

```
count++;  
}  
}  
}
```

20. Find the complexity of the following code fragment

```
for( i=1; i <= n; i = i * 2)  
for( j = 1; j <= i; j++ )  
sum = sum + j
```

21. Find the complexity of the following code fragment while the function `foo()` is defined as follows:

```
foo(n)  
{  
//returns closest integer to  $\frac{n}{2}$  in  $\Theta(\log n)$  time  
}
```

```
int i, j, k, count 0;  
for (i = n; i >= 1; i --)  
{  
for (j = i; j <= n!; j = j * j)  
{  
for (k = n; k >= 1; k = foo(k))  
{  
count++;  
}  
}  
}
```

2 C Programming

1. What does the following C program do? Assume all necessary libraries are included.

```
int main()  
{  
int x, i;  
scanf("%d", &x);  
for(i = 1; i <= 100; i ++)  
{  
if((i%x) == 3)  
{
```

```
printf(“%d ”, i);
}
}
return 0;
}
```

- (A) Prints all numbers from 1 to 100 divisible by 3
- (B) Prints all numbers from 1 to 100 divisible by some integer x.
- (C) Prints all numbers from 1 to 100 leaving remainder 3 when divided by some integer x.
- (D) Compilation error

2. What does this program do for some integers n1 and n2 already defined?

```
int foo(int n);
int main()
{
int n1, n2, i, flag;
for(i=n1+1; i≤n2; ++i)
{
flag = foo(i);
if(flag == 1)
printf(“%d ”, i);
}
return 0;
}
```

```
int foo(int n)
{
int j, flag = 1;
for(j=2; j ≤ n/2; ++j)
{
if (n%j == 0)
{
flag =0;
break;
}
}
return flag;
}
```

Choose the correct statement:

- (A) Prints all numbers less than n1 which are divisible by n2
- (B) Prints all numbers between n1 and n2 which are mutually co-prime
- (C) Prints all prime numbers between n1 and n2.
- (D) Prints all numbers between n1 and n2 which are not prime.

3. What is the output of this program?

```
int main(){
printf("%d",printf("GATE 2019 ")*10);
return (0);
}
```

- (A) GATE 2019
- (B) GATE 40
- (C) GATE 2019 90
- (D) GATE 2019 100
- (E) Compilation error

4. What is the output of this code:

```
int main(){
printf("\hello\", \".\.\\.\\.\\.\"world.");
}
```

- (A) "hello", "...", "rld.
- (B) "hello", "...", "world.
- (C) hello...world.
- (D) hello, rld.
- (E) "hello" ... "world."

5. Consider the following program:

IsPrime() returns 1 if number is prime else returns 0.

```
int main() {
int n, i, flag = 0;
scanf("%d", &n);
for(i = 2; i ≤ n/2; ++i)
{
if (IsPrime(i) == 1)
{
if (IsPrime(n-i) == 1) {
flag = 1;
printf("Yes");
}
}
}
if(!flag) printf("No");
return 0; }
```

Which of the following statements is correct regarding the behaviour of this program?

- (A) It finds any possible prime number smaller than n and then prints “Yes” else prints “No”.
- (B) It finds 2 numbers coprime to each other smaller than n and then prints “Yes” else prints “No”
- (C) Prints all prime numbers smaller than n and prints “Yes” else prints “No”.
- (D) Prints “Yes” if n can be written as sum of 2 prime numbers else prints “No”.

6. What is the following piece of code doing?

```
int main(){
int n1, n2, i, j;
scanf("%d %d", &n1, &n2);
for(i=1; i ≤ n1 && i ≤ n2; ++i)
{
if(n1%i==0 && n2%i==0)
j = i;
}
printf("%d", j);
return 0;
}
```

- (A) Prints all numbers which are divisible by either n1 or divisible by n2.
- (B) Prints all numbers which are divisible by both n1 AND n2.
- (C) Prints a number which divides both of n1 and n2.
- (D) Prints a number which divides both of n1 and n2 such that there is no other such number which is greater than this number.

7. What is the output of this program?

```
int main(){
int i=-3,j=2 k=0,m;
m=++i&&++j|| + +k;
printf(" %d%d%d%d ",i,j,k,m);
}
```

- (A) -2301
- (B) -3301
- (C) -2311
- (D) -3311

8. Write down the output of this program?

```
int main() {
char *s[]={“iceland”,“Greenland”,“Ireland”,“Switzerland”};
char **ptr[]={s+3,s+2,s+1,s};
char ***p=ptr;
printf(“%s ”,**++p);
}
```

```
printf(“%s ”,*- -*++p+3);
printf(“%s ”,*p[-2]+3);
printf(“%s ”,p[-1][-1]+1);
}
```

9. What is printed by following C program:

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

- (A) 18
(B) 19
(C) 20
(D) 21

10. What is the output of this program?

```
int main(void){
char *ptr = “Linux”;
printf(“ [%c] ”,*ptr++);
printf(“ [%c] ”,*++ptr);
return 0;
}
```

- (A) [L][i]
(B) [L][u]
(C) [L][n]
(D) [i][u]

11. Something is wrong with this code. Can you detect it?

```
int main(void){
char *ptr = "Linux";
ptr = 'T';
printf(" [%s] ", ptr);
return 0;
}
```

12. What is the output of this program?

```
int main(void){
int a = 10, b = 20, c = 30;
printf(" %d..%d..%d ", a+b+c, (b = b*2), (c = c*2));
return 0;
}
```

- (A) 60..40..60
(B) 110..40..60
(C) 110..20..30
(D) 60..20..30

13. Write a C program which prints a semicolon, *i.e.* “;”, without using it anywhere in the program.

14. What is the functionality of this function operating on a given integer array and its size?

```
void foo(int arr[], int n){
int g, i, j;
for (i=0; i<n; i++)
{
g = -1;
for (j = i+1; j<n; j++)
{
if (arr[i] < arr[j])
{
g = arr[j];
break;
}
}
printf("%d : %d ", arr[i], g);
}
}
```

- (A) Reverses the given array
(B) Prints the maximum element in the array if any else returns -1

- (C) Prints the next greater element if any else returns -1
- (D) Prints the next smaller element if any else returns -1

15. There is some issue with this program. Which of the following Lines contains it?

```
int* inc(int val){
Line1: int a = val;
a++;
Line2: return &a;
}
```

```
int main(void)
{
int a = 10;
Line3: int *val = inc(a);
Line4: printf(" Incremented value is equal to [%d] ", *val);
return 0;
}
```

- (A) Line 1
- (B) Line 2
- (C) Line 3
- (D) Line 4

16. What this program is doing?

```
int main(){
int x = 10;
int y = 10;
if ( !(x ^ y) )
printf(" Yes");
else
printf(" No ");
return 0;
}
```

- (A) Computes x^y and prints Yes or No based on the result.
- (B) Computes y^x and prints Yes or No based on the result.
- (C) Compare x and y and prints Yes or No based on the result.
- (D) Performs EXOR operation with x and y and prints Yes or No based on the result.

17. Which of the following statements is correct for this piece of code?

```
int main() {
    int i=1,j;
    for(i=49,j=57;i<=j;i++,j--)
    {
        printf("%c %c ",i,j);
    }
}
```

- (A) 49 57 50 56 51 55 52 54 53 53
(B) 49 57 50 56 51 55 52 54
(C) 1 9 2 8 3 7 4 6 5 5
(D) 1 9 2 8 3 7 4 6

18. Which of the following statements about the below program is correct?

```
int main(void){
    char buff[10];
    memset(buff,0,sizeof(buff));
    gets(buff);
    printf(" The buffer entered is [%s] ",buff);
    return 0;
}
```

- (A) The program assigns a memory area to *buff* successfully and works fine.
(B) The program may not compile properly.
(C) The program is compiled fine but it may lead to buffer overflow sometimes.
(D) The program is fine and has no issue.

19. Which of the following statements about the below program is correct?

```
void main(void){
    char *ptr = (char*)malloc(10);
    if(NULL == ptr)
    {
        printf(" Malloc failed ");
        return;
    }
    else
    {
        // Do some processing
        free(ptr);
    }
    return;
}
```

- ```
}
```
- (A) The code will compile error free
  - (B) The code will compile error free but with a warning
  - (C) The code fails to compile
  - (D) It may lead to memory leak.

20. `atexit(func())` is a tool through which user can call certain user defined function while exiting the program. Consider following program.

```
void func(void){
printf(" Cleanup function called ");
return;
}
int main(void){
int i = 0;
atexit(func);
for(;i<=0xfffff;i++);
_exit(0);
}
```

Which of the following statements about the below program is correct?

- (A) The code will not compile.
  - (B) The code will compile error free but it won't print anything.
  - (C) The code will compile error free and it prints *Cleanup function called*.
  - (D) It leads to a run time error.
21. 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 (sizeof(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

22. 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 ", sum);
getchar();
}
```

What does the above program print?

- (A) 8, 4, 0, 2, 14
- (B) 8, 4, 0, 2, 0
- (C) 2, 0, 4, 8, 14
- (D) 2, 0, 4, 8, 0

23. 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);
getchar();
}
```

The output of this program is

- (A) 1276
- (B) 22 12 11
- (C) 14 6 6
- (D) 766

24. Consider the following C functions:

```
int f1(int n){
if(n == 0||n == 1)
return n;
else
return (2*f1(n-1) + 3*f1(n-2));
}
```

```
int f2(int n){
int i;
int X[N], Y[N], Z[N] ;
X[0] = Y[0] = Z[0] = 0;
X[1] = 1; Y[1] = 2; Z[1] = 3;
for(i = 2; i ≤ n; i++){
X[i] = Y[i-1] + Z[i-2];
Y[i] = 2*X[i];
Z[i] = 3*X[i];
}
return X[n] ;
}
```

The running time of  $f1(n)$  and  $f2(n)$  are:

- (A)  $\Theta(n)$  and  $\Theta(n)$
- (B)  $\Theta(2^n)$  and  $\Theta(n)$
- (C)  $\Theta(n)$  and  $\Theta(2^n)$
- (D)  $\Theta(2^n)$  and  $\Theta(2^n)$

25. Consider the following C functions:

```
int f1(int n){
if(n == 0||n == 1)
return n;
else
return (2*f1(n-1) + 3*f1(n-2));
}
```

```
int f2(int n){
int i;
int X[N], Y[N], Z[N] ;
X[0] = Y[0] = Z[0] = 0;
X[1] = 1; Y[1] = 2; Z[1] = 3;
for(i = 2; i ≤ n; i++){
X[i] = Y[i-1] + Z[i-2];
Y[i] = 2*X[i];
Z[i] = 3*X[i];
}
return X[n] ;
}
```

f1(8) and f2(8) return the values

- (A) 1661 and 1640
  - (B) 59 and 59
  - (C) 1640 and 1640
  - (D) 1640 and 1661
26. 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));
getchar();
return 0;
}
```

- (A) -9
- (B) 5
- (C) 15
- (D) 19

27. What does the following fragment of C-program print?

```
char c[] = "CSGATE2019";
char *p =c;
printf("%s", p + p[5] - p[3]) ;
```

- (A) GATE2019
- (B) TE2019
- (C) T2019
- (D) 2019

28. 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; }
```

What is the return value of the function foo when it is called as foo(2049, 2)?

- (A) 9
- (B) 8
- (C) 5
- (D) 2

29. 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; }
```

What is the return value of the function foo when it is called as foo(2019, 20) ?

- (A) 109
- (B) 24
- (C) 19

(D) 7

30. 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() {
printf ("Enter Text ");
reverse();
}
```

- (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);`
31. Consider the following C declaration:
- ```
struct {
short s [50];
union {
float x,y,z;
long w;}u;
int p,q; } t;
```
- Assume that objects of the type short, int, float and long occupy 2 bytes, 4 bytes, 4 bytes and 8 bytes, respectively. The memory requirement for variable t, ignoring alignment considerations, is :
- (A) 118 bytes
(B) 120 bytes
(C) 122 bytes
(D) 116 bytes
32. 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<n) && (j<m))
        if (a[i] < b[j]) c[k++] = a[i++];
        else c[k++] = b[j++];
    }
```

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

- (i) $j < m$, $k = n+j-1$, and $a[n-1] < b[j]$ if $i = n$
 - (ii) $i < n$, $k = m+i-1$, and $b[m-1] \leq a[i]$ if $j = m$
- (A) only (i)
(B) only (ii)
(C) either (i) or (ii) but not both
(D) neither (i) nor (ii)

33. Consider the following C declaration

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

34. Consider the following C program segment:

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

The output of the program is:

- (A) gnirts
(B) gnirt
(C) string

(D) no output is printed

35. 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

36. Consider the function func shown below:

```
int func(int num){
int count = 0;
while (num)
{
count++;
num >>= 1;
}
return (count);
}
```

The value returned by func(2019) is:

- (A) 10
- (B) 11
- (C) 12
- (D) 13

37. Assume the following C variable declaration

```
int *A [10], B[10][10];
I A[2]
II A[2][3]
III B[1]
IV B[2][3]
```

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

- (A) I, II, and IV only
- (B) II, III, and IV only
- (C) II and IV only

(D) IV only

38. What is the interpretation of following C declaration:

```
char* (*(foo[5])(char*))[];
```

- (A)foo is an array of 5 pointers to a function that accepts a pointer to an array of pointers to a char.
- (B)foo is an array of 5 pointers to a function that accepts a pointer to a char and returns a pointer to an array of pointers to a char.
- (C)foo is an array of 5 pointers to a function that accepts a pointer to a char and returns a pointer to a char.
- (D) Not a valid C declaration.

39. What is the interpretation of following C declaration:

```
char *(*foo[8]())[];
```

- (A)foo is array of 8 pointer to pointer to function returning pointer to array of pointer to char.
- (B)foo is array of array of 8 pointer to pointer to function returning pointer to pointer to array of char.
- (C) foo is array of array of 8 pointer to pointer to function returning pointer to array of pointer to char.
- (D) None

40. 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 in row major order from memory address 0, the address of a[40][50] is:

- (A) 4040
- (B) 4050
- (C) 5040
- (D) 5050

41. Consider the following declaration of a 'two-dimensional array in C:

```
char a[100][200];
```

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

- (A) 4040

- (B) 4050
- (C) 5040
- (D) 5050

3 Stacks and Queues

1. Which one of the following is an application of Stack Data Structure?
 - (A) Managing function calls
 - (B) The stock span problem
 - (C) Arithmetic expression evaluation
 - (D) All of the above

2. Which of the following is true about linked list implementation of stack?
 - (A) In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
 - (B) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
 - (C) In push operation, if new nodes are inserted at the beginning, then in pop operation, nodes must be removed from the beginning.
 - (D) None of the above

3. Following is C like pseudo code of a function that takes a number as an argument, and uses a stack S to do processing.

```
void fun(int n){
Stack S; // Say it creates an empty stack S
while (n > 0)
{
// This line pushes the value of n%2 to stack S
push(&S, n%2);
n = n/2;
}
// Run while Stack S is not empty
while (!isEmpty(&S))
printf("%d ", pop(&S)); // pop an element from S and print it
}
```

What does the above function do in general?

- (A) Prints binary representation of n in reverse order.
- (B) Prints binary representation of n
- (C) Prints the value of $\log n$.
- (D) Prints the value of $\log n$ in reverse order.