Index- Programming and Data Structures

Data types and Size

1) The smallest integer than can be represented by an 8-bit number in 2's complement form is

1 Marks GATE-CSE/IT-2013( )

[A] -256 [B] -128

[C] -127 [D] 0

2) 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

2 Marks GATE-CSE/IT-2002( )

[A] 4040 [B] 4050

[C] 5040 [D] 5050

3) Randomized quicksort is an extension of quicksort where the pivot is chosen randomly. What is the worst case complexity of sorting n numbers using randomized quicksort?

1 Marks GATE-CSE/IT-2001( )

[A] O(n) [B] O(n log n)

[C] O(n^2) [D] O(n!)

4) Consider any array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the array ($i \leq n$), the index of the parent is

1 Marks GATE-CSE/IT-2001( )

[A] i-1 [B] [i / 2]

[C] [i / 2] [D] (i+1) /2

5) 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 locations 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]?

2 Marks GATE-CSE/IT-1998( )

[A] 15i + j + 84 [B] 15j + i + 84

[C] 10i + j + 89 [D] 10j + i + 89

Key Paper

1.	**B**	**2.**	**B**	**3.**	**B**	**4.**	**B**	**5.**	**B**

Functions & Parameters

Common Data for Q1 and Q2 is given below

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 (513, 2)?

2 Marks GATE-CSE/IT-2011( )

[A] 9 [B] 8

[C] 5 [D] 2

2) What is the return value of the function foo when it is called as foo (345, 10)?

2 Marks GATE-CSE/IT-2011( )

[A] 345 [B] 12

[C] 5 [D] 3

Common Data for Q3 and Q4 is given below

Consider the following C code segment:

```
int a, b, c = 0;
void prtFun(void);
main( )
{
static int a = 1;          /* Line 1 */
prtFun( );
a += 1;
prtFun( )
printf("\n od od ", a, b);
}
Void prtFun(void)
{
static int a=2;        /* Line 2 */
int b=1;
a+=++b;
printf("\n od od ", a, b);
}
```

3) What output will be generated by the given code segment if:

Line 1 is replaced by auto int a = 1;

Line 2 is replaced by register int a = 2;

2 Marks GATE-CSE/IT-2012,GATE-CSE/IT-2012( )

[A] 3 1
4 1
4 2

[B] 4 2
6 1
6 1

[C] 4 2
6 2
2 0

[D] 4 2
4 2
2 0

4) What output will be generated by the given code segment?

2 Marks GATE-CSE/IT-2012( )

[A] 3 1
4 1
4 2

[B] 4 2
6 1
6 1

[C] 4 2
6 2
2 0

[D] 3 1
5 2
5 2

Common Data for Q6 and Q5 is given below

Functions & Parameters

The following program fragment is written in a programming language that allows variables and dows not allow nested declarations offunctions.

```
Global int I = 100, j=5;
Void P(x) {
        int I = 10;
        print(x + 10);
        i= 200;
        j=20;
        print (x);
}
Main ( ) { P(i+j);}
```

5) If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are

2 Marks GATE-CSE/IT-2003( )

[A]115, 220 [B]25, 220
[C]25,15 [D]115,105

6) If the programming language uses dynamic scoping and call by name parameter passing mechanism, the values printed by the above program are

2 Marks GATE-CSE/IT-2003( )

[A]115, 220 [B]25,200
[C]25,15 [D]115, 105

7) Consider the following class definitions in a hypothetical object oriented language that supports inheritance and uses dynamic binding. The language should not be assumed to be either Java or C ++, thogh the syntax is similar

```
Class P {                    Class Q subclass of P {
        Void f(int i) {              void f(int i) {
                Print(i) ;                   print (2*i);
        }                            }
}                                  }
```

Now consider the following program fragment :

```
P x = new Q ( ) ;
Q y = new Q ( ) ;
P z = new Q ( );
x.f(1); ((P)y).f(1);z.f*(1);
```

Here ((P)y) denotes a typecast of y to P. The output produced by executing the above program fragment will be

2 Marks GATE-CSE/IT-2003( )

[A]1 2 1 [B]2 1 1
[C]2 1 2 [D]2 2 2

8) Which of the following are true ?

(i) A programming language which does not permit global variables of any kind and has no nesting of procedures / functions, but permits recursion can be implemented with static storage allocation

(ii) Multi-level access link (or display arrangement is needed to arrange activation records only if the programming language being implemented has nesting of procedures/function

(iii) Recursion in programming languages cannot be implemented with dynamic storage allocation

(iv) Nesting of procedures/functions and recursion require a dynamic heap allocation scheme for activation records

(v) Programming languages which permit a function to return a function as its result cannot be implemented with a stack-based storage allocation scheme for activation records

2 Marks GATE-CSE/IT-2008( )

[A] (ii) and (v) only
[B] (i), (iii) and (iv) only
[C] (i), (ii) and (v)
[D] (ii), (iii) and (v) only

9) What value would the following function return for the input x = 95?

```
Function fun (x:integer):integer;
Begin
If x >100 then fun : x - 10
Else fun : fun(fun (x + 11))
End;
```

2 Marks GATE-CSE/IT-1998( )

[A] 89
[B] 90
[C] 91
[D] 92

10) 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};
printf ("%d", f(a, 6));
return 0;
}
```

2 Marks GATE-CSE/IT-2010( )

[A] -9
[B] 5
[C] 15
[D] 19

11) Which is the most appropriate match for the items in the first column with the items in the second column

X. Indirect Addressing	I. Array implementation
Y. Indexed Addressing	II. Writing re-locatable code
Z. Base Register Addressing	III. Passing array as parameter

2 Marks GATE-CSE/IT-2001( )

[A] (X, III) (Y, I) (Z, II)
[B] (X, II) (Y, III) (Z, I)
[C] (X, III) (Y, II) (Z, I)
[D] (X, I) (Y, III) (Z, II)

12) A data structure is required for storing a set of integers such that each of the following operations can be done in (log n) time, where n is the number of elements in the set.
1. Delection of the smallest element
2. Insertion of an element if it is not already present in the set.
Which of the following data structures can be used for this purpose ?

2 Marks GATE-CSE/IT-2003( )

[A] A heap can be used but not a balanced binary search tree
[B] A balanced binary search tree can be used but not a heap
[C] Both balanced binary search tree and heap can be used
[D] neither balanced binary search tree nor heap can be used.

13) Consider the following C-program

```
void foo (int n, int sum 0 ) {
        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);
pritf("%d/n", sum);
        }
```

What does the above program print ?

1 Marks GATE-CSE/IT-2005( )

[A] 8,4,0,2,14

[B] 8,4,0,2,0

[C] 2,0,4,8,14

[D] 2,0,4,8,0

14) Consider the following C-program

```
double foo (double a); /*Line 1*/
 int main () {
 double da, db;
 //input da
 db =foo (da);
 }
double foo (double) {
   returna;
 }
```

The above code compliedwithout any error or warning. If Line 1 is deleted,The above code will show.

1 Marks GATE-CSE/IT-2005( )

[A] no compile warning or error

[B] some complier-warning not leading to unitended results

[C] Some complier-warning due to type-mismatch eventually leading to unitended results

[D] Complier errors

15) Choose the correct option to fill ?1 and ?2 so that the program below prints an input string in reverse order. Assume that the input string is terminated by a newline character.

```
void reverse (void) {
  int c;
 if (?1) reverse ();
 ?2
}
main ( ) {
  print f("Enter Text'); printf ("\n");
  reverse( ); printf("\n");
}
```

1 Marks GATE-CSE/IT-2008( )

[A] ? 1 is (getchar ( ) ! = '\ n')
? 2 is getchar (c) ;

[B] ? 1 is (c= getchar ( )) ! = '\n')
? 2 is getchar (c)

[C] ? 1 is (c ! = ' \n ')
? 2 is putchar (c) ;

[D] ? 1 is ((c= getchar ( )) ! = ' \n')
? 2 is putchar (c) ;

16) Consider the program below:

```
#include
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 :

1 Marks GATE-CSE/IT-2009( )

[A] 6 [B] 8

[C] 14 [D] 15

17) 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.

```
int f (int &x, int c) {
c = c − 1;
if (c == 0) return 1;
x = x + 1;
return f (x, c) * x;
}
```

2 Marks GATE-CSE/IT-2013( )

[A] 3024 [B] 6561

[C] 55440 [D] 161051

18) Consider the following function

```
int unknown (int n) {
int i, j, k = 0;
for (i = n / 2; i <= n; i + + )
for ( j = 2; j <= n; j = j * 2)
k = k + n / 2;
 return (k );
}
```

2 Marks GATE-CSE/IT-2013( )

[A] $\Theta(n^2)$ [B] $\Theta(n^2 log\ n)$

[C] $\Theta(n^3)$ [D] $\Theta(n^3 logn)$

Functions & Parameters

Key Paper

1.	**D**	**2.**	**B**	**3.**	**D**	**4.**	**C**	**5.**	**D**
6.	**B**	**7.**	**B**	**8.**	**B**	**9.**	**C**	**10.**	**C**
11.	**A**	**12.**	**B**	**13.**	**D**	**14.**	**D**	**15.**	**D**
16.	**B**	**17.**	**B**	**18.**	**B**				

Data Types

1) 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= - 1;
   Print (x);
}
 main (void) {
        x =5;
        p(&x);
        print (x);
}
```

The output of this program is

2 Marks GATE-CSE/IT-2003( )

[A] 12 7 6 [B] 22 12 11

[C] 14 6 6 [D] 7 6 6

2) 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

1 Marks GATE-CSE/IT-2004( )

[A] (top 1 = MAXSIZE /2) and (top 2 = MAXSIZE/2 + 1) [B] top 1 + top 2 = MAXSIZE

[C] top 1 = MAXSIZE/2) or (top2 = MAXSIZE [D] top =1 top 2 - 1

3) A variant record in Pascal is defined by

```
type varirec        =    record
number : integer;
case (var1,var2) of
var1: (x,y : integer);
var2: (p.q.: real)
end
end
```

Suppose an array of 100 records was declared on a machine which uses 4 bytes for an integer and 8 bytes for a real. How much space would the compiler have to reserve for the array?

1 Marks GATE-CSE/IT-1995( )

[A] 2800 [B] 2400

[C] 2000 [D] 1200

Data Types

Key Paper

1.	**A**	**2.**	**D**	**3.**	**C**

Stacks, Queues

1) Let S be a stack of size $n \geq 1$. 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 $m \geq 1$, define the stack-life of m cs the time elapsed from the end or 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

2 Marks GATE-CSE/IT-2003( )

[A] n(X + Y) [B] 3Y + 2X

[C]n(X + Y) -X [D]Y + 2X

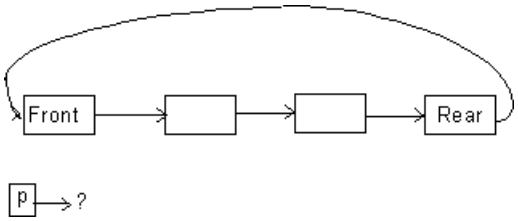
2) The best data structure to check whether an arithmetic expression has balanced parentheses is a

1 Marks GATE-CSE/IT-2004( )

[A] queue [B] stack

[C]tree [D]list

3) 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 ?



1 Marks GATE-CSE/IT-2004( )

[A] rear node [B]front node

[C]not possible with a single pointer [D]node next to front

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

1 Marks GATE-CSE/IT-2004( )

[A] abc x + def ^f ^_ [B] abc x + de ^ f ^

[C]ab + c x d -e ^ f ^ [D]- + a x b c ^ ^ def

5) 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(≤ 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?

2 Marks GATE-CSE/IT-2006( )

[A] n + m ≤ x< 2n and 2m ≤ 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

6) The number of elements that can be sorted in Θ (logn) time using heap sort is

2 Marks GATE-CSE/IT-2013( )

[A] $\Theta(1)$ [B] $\Theta(\sqrt{\log n})$

[C] $\Theta(\frac{\log n}{\log \log n})$ [D] $\Theta(\log n)$

7) Which of the following scheduling algorithms is non-preemptive?

1 Marks GATE-CSE/IT-2002( )

[A] Round Robin [B] First-In First-Out

[C] Multilevel Queue Scheduling [D] Multilevel Queue Scheduling with Feedback

8) To evaluate an expression without any embedded function calls

2 Marks GATE-CSE/IT-2002( )

[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

9) What is the minimum number of stacks of size n required to implement a queue of size n?

2 Marks GATE-CSE/IT-2001( )

[A] One [B] Two

[C] Three [D] Four

10) Consider the following C code segment.

```
for (i=0 i
        for(J=0; J< N; J ++)
                if (i%2)
                        { x + = (4* j+5 x i);
                        Y + = ( 7 + 4 * j);
                        }
                }
}
```

Which one of the following is false?

2 Marks GATE-CSE/IT-2006( )

[A] The code contains loop-in variant computation [B] There is scope of common sub-expression elimination in this code

[C] There is scope strength reduction in this code [D] There is scope of dead code elimination in this code.

11) Consider the following C code segment.

```
for (i=0 i
        for(J=0; J< N; J ++)
                if (i%2)
                        { x + = (4* j+5 x i);
                        Y + = ( 7 + 4 * j);
                        }
                }
}
```

Which one of the following is false?

2 Marks GATE-CSE/IT-2006( )

[A] the code contains loop-in variant computation [B] there is scope of common sub-expression elimination in this code

[C] There is scope strength reduction in this code [D] There is scope of dead code elimination in this code.

12) Which of the following is essential for converting an infix expression to the postfix form efficiently?

1 Marks GATE-CSE/IT-1997( )

[A] An operator stack [B] An operand stack

[C] An operand stack and an operator stack [D] A parse tree

13) Consider the following statements:
(i) First-in-first out types of computations are efficiently supported by STACKS.
(ii) Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.
(iii) Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
(iv) Last-in-first-out type of computations are efficiently supported by QUEUES.

2 Marks GATE-CSE/IT-1996( )

[A] (ii) and (iii) are true
[B] (i) and (ii) are true
[C] (iii) and (iv) are true
[D] (ii) and (iv) are true

14) Which of the following statements is true?

1 Marks GATE-CSE/IT-1995( )

[A] ROM is a Read/Write memory
[B] PC points to the last instruction that was executed
[C] Stack works on the principle of LIFO
[D] All instructions affect the flags

15) The postfix expression for the infix expression _A _+ _B _* (_C _+ _D)_/ _F _+ _D _* _E _i_s

2 Marks GATE-CSE/IT-1995( )

[A] AB+CD+*F/D+E*
[B] ABCD+*F/DE*++
[C] A*B+CD/F*DE++
[D] A+*BCD/F* DE++

16) 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?

1 Marks GATE-CSE/IT-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

17) The following sequence of operations is performed on a stack:
PUSH (10), PUSH (20), POP, PUSH (10), PUSH (20), POP, POP, POP,PUSH (20), POP
The sequence of values popped out is:

1 Marks GATE-CSE/IT-1991( )

[A] 20,10,20,10,20
[B] 20,20,10,10,20
[C] 10,20,20,10,20
[D] 20,20,10,20,10

Statement for Linked answer Q18 and Q19 is given below

18) A hash table of length 10 uses open addressing with hash function h(k)=k mod 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.
Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

2 Marks GATE-CSE/IT-2010,GATE-CSE/IT-2010( )

[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, 52

19) A hash table of length 10 uses open addressing with hash function h(k)=k mod 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: 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?

2 Marks GATE-CSE/IT-2010( )

[A] 10
[B] 20
[C] 30
[D] 40

Stacks, Queues

Key Paper

1.	**D**	**2.**	**B**	**3.**	**C**	**4.**	**A**	**5.**	**A**
6.	**C**	**7.**	**B**	**8.**	**A**	**9.**	**B**	**10.**	**D**
11.	**D**	**12.**	**A**	**13.**	**A**	**14.**	**C**	**15.**	**B**
16.	**B**	**17.**	**B**	**18.**	**C**	**19.**	**C**		

Linked Lists

1) 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;
  }
}
```

1 Marks GATE-CSE/IT-2008( )

[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

2)

The following C function takes a simply-linked list as input argument. It modifies 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.

```
 type defstruct 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.

2 Marks GATE-CSE/IT-2010( )

[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;

3) The program below uses six temporary variables a, b, c, d, e, f.

```
a = 1
b = 10
c = 20
d = a + b
e = c + d
f = c + e
b = c + e
e = b + f
d = 5 + e
return d + f
```

Assuming that all operations take their operands from registers, what is the minimum number of registers needed to execute t h i s program without spilling?

2 Marks GATE-CSE/IT-2010( )

[A] 2 [B] 3

[C] 4 [D] 6

4) In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is

1 Marks GATE-CSE/IT-2002( )

[A] log n [B] n/2

[C] $log_2^n - 1$ [D] n

5) Which of the following is NOT an advantage of using shared,dynamically linked libraries as opposed to using statically linked libraries?

2 Marks GATE-CSE/IT-2003( )

[A] Smaller sizes of executable [B] Lesser overall page fault rate in the system

[C] Faster program startup [D] Existing programs need not e relinked to take advantage of newer versions of libraries

6) A language with string manipulation facilities uses the following operations
head(s): first character of a string
tail(s): all but the first character of a string
concat(s1,s2):s1 s2
for the string acbc what will be the output of
concat(head(s), head(tail(tail(s))))

2 Marks GATE-CSE/IT-1995( )

[A] ac [B] bc

[C] ab [D] cc

7) Linked lists are not suitable data structures of which one of the following problems?

1 Marks GATE-CSE/IT-1994( )

[A] Insertion sort [B] Binary search

[C] Radix sort [D] Polynomial manipulation

Linked Lists

Key Paper

1.	**B**	**2.**	**D**	**3.**	**C**	**4.**	**D**	**5.**	**B**
6.	**C**	**7.**	**B**						

Trees & Graphs

Common Data for Q1 and Q2 is given below

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 ge inserted into a 3-ary heap containing n items by [placing x in the location a [n] and pushing it ip the tree to satisfy the heap property.

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

2 Marks ( )

[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

2) Suppose the elements 7,2,10, and 4 are insetrted, in that order, into the vlaid 3-ary max heap found in the above question, Q.87. Which one of the following is the sequence of items in the arrary representing the resultant heap ?

2 Marks GATE-CSE/IT-2006( )

[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

3) The following postfix expression with single digit operands in evaluated using a stack

8 2 3 ^ / 2 3* + 5 1* -

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

2 Marks GATE-CSE/IT-2007( )

[A] 6, 1
[B] 5,7
[C] 3,2
[D] 1,5

4) What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.

2 Marks GATE-CSE/IT-2009( )

[A] 2
[B] 3
[C] 4
[D] 5

5) A process executes the code

```
fork ();
fork ();
fork ();
```

The total number of child processes created is

1 Marks GATE-CSE/IT-2012( )

[A] 3
[B] 4
[C] 7
[D] 8

6) The worst case running time to search for an element in a balanced binary search tree with n2n elements is

2 Marks GATE-CSE/IT-2012( )

[A] Θ (n logn)
[B] $\Theta(n2^n)$
[C] Θ (n)
[D] Θ (logn)

7) The height of a tree is defined as the number of edges on the longest path in the tree. The function shown in the pseudocode below is invoked as height(root) to compute the height of a binary tree rooted at the tree pointer root.

```
int height (treeptr n)
{ if(n==NULL) return -1;
if (n → left == NULL )
if (n → right == NULL ) return 0;
else return B1;          //Box 1
else { h1 = height (n-->left);
if (n → right == NULL) return (1 + h1);
else {h2 = height (n → right);
return B2;               //Box 2
}
{
}
```

The appropriate expressions for the two boxes B1 and B2 are

2 Marks GATE-CSE/IT-2012( )

[A] B1 : (1 + height (n → right))
B2 : (1 + max (h1,h2))

[B] B1 : (height (n → right))
B2 : (1 + max (h1,h2))

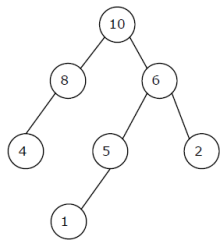
[C] B1 : height (n → right)
B2 : max (h1,h2)

[D] B1 : (1 + height (n → right))
B2 : max (h1,h2)

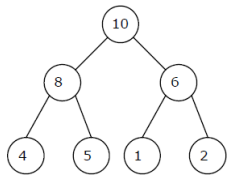
8) A max-heap is a heap where the value of each parent is greater than or equal to the value of its children. Which of the following is a max-heap?

1 Marks GATE-CSE/IT-2011( )

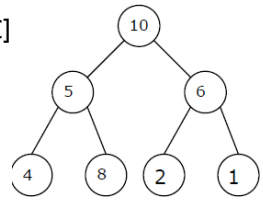
[A]



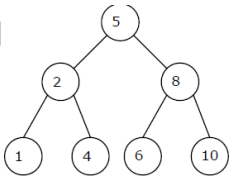
[B]



[C]



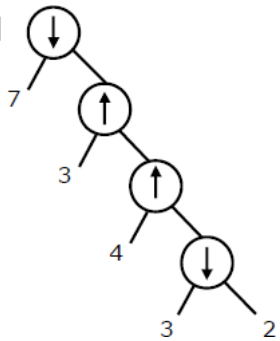
[D]



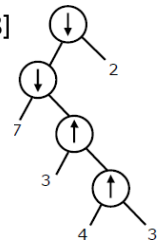
9) Consider two binary operators '↑' and '↓' with the precedence of operator ↓ being lower than that of the operator ↑. Operator ↑ is right associative while operator ↓, is left associative. Which one of the following represents the parse tree for expression (7 ↓ 3 ↑ 4 ↑ 3 ↓ 2) ?

2 Marks GATE-CSE/IT-2011( )

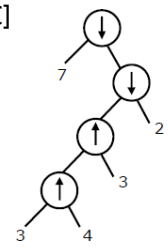
[A]



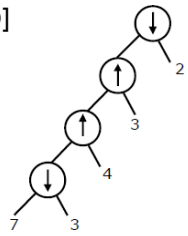
[B]



[C]



[D]



10) 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?

1 Marks GATE-CSE/IT-2011( )

[A] 0 [B] 1

[C] n! [D] $\frac{1}{n+1}.2n_{C_n}$

11) In a binary tree with n nodes, every node has an odd number of descendants. Every node is considered to be its own descendant. What is the number of nodes in the tree that have exactly one child?

2 Marks GATE-CSE/IT-2010( )

[A] 0 [B] 1

[C] (n-1) / 2 [D] n-1

12) The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is:

2 Marks GATE-CSE/IT-2002( )

[A] n/2 [B] (n-1) / 3

[C] (n-1) / 2 [D] (2n+1) / 3

13) Consider the following algorithm for searching for a given number x in an unsorted array A[1..n] having n distinct values:

1. Choose an i uniformly at random from 1..n
2. If A[i]=x then Stop else Goto 1;

Assuming that x is present A, what is the expected number of comparisons made by the algorithm before it terminates?

2 Marks GATE-CSE/IT-2002( )

[A] n [B] n-1

[C] 2n [D] n/2

14) 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?

2 Marks GATE-CSE/IT-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

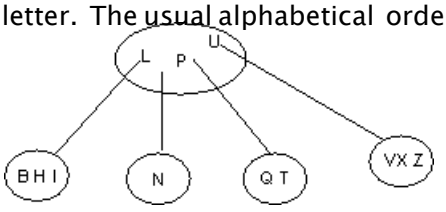
15) A binary search tree is generated by inserting in order the following integers:
50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24
The number of nodes in the left subtree and right subtree of the root respectively is

2 Marks GATE-CSE/IT-1996( )

[A] (4,7) [B] (7,4)

[C] (8,3) [D] (3,8)

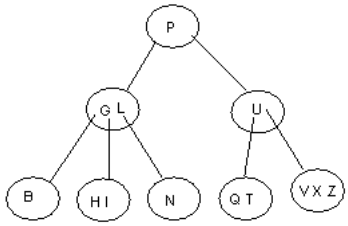
16) Consider the following 2-3-4 tree(i.e., B-tree with a minimum degree of two) in which each data item is a letter. The usual alphabetical ordering of letters is used in constructing the tree



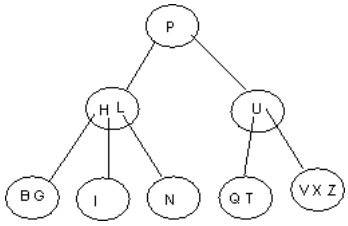
What is the result of inserting G in the above tree ?

2 Marks GATE-CSE/IT-2003( )

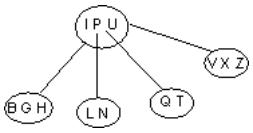
[A]



[B]



[C]



[D]None of the above

17) The following numbers are insert 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)?

1 Marks GATE-CSE/IT-2004( )

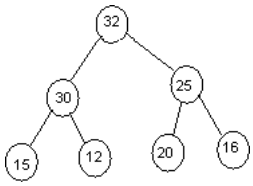
[A] 2 [B] 3

[C] 4 [D] 6

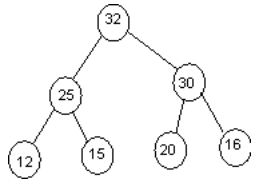
18) The elements 32, 15,20,30,12,35,16 are inserted one by one in the given order into a maxHeap. The resultant maxHeap is

1 Marks GATE-CSE/IT-2004( )

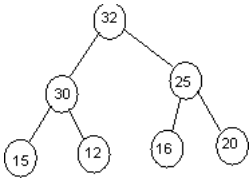
[A]



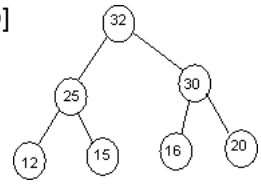
[B]



[C]



[D]



19) 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 ?

2 Marks GATE-CSE/IT-2006( )

[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] 29,15,9,10,25,22,23,27,40,0,50,95 [D] 95,50,60,40,27,23,22,25,10,0,15,29

Statement for Linked answer Q20 and Q21 is given below

20) Consider a binary max-heap implemented using an array
Which one of the following array represents a binary max-heap ?

2 Marks GATE-CSE/IT-2009( )

[A] {25,12,16,13,10,8,14} [B] {25,14,13,16,10,8,12}

[C] {25,14,16,13,10,8,12,16} [D] {25,14,12,13,10,8,16}

21) What is the content of the array after two delete operations on the correct answer to the previous question ?

2 Marks GATE-CSE/IT-2009( )

[A] {14,13,12,10,8} [B] {14,12,13,8,10}

[C] {14,13,8,12,10} [D] {14,13,12,8,10}

Trees & Graphs

Key Paper

1.	**D**	**2.**	**A**	**3.**	**A**	**4.**	**B**	**5.**	**C**
6.	**C**	**7.**	**A**	**8.**	**B**	**9.**	**B**	**10.**	**B**
11.	**A**	**12.**	**B**	**13.**	**A**	**14.**	**D**	**15.**	**B**
16.	**C**	**17.**	**B**	**18.**	**A**	**19.**	**A**	**20.**	**C**
21.	**D**								

Common Data for Q1 and Q2 is given below

The following code segment is executed on a processor which allows only register operands in its instructions. Each instruction can have almost two source operands and one destination operand. Assume that all variables are dead after this code segment

```
c = a + b;
d = c + a;
x = c * c;

if (x > a){
y = a * a;
}
else {

d = d * d;
e = e * e;

}
```

1) Suppose the instruction set architecture of the processor has only two registers. The only allowed compiler optimization is code motion, which moves statements from one place to another while preserving correctness. What is the minimum number of spills to memory in the compiled code?

2 Marks GATE-CSE/IT-2013,GATE-CSE/IT-2013( )

[A] 0 [B] 1

[C] 2 [D] 3

2) What is the minimum number of registers needed in the instruction set architecture of the processor to compile this code segment without any spill to memory? Do not apply any optimization other than optimizing register allocation

2 Marks GATE-CSE/IT-2013( )

[A] 3 [B] 4

[C] 5 [D] 6

Common Data for Q3 and Q4 is given below

The procedure given below is required to find and replace certain characters inside an input character string supplied in array A. The characters to be replaced are supplied in array oldc, while their respective replacement characters are supplied in array newc. Array A has a fixed length of five characters, while arrays oldc and newc contain three characters each. However, the procedure is flawed

```
void find_and_replace (char * A, char * oldc, char * newc) {
for (int i = 0; i <5; i + + )

for (int j = 0; j <3; j + +)

if(A[i]==oldc[j]) A[i]=newc[j];

}
```

The procedure is tested with the following four test cases

(1) oldc = " abc ", newc = " dab " (2) oldc = " cde ", newc = "bcd"

(3) oldc = " bca", newc = " cda" (4) oldc = " abc ", newc = "bac"

3) The tester now tests the program on all input strings of length five consisting of characters ‘a’, ‘b’, ‘c’, ‘d’ and ‘e’ with duplicates allowed. If the tester carries out this testing with the four test cases given above, how many test cases will be able to capture the flaw?

2 Marks GATE-CSE/IT-2013,GATE-CSE/IT-2013( )

[A] Only one [B] Only two

[C] Only three [D] All four

4) If array A is made to hold the string “abcde”, which of the above four test cases will be successful in exposing the flaw in this procedure?

2 Marks GATE-CSE/IT-2013( )

[A] None [B] 2 only

[C] 3 and 4 only [D] 4 only

5) 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 ?

1 Marks GATE-CSE/IT-2003( )

[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

6) In the following C program fragment, j,k, n and Two Log_n are integer variables, and A is an array of integers. The variable n is initialized to an integer 3, and Two Log_n is initialized to the value of $2 * \lceil log_2 n \rceil$

```
for (k=3; k <= n; k++)
   A[k] =0;
for (k=2; k <= TwoLog_n; k++)
  for (j=k+1 ; j <=n; j++)
     A[j]=A[j]||(j%k);
for (j=3; j <= n; j++)
   if (!A[j]print f("%d",j);
```

The set of numbers printed by this program fragment is

2 Marks GATE-CSE/IT-2003( )

[A] {m | m n, (i) [m=i!]}

[B] {m | m n, (i) [m=i^2]}

[C] {m | m n, m is prime }

[D] { }

7) Consider the function f defined below ;

```
struct item {
        int data;
        struct item * next;
};
int f(struct item * p) {
        return ((P = = NULL)| | (p->net = = NULL) | |
                ((P -> data <= p->next->data) &&
                        f(p ->next));
}
```

For a given linked list p, the function f returns 1 if and only if

2 Marks GATE-CSE/IT-2003( )

[A] the list is empty or has exactly one element

[B] the elements in the list are sorted in non-decreasing order of data value

[C] the elements in the list are sorted in non-increasing order of data value

[D] not all elements in the list have the same data value.

8) the goal of structured programming is to

2 Marks GATE-CSE/IT-2004( )

[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

9) Consider tech following C function

```
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 Marks GATE-CSE/IT-2004( )

[A] call swap (x,y)

[B] call swap (&x, &y)

[C] swap (x,y) cannot be used as it does not return any value

[D] swap (x,y) cannot be used as the parameters are passed by value

10) Consider the following C function

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

The value returned by f(1) is

1 Marks GATE-CSE/IT-2004( )

[A] 5

[B] 6

[C] 7

[D] 8

11) Consider the following program fragment for reversing the digits in a given integer to obtain a new integer. Let n = d1d2…, dm.

```
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

1 Marks GATE-CSE/IT-2004( )

[A] $n= d_1 d_2 \dots d_{m-i}$ and $rev=d_m d_{m-1} \dots d_{m-i+1}$

[B] $n = d_{m-i+1} \dots d_{m-1} d_m$ or $rev= d_{m-i} \dots d_2 d_1$

[C] $n \neq rev$

[D] $n= d_1 d_2 \dots d_m$ or $rev= d_m \dots d_2 d_1$

12) Consider the following C program segment :

```
char p [20] ;
char * s = " string ' ;
int length= strlen (s);
for   (i=0 ; i < length ; i++)
     p[i] = s[length-i];
print f("%s", p);
```

The output of the program is

1 Marks GATE-CSE/IT-2004( )

[A] gnirts

[B] string

[C] gnirt

[D] no output is printed

13) Consider the following C program

```
main ( )
{ int x,y,m,n ;
   scan f ("%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;
        }
    Print f("%d", n);
    }
```

The program computes

1 Marks GATE-CSE/IT-2004( )

[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

14). What does the following algorithm approximate ? (Assume m> 1, € >0).

```
x = m,
        y = 1;
        while (x-y >€ )
                {       x = (x +y)/2;
                        y = m / x ;
                }
        Print (x);
```

1 Marks GATE-CSE/IT-2004( )

[A] log m
[B] m^2
[C] $m^{\frac{1}{2}}$
[D] $m^{\frac{1}{3}}$

15) Choose the best matching between the programming styles in Group 1 and their characteristics in Group 2

Group –1	Group-2
P. Functional	1. Command-based,procedural
Q. Logic	2. Imperative, abstract data types
R. Object-oriented	3. Side-effect free, declarative, expression evaluation
S. Imperative	4. Declarative, clausal representation, theorem proving

1 Marks GATE-CSE/IT-2004( )

[A] P-2, Q-3, R-4, S-1
[B] P-4, Q-3, R-2, S-1
[C] P-3, Q-4, R-1, S-2
[D] P-3, Q-4, R-2, S-1

16) What does the following C-statement declare ?
int(*f) (int*);

1 Marks GATE-CSE/IT-2004( )

[A] A function that takes an integer pointer as argument and returns and interger
[B] A function that takes an integer pointer as argument and return 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 returns a function pointer

17) Which of the following are essential features of an object-oriented programming languages ?
1. Abstraction and encapsulation
2. Strictly-typedness
3. Type-safe property coupled with sub-type rule
4. Polymorphism in the presence of inheritance

1 Marks GATE-CSE/IT-2005( )

[A] 1 and 2 only
[B] 1 and 4 only
[C] 1,2 and 4 only
[D] 1,3 and 4 only

18) An Abstract Data type (ADT) is

1 Marks GATE-CSE/IT-2005( )

[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

19) A common property of logic programming languages and functional languages is

1 Marks GATE-CSE/IT-2005( )

[A] both are procedural language
[B] both are based on - calculus
[C] both are declarative
[D] all of the above

20) 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

1 Marks GATE-CSE/IT-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

21) Consider these two functions and two statements S1 and S2 about them.

```
int work 1(int*a,int i, int j)
{
        int x=a[i+2];
        a[j]=x+1;
        return a[i+2]-3;}
}
```

```
intwork 2(int*a,int i, int j)
{
        int t1=i+2;
        int t2=a[t1];
        a[j] = t2 +1;
         return t2 -3;
}
```

S1: The transformation from work 1 to work 2 is valid, ie., for any program state and input arguments, work 2 will compute the same output and have the same effect on program state as work 1
S2: All the transformations applied to work 1 to get work 2 will always improve the performance (i.e., reduce CPU time) of work 2 compared to work 1

2 Marks GATE-CSE/IT-2006( )

[A] S1 is false and S2 is false
[B] S1 is false and S2 is true
[C] S1 is true and S2 is false
[D] S1 is true and S2 is true

22) Consider this C code to swap two integers and these five statements : the code

```
void swap (int*px, int*py) {
        *px = *px - *py;
        *py = *px + *py ;
        *px = *py - *px ;
}
```

S1 : will generate a compilation error
S2: may generate a segmentation fault at runtime depending on the arguments passed
S3: Correctly implements the swap procedure for all input pointers refereeing to integers stored in memory locations accessible to the process
S4: implements the swap procedure correctly for some but not all valid input pointers
S5: may add or subtract integers and pointers

2 Marks GATE-CSE/IT-2006( )

[A] S_1
[B] S_2 and S_3
[C] S_2 and S_4
[D] S_2 and S_5

23) Consider the followingsegment of C-code

```
int, J, n;
  j =1;
while (j <=n)
   j=j*2;
```

The number of comparisonsmade in the execution of the loop for any n>0 is

1 Marks GATE-CSE/IT-2007( )

[A] $[log_2 n]+1$
[B] n
[C] $[log_2 n]$
[D] $|log_2 n| + 1$

24) 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) ?

2 Marks GATE-CSE/IT-2007( )

[A] 5
[B] 7
[C] 9
[D] 18

25) Consider the following Cprogram segment where Cell Node represents a node in a binary tree

```
struct CellNode {
struct CelloNode * leftchild;
int element;
struct CellNode *rightchild;
};
int GetValue (structCellNode * ptr){
        int value=0;
if (ptr!=NULL){
    if((ptr-> leftChild = = NULL)&&
        (ptr-> rightChild = = NULL))
        Value =1 ;
else
value = value + GetValue
                (ptr->left Child)
        +        Get Value
                (ptr->right Child);
}
return (value);
}
```

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

2 Marks GATE-CSE/IT-2007( )

[A] the number of nodes
[B] the number of internal nodes in the tree
[C] the number of leaf nodes in the tree
[D] the height of the tree

26) Which combination of the integer variables x,y and z makes the variable a get the value4 in the following expression ?

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

1 Marks GATE-CSE/IT-2008( )

[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

27) What will be the output of the following C program segment?

```
Char inChar = 'A';
switch (inChar ) {
case 'A' : printf ("Choice A \ n");
case 'B' :
case 'C' : print f("Choice B");
case 'D' :
case 'E' :
default : printf ("No Choice") ; }
```

1 Marks GATE-CSE/IT-2012( )

[A] No choice
[B] Choice A
[C] Choice A
Choice B is No choice
[D] Program gives no output as it is erroneous

28) Consider the program given below, in a block-structured pseudo-language with lexical scoping and nesting of procedures permitted.

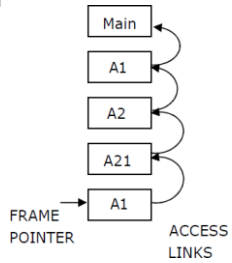
```
Program main;
Var . . .
Procedure A1;
Var ..
Call A2;
End A1
Procedure A2;
Var . . .
Procedure A21;
Var . . .
Call A1;
End A21
Call A21;
End A2
Call A1;
End main.
```

Consider the calling chain: Main → A1 → A2 → A21 → A1

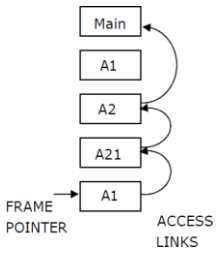
The correct set of activation records along with their access links is given by

2 Marks GATE-CSE/IT-2012( )

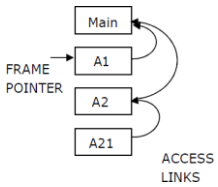
[A]



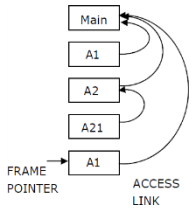
[B]



[C]



[D]



29) What does the following fragment of C-program print?

```
char c [] = " GATE2011";
char *p =c;
printf ("%s", p+p [3] - p [1]);
```

1 Marks GATE-CSE/IT-2011( )

[A] GATE2011

[B] E2011

[C] 2011

[D] 011

30) In a compiler, keywords of a language are recognized during

1 Marks GATE-CSE/IT-2011( )

[A] parsing of the program

[B] the code generation

[C] the lexical analysis of the program

[D] dataflow analysis

31) 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);
pr int f (" %d %d \ n ", i, j);
return 0;
}
```

1 Marks GATE-CSE/IT-2010( )

[A] 2 2 [B] 2 1

[C] 0 1 [D] 0 2

32) Which data structure in a compiler is used for managing information about variables and their attributes?

1 Marks GATE-CSE/IT-2010( )

[A] Abstract syntax tree [B] Symbol table

[C] Semantic stack [D] parse table

33) Which languages necessarily need heap allocation in the runtime environment?

1 Marks GATE-CSE/IT-2010( )

[A] Those that support recursion [B] Those that use dynamic scoping

[C] Those that allow dynamic data structures [D] Those that use global variables

34) In the C language

1 Marks GATE-CSE/IT-2002( )

[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 fro 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.

35) The results returned by function under value-result and reference parameter passing conventions

1 Marks GATE-CSE/IT-2002( )

[A] Do not differ [B] Differ in the presence of loops

[C] Differ in all cases [D] May differ in the presence of exception

36) The C language is:

2 Marks GATE-CSE/IT-2002( )

[A] A context free language [B] A context sensitive language

[C] A regular language [D] Parsable fully only by a Turing machine

37) Dynamic linking can cause security concerns because

2 Marks GATE-CSE/IT-2002( )

[A] Security is dynamic

[B] The path for searching dynamic libraries is not known till runtime

[C] Linking is insecure

[D] Cryptographic procedures are not available for dynamic linking

38) 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;
}
```

2 Marks GATE-CSE/IT-2001( )

[A] 10,3 [B] 31, 3

[C] 27, 7 [D] None of the above

39) Consider the following three C functions:

```
[P1]  int*g(void)
      {
       intx=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?

2 Marks GATE-CSE/IT-2001( )

[A] Only P3 [B] Only P1 and P3

[C] Only P1 and P2 [D] P1, P2 and P3

40) Consider the following program

```
Program P2
var n:int:
procedure W(var x:int)
begin
x=x+1;
printx;
end
procedure D
begin
var n:int;
n=3;
W(n);
End
begin                    \\begin P2
n=10;
D;
end
```

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

2 Marks GATE-CSE/IT-2001( )

[A] 10 [B] 11

[C] 3 [D] None of the above

41) 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

1 Marks GATE-CSE/IT-2003( )

[A] X^y [B] e^x

[C] In(1+x) [D] X^x

42) Consider the following C-function in which a [n] and b [m]are two sorted integer arrays and c[n + m]be another integer array.

```
void xyz(int a[], int b[], int c []){
    int i,j,k;
   i=j=k=0;
   while ((i
   if(a[i]
    else c [k++] = b[j++];
}
```

Which of the followingcondition (s) hold (s) after the termination of the while loop ?

i,jii, i

2 Marks GATE-CSE/IT-2006( )

[A] Only (i) [B] Only (ii)

[C] Neither (i) nor (ii) [D] Either (i) or (ii) but not both

43) Which of the following are true ?

(i) A programming language which does not permit global variables of any kind and has no nesting of procedures / functions, but permits recursion can be implemented with static storage allocation

(ii) Multi-level access link (or display arrangement is needed to arrange activation records only if the programming language being implemented has nesting of procedures/function

(iii) Recursion in programming languages cannot be implemented with dynamic storage allocation

(iv) Nesting of procedures/functions and recursion require a dynamic heap allocation scheme for activation records

(v) Programming languages which permit a function to return a function as its result cannot be implemented with a stack-based storage allocation scheme for activation records

1 Marks GATE-CSE/IT-2008( )

[A] (ii) and (v) only
[B] (i), (iii) and (iv) only
[C] (i), (ii) and (v)
[D] (ii), (iii) and (v) only

44) Consider line number 3 of the following C-program.

```
int main ( ) {                  |* Line 1 * |
int I, N;                       |* line 2 * |
fro (I=0, 1
}
```

Identify the compiler's response about this line while creating the object-module

2 Marks GATE-CSE/IT-2005( )

[A] No compilation error
[B] Only a lexical error
[C] Only syntactic errors
[D] Both lexical and syntactic errors

45) The following C declarations

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

define s to be

1 Marks GATE-CSE/IT-2000( )

[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

46) The most appropriate matching for the following pairs

X: m=malloc(5); m= NULL;	1: using dangling pointers
Y: free(n); n->value=5;	2: using uninitialized pointers
Z: char *p; *p='a';	3. lost memory

is:

1 Marks GATE-CSE/IT-2000( )

[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

47) Aliasing in the context of programming languages refers to

1 Marks GATE-CSE/IT-2000( )

[A] multiple variables having 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

48) 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

1 Marks GATE-CSE/IT-2000( )

[A] 22 bytes [B] 14 bytes

[C] 18 bytes [D] 10 bytes

49) The number of tokens in the following C statement

```
printf("i=%d, &i=%x",i,&i);
```

is

1 Marks GATE-CSE/IT-2000( )

[A] 3 [B] 26

[C] 10 [D] 21

50) The value of j at the end of the execution of the following C program

```
int incr (int i)
{
static int count = 0;
count = count + i;
return (count);
}
main() {
int i,j;
for(i=0; i<=4; i++)
j=incr(i)
}
```

is

2 Marks GATE-CSE/IT-2000( )

[A] 10 [B] 4

[C] 5 [D] 7

51) Given the programming constructs (i) assignment (ii) for loops where the loop parameter cannot be changed within the loop (iii) if-then-else (iv) forward go to (v) arbitrary go to (vi) non-recursive procedure call (vii) recursive procedure/function call (viii) repeat loop, which constructs will you not include in a programming language such that it should be possible to program the terminates (i.e., halting) function in the same programming language.

2 Marks GATE-CSE/IT-1999( )

[A] (ii), (iii), (iv) [B] (v), (vii), (viii)

[C] (vi), (vii), (viii) [D] (iii), (vii), (viii)

52) Consider the following program in a language that has dynamic scooping:

```
var x: real;
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:

2 Marks GATE-CSE/IT-1999( )

[A] 0.125 0.125 [B] 0.25 0.25

[C]0.25 0.125 [D]0.125 0.25

53) Consider the following C function definition

```
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:

2 Marks GATE-CSE/IT-1999( )

[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

54) Heap allocation is required for languages.

1 Marks GATE-CSE/IT-1997( )

[A] that support recursion [B] that support dynamic data structures

[C]that use dynamic scope rules [D]None of the above

55) Assume that X and Y are non-zero positive integers. What does the following Pascal program segment do?

```
while X <>Y do
if        X >Y then
X:=X - Y
else
Y:=Y -X;
write(X);
```

2 Marks GATE-CSE/IT-1995( )

[A] Computes the LCM of two numbers [B]Divides larger number by the smaller number

[C]Computes the GCD of two numbers [D]None of the above

56) Which of the following statements is true?

I. As the number of entries in a hash table increases, the number of collisions increases.
II. Recursive programs are efficient
III. The worst case complexity for Quicksort is O _(n_2 _)
IV. Binary search using a linear linked list is efficient.

2 Marks GATE-CSE/IT-1995( )

[A]I and II [B]IIandIII

[C]IandIV [D]IandIII

57) FORTRANimplementationdonotpermitrecursionbecause

1 Marks GATE-CSE/IT-1994( )

[A] theyusestaticallocationforvariables [B] theyusedynamicallocationforvariables

[C]stacksarenotavailableonallmachines [D]itisnotpossibletoimplementrecursiononallmachines

58) Anunrestricteduseofthe“goto” statementisharmfulbecause

1 Marks GATE-CSE/IT-1994( )

[A] itmakesitmoredifficulttoverifyprograms [B] itincreasestherunningtimeoftheprograms

[C]itincreasesthememory required fortheprograms [D]itresultsinthecompilergeneratinglongermachinecode

59) For the program segment given below, which of the following are true?

```
program main (output);
type link = ^data;
data = record
d : real;
n : link
end;
var ptr : link;
 begin
new (ptr);
 ptr:=nil;
.ptr^.d:=5.2;
write ln(ptr)
end.
```

2 Marks GATE-CSE/IT-1993( )

[A] The program leads to compile time error

[B] The program leads to run time error

[C] The program outputs 5.2

[D] The program produces error relating to nil pointer dereferencing

[E] None of the above

Program analysis

Key Paper

1.	**B**	**2.**	**B**	**3.**	**B**	**4.**	**C**	**5.**	**A**
6.	**D**	**7.**	**B**	**8.**	**C**	**9.**	**D**	**10.**	**C**
11.	**A**	**12.**	**D**	**13.**	**C**	**14.**	**C**	**15.**	**D**
16.	**C**	**17.**	**B**	**18.**	**C**	**19.**	**C**	**20.**	**A**
21.	**D**	**22.**	**B**	**23.**	**D**	**24.**	**D**	**25.**	**C**
26.	**A**	**27.**	**C**	**28.**	**D**	**29.**	**C**	**30.**	**C**
31.	**D**	**32.**	**B**	**33.**	**C**	**34.**	**B**	**35.**	**D**
36.	**B**	**37.**	**B**	**38.**	**B**	**39.**	**C**	**40.**	**D**
41.	**B**	**42.**	**D**	**43.**	**B**	**44.**	**C**	**45.**	**A**
46.	**D**	**47.**	**A**	**48.**	**C**	**49.**	**C**	**50.**	**A**
51.	**B**	**52.**	**C**	**53.**	**C**	**54.**	**B**	**55.**	**C**
56.	**D**	**57.**	**A**	**58.**	**A**	**59.**	**E**		