43 * –

Which of the following statements are TRUE?

(a) Maximum stack size used is 3.
(b) Maximum stack size used is 4.
(c) To elevate the above expression we need operand stack.
(d) Answer of the above postfix evaluation is 4.

Q.58 Consider the below statements about linked list and array?

(a) Random access is NOT allowed in the linked list.

(b) Insertion and deletion of elements is easy in linked list than array.

(c) Elements are stored in continuous memory locations in the linked list.

(d) Inserting an element in specific position is easy than an array.

Q.59 Which of the following is TRUE regarding the implementation of queue using 2 stacks?

(a) If inserting an element takes $O(1)$ time, then deletion must take $O(n)$ time.

(b) If inserting an element takes $O(n)$ time, then deletion must take $O(1)$ time.

(c) Both insertion and deletion will take $O(1)$ time.

(d) Both insertion and deletion will take $O(n)$ time.

Q.60 You are entrusted with the task of deleting a node in a singly linked list, whose data field is x. Note that the node which is to be deleted can be at any arbitrary position in the linked list.

Consider the following scenarios:

S_1 : You are only provided with a pointer to the node which is to be deleted in the linked list.

S_2 : You are only provided with a pointer to the starting node of the linked list.

Which of the following options is incorrect?

(a) In both the scenario's, deletion is possible for all inputs, and deletion will be more efficient in S_1 than S_2.

(b) In both the scenario's, deletion is possible for all inputs and deletion will be more efficient in S_2 than S_1.

(c) In S_1, deletion is NOT possible in certain cases, but in S_2 deletion is possible for all inputs.

(d) In S_2, deletion is NOT possible in certain cases, but in S_1, deletion is possible for all cases.

Try Yourself

T1. If we implement stack using single linked list how much time push and pop operations will take?

[Ans: Push-O(1), Pop-O(1)]

T2. If we implement queue using single linked list how much time enqueue and dequeue operations will take?

[Ans: enqueue-O(1), dequeue-O(n)]

T3. Consider the implementation of multiple stacks in single array S of size P from index 0 to $P-1$. Number of stacks are Q. The following function PUSH ( ), used to push data x on to a particular stack i where T_i is used as top variable for stack i (i indicates stack number).

```
PUSH (S, P, Q, T_i, x)
{
    if (_____A_____)
    {
        printf ("stack overflow");
        exit (1);
    }
    else
        T_i++;
    S[T_i] = x;
}
```

Stack 0 stores elements from 0 to $Q-1$, stack 1 stores from q to $2Q-1$, and similarly other stack will store elements. Which of the following is the correct expression to replace A in the above function?

(a) $T_i == \left(\frac{P}{Q} \times i - 1\right)$

(b) $T_i == \left(\frac{P}{Q} \times i + 1\right)$

(c) $T_i == \left(\frac{P}{Q} \times (i-1) - 1\right)$

(d) $T_i == \left(\frac{P}{Q} \times (i+1) - 1\right)$

[Ans: (d)]

T4. An implementation of a queue Q, using two stacks S1 and S2, is given below

```
void insert (Q, x)
{
    push (S1, x);
}
void delete (Q, x)
{   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 ($\leq n$) 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 processes. Which one of the following is true for all m and n?

(a) $n + m \leq x \leq 2n$ and $2m \leq y \leq n + m$

(b) $n + m \leq x < 2n$ and $2m \leq y \leq 2n$

(c) $2m \leq x < 2n$ and $2m \leq y \leq n + m$

(d) $2m \leq x < 2n$ and $2m \leq y \leq 2n$

[Ans: (a)]

T5. Consider 3 dimensional Array A[90] [30] [40] stored in linear array in column major order. If the base address starts at 10, what is the location of A [20] [20] [30]?
Assume the first element is stored at A[1][1][1].

[Ans: (22079)]

T6. Consider the following infix expression which is to be converted to postfix expression using stack.

(((P + Q) * (R + S)) / T) + (A * (B + C))

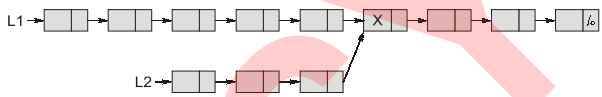
What is the maximum size of stack?

[Ans: (5)]

T7. Consider the hashing table with 'm' slots and 'n' keys. If the expected number of probes in unsuccessful search is 3, then find the expected number of probes in a successful search.

[Ans: (0.7324)]

T8. Consider the two linked list L1 and L2 given below each of length m and n



What will be the worst case time complexity to find address of node 'X', where both linked list will intersect with each other?

(a) $O(m \log n)$ (b) $O(m + n)$

(c) $O(n \log m)$ (d) $O(n^2 \log m)$

[Ans: (b)]

T9. In an adjacency list representation of an undirected simple graph $G = (V, E)$, each edge (u, v) has two adjacency list entries: $[v]$ in the adjacency list of u, and $[u]$ in the adjacency list of v. These are called twins of each other. A twin pointer is a pointer from an adjacency list entry to its twin. If $|E| = m$ and $|V| = n$, and the memory size is not a constraint, what is the time complexity of the most efficient algorithm to set the twin pointer in each entry in each adjacency list?

(a) $\Theta(n^2)$ (b) $\Theta(n + m)$

(c) $\Theta(m^2)$ (d) $\Theta(n^4)$

[GATE-2016, Ans: (b)]

■■■■

3 Trees

Multiple Choice Questions

Q.1 Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?

(a) 7 5 1 0 3 2 4 6 8 9
(b) 0 2 4 3 1 6 5 9 8 7
(c) 0 1 2 3 4 5 6 7 8 9
(d) 9 8 6 4 2 3 0 1 5 7

[GATE-2003]

Q.2 Among the following statements, identify the false statement.

(a) We must balance a left-of-left unbalanced AVL tree by rotating the out-of-balance node to the right.
(b) The inorder traversal of a binary search tree produces an ordered list.
(c) When a module calls a subroutine recursively, in each call, all of the information is popped in the same order when subroutines are terminated one after another and finally the control is returned to the calling module.
(d) A recursion algorithm has two elements: Each call either solves only part of the problem or it reduces the size of the problem.

[JNUEE-2003]

Q.3 Postorder 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

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

(a) 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95
(b) 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29
(c) 29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95
(d) 95, 50, 60, 40, 27, 23, 22, 25, 10, 0, 15, 29

[GATE-2004]

Q.4 Suppose a binary tree has only three nodes A, B, and C and you are given that the post-order traversal for the tree is B-A-C. The exact pre-order traversal for the tree is.

(a) C-A-B
(b) A-B-C
(c) C-B-A
(d) a definite pre-order traversal cannot be determined from the information given

Q.5 Level order traversal of a rooted tree can be done by starting from the root and performing

(a) preorder traversal
(b) inorder traversal
(c) depth first search
(d) breadth first search

[GATE-2004]

Q.6 Consider the label sequences obtained by the following pairs of traversals on a labeled binary tree.

Which of these pairs identify a tree uniquely?

1. Preorder and postorder
2. Inorder and postorder
3. Preorder and inorder
4. Level order and postorder

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

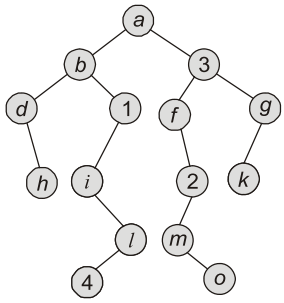
[GATE-2004]

Q.7 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 postorder traversal of the binary tree is

(a) *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*

[GATE-2007]

Q.8 Consider the following tree:



If the inorder traversal of the above tree is *dhbinleafmojckg* then fill the numbers 1, 2, 3 and 4 in the above tree.

(a) *n*, *e*, *j* and *c* (b) *e*, *j*, *c* and *n*
(c) *j*, *c*, *n* and *e* (d) All of these

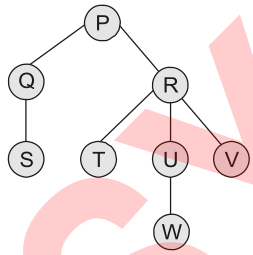
Q.9 A binary search tree contains the values 11, 22, 33, 44, 55, 66, 77 and 88. Which of the following is a valid preorder traversal?

(a) 55, 33, 11, 22, 44, 77, 66, 88
(b) 55, 33, 22, 44, 11, 66, 77, 88
(c) 55, 33, 11, 22, 66, 44, 88, 77
(d) 55, 33, 11, 44, 22, 66, 77, 88

Q.10 In delete operation of binary search tree, we need inorder successor (or predecessor) of a node when a node to be deleted where it has both left and right child. Which of the following is true about inorder successor needed in delete operation?

(a) Inorder successor is always either leaf node or a node with empty right child.
(b) Inorder successor maybe a an ancestor of the node.
(c) Inorder successor is always a leaf node.
(d) Inorder successor is always either a leaf node or a node with empty left child.

Q.11 Consider the following rooted tree with the vertex labeled P as the root:



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

(a) SQPTRWUV (b) SQPTUWRV
(c) SQPTWUVR (d) SQPTRUWV

[GATE-2014]

Q.12 Consider the pseudocode given below. The function **DoSomething()** takes as argument a pointer to the root of an arbitrary tree represented by the *leftMostChild-rightSibling* representation. Each node of the tree is of type **treeNode**.

```
typedef struct treeNode* treeptr;
struct treeNode
{
    treeptr leftMostChild, rightSibling;
};
int DoSomething (treeptr tree)
{
    int value=0;
    if (tree != NULL)
    {
        if (tree->leftMostChild == NULL)
            value = 1;
        else
            value = DoSomething(tree->
                                leftMostChild);
        value=value+DoSomething(tree->
                                rightSibling);
    }
    return(value);
}
```

When the pointer to the root of a tree is passed as the argument to **DoSomething**, the value returned by the function corresponds to the
(a) number of internal nodes in the tree.
(b) height of the tree.
(c) number of nodes without a right sibling in the tree.
(d) number of leaf nodes in the tree.

[GATE-2014]

Q.13 Let LASTPOST, LASTIN and LASTPRE denote the last vertex visited in a postorder, inorder an preorder traversal. Respectively, of a complete binary tree. Which of the following is always tree?
(a) LASTIN = LASTPOST
(b) LASTIN = LASTPRE
(c) LASTPRE = LASTPOST
(d) None of the above

[GATE-2000]

Q.14 The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree (the height is the maximum distance of a leaf node from the root)?
(a) 2 (b) 3
(c) 4 (d) 6

[ISRO-2009]

Q.15 A binary tree can be uniquely reconstructed from the following traversal(s):
(a) Preorder
(b) Postorder
(c) Preorder and Postorder
(d) Inorder and Preorder

[DRDO-2008]

Q.16 Insert keys 4, 12, 8, 16, 6, 18, 24, 7 into an initially empty binary search tree. Delete the node having the key 6. The preorder traversal after deletion is
(a) 4, 12, 7, 8, 24, 18, 16
(b) 4, 12, 8, 7, 16, 18, 24
(c) 4, 12, 8, 7, 24, 18, 16
(b) 4, 12, 7, 8, 16, 18, 24

[DRDO-2009]

Directions for Question 17 to 19:

A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of keys.

I. 81, 537, 102, 439, 285, 376, 305
II. 52, 97, 121, 195, 242, 381, 472
III. 142, 248, 520, 386, 345, 270, 307
IV. 550, 149, 507, 395, 463, 402, 270

Q.17 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?
(a) II and III only (b) I and III only
(c) III and IV only (d) III only

[GATE-2008]

Q.18 Which of the following statements is TRUE?
(a) I, II and IV are inorder sequences of three different BSTs
(b) I is a preorder sequence of some BST with 439 as the root
(c) II is an inorder sequence of some BST where 121 is the root and 52 is a leaf
(d) IV is a postorder sequence of some BST with 149 as the root

[GATE-2008]

Q.19 How many distinct BSTs can be constructed with 3 distinct keys?
(a) 4 (b) 5
(c) 6 (d) 9

[GATE-2008]

Q.20 Consider the label sequences obtained by the following pairs of traversals on a labeled binary tree. Which of these pairs identify a tree uniquely?
1. Preorder and postorder
2. Inorder and postorder
3. Preorder and inorder
4. Level order and postorder
(a) 1 only (b) 2 and 3
(c) 3 only (d) 4 only

[GATE-2004]

Q.21 You are given the postorder traversal, P of a binary search tree on the n elements 1, 2, ... , *n*. You have to determine the unique binary search tree that has P as its postorder traversal. What is the time complexity of the most efficient algorithm for doing this?

(a) $\Theta(\log n)$
(b) $\Theta(n)$
(c) $\Theta(n \log n)$
(d) None of the above, as the tree cannot be uniquely determined.

[GATE-2008]

Q.22 Let *T* be a binary search tree with 15 nodes. The minimum and maximum possible heights of *T* are:

Note: The height of a tree with a single node is 0.

(a) 4 and 15 respectively
(b) 3 and 14 respectively
(c) 4 and 14 respectively
(d) 3 and 15 respectively

[GATE-2017]

Numerical Answer Type Questions

Q.23 What is the maximum possible height of an AVL tree with 20 nodes?

[DRDO-2008]

Q.24 A binary search tree was constructed by inserting following elements in to an initially empty binary tree.

50, 27, 16, 88, 34, 65, 52, 77, 93, 4, 12, 29, 44, 92

Preorder and postorder traversals of the resultant binary search tree were stored in arrays A and B respectively. How many elements have same index location in both the arrays? [Assume arrays A and B start from the same index]

Q.25 Consider the following elements which are inserted into initially empty AVL tree in the given order.

11, 7, 12, 13, 2, 9, 1, 3, 4

What is the last element in preorder traversal of left subtree of resultant AVL tree?

Q.26 The key 14, 4, 6, 16, 32, 50 in the order are inserted into an initially empty AVL tree. Find total number of rotations to make AVL with the given keys. Assume "single rotation = 1 rotation" and "double rotation = 1 rotation".

Q.27 Assume the preorder traversal of binary tree is "*abc*". How many total different binary trees are possible whose postorder traversal is "*cba*" with the given preorder traversal?

Multiple Select Questions

Q.28 In delete operation of a binary search tree, we need inorder predecessor or successor of a node to be deleted where it has both left and right child. Which of the following is FALSE about inorder predecessor in delete operation?

(a) Inorder predecessor is always either a leaf node or a node with empty right child.
(b) Inorder predecessor is always either a leaf node or a node with empty left child.
(c) Inorder predecessor is always a leaf node.
(d) Inorder predecessor is always an ancestor of the leaf node.

Q.29 Which of the following is TRUE with respect to BST and AVL Tree's.

(a) Worst case time complexity to search an element in BST is $O(\log n)$.
(b) Worst case time complexity to search an element in AVL Tree is $O(\log n)$.
(c) AVL tree is always height balanced but BST may or may not.
(d) In case of left skewed or right skewed BST, the search time complexity is $O(n)$.

■■■■