39. Consider the following program:

```
Program main
x : integer ;
Procedure Sub1
begin
      write (x) ;
end
Procedure Sub2
   x : integer;
begin
   x = 10;
   Sub1;
end
begin
   x = 10 ;
   sub1;
end
begin
   x = 5 ;
   sub2 ;
end
```

What is the output of the program under static & Dynamic scoping?

(a) 10, 5

(b) 10, 10

(c) 5, 5

(d) 5, 10

@ 39⇒ main –

```
x : int ;
Procedure sub1
begin
    write (x);
end
Procedure sub2
    x : int
begin
    x = 10
    sub1;
end
begin
    x = 10 ; sub1;
end
begin
    x = 5 ; sub2;
end
```

Under Static & Dynamic –

Static	Dynamic
Sub2 call Sub1	Sub2 call Sub1
print (10)	→ 10
→ (10)	Again
	Sub2 call Sub2
	→ (5)

Ans → @ (10, 5)

Teacher's Signature .....

40. Which of the following is a logic / constraint based language?

(a) LISP
(b) Small talk
(c) C
(d) Prolog

(D) 40⇒ Logic/Constraint based Language ?

(a) LISP
(b) Small Talk → ( )
(c) C
(d) Prolog

41. A certain computer architecture supports only the immediate and the direct addressing modes. Which of the following programming language features cannot be implemented on this processor?

(a) Pointers

(b) Arrays

(c) Records

(d) Recursive procedures with local variable

(a) 41] Only the immediate and the direct addressing modes. Can not implemented on this processor.

(a) Pointers uses indirect address mode.

42. Consider the following program segment of a hypothetical high – level programming language.

```
x, y : integer ;
Procedure A(z : integer)
x : integer ;
begin
        x = 1;
        B ;
        z = x ;
end ;
Procedure  B ( )
begin
        x = x+1;
end;
```

```
begin { main }
        x = 5 ;
        A (y) ;
        Write (y) ;
end ;
```

Assuming the parameter 'z' is passed by reference and the language uses dynamic scoping, what is the output of the program.

(a) 5 (b) 6 (c) 7 (d) 2

(d) 42=8

```
x, y : int
procedure A (z: int)
    x : int ;
    begin
        x = 1;
        B; z = x
    end;
Procedure B()
    begin
        x = x + 1;
    end;
    begin {main}
        x = 5; A(y); Print(y);
    end;
```

```
A(z)
{ x = 1;
  B;
  z = x;
}
B()
{ x = x + 1; => x = 1 + 1 = 2
}
main()
{ x = 5;
  A(y); Print(y);
}
```

z = 2

Teacher's Signature .....

'z' Passed by reference & Dynamic Scopic

z holds 2

print (z) → 2

Ans - d. 2

43. Consider the following program segment

```
integer x;
procedure Q(z: integer)
begin
        z = z+x;
        Print (z);
end

procedure P(y: integer)
        x: integer
begin
        x = y+2;
        Q(x);
        Print (x);
end;
begin {main}
        x=5;
        P(x);
        print(x);
end
```

What is the output of the program with call by reference parameter passing & dynamic scoping rule?

(a) 14, 14, 5
(b) 7, 14, 5
(c) 14, 7, 5
(d) 14, 14, 14

(d) 43⇒

```
int x;
Procedure Q(z: int)
begin
    z = z + x;
    print(z);
end
Procedure P(y: int)
~~begin~~ x: int
begin
    x = y + 2;
    Q(x);
    print(x);
end;
begin {main}
    x = 5;
    P(x);
    print(x);
end
```

Ans → (d) 14, 14, 14

Call by reference

```
main()
{ call P(x)
  P()
{ x = 5 + 2
  x = 7
  Q(7)
{ z = 7 + 7 = 14
  print(14)
}
14
}
```

x: 5 → 7 → 100 (100 → 200)

z: 100

Parameter passing

Same passing value

Print(14)

14

Pynomic

print(14)

Teacher's Signature .....

44. What is the output of the following C program:

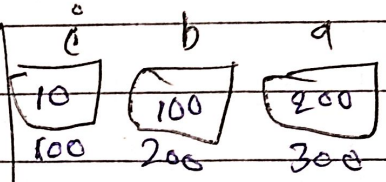
```
int f(int x, int xy, int xxz)
{
        int k, m;
        **z + = 2;
        m = **z;
        *y+ = 1;
        k = *y;
        x+ = 5;
        return (x+k+m);
}
main( )
{
        int i = 10, *b, **a, t ;
        b = &i;
        a = &b;
        t= f(i,b,a);
        printf("%d", t);
}
```

(a) 32 (b) 35

(c) 38 (d) 40

(d) 495 Program:

```
int f(int x, int *y, int **z)
{
    ------
    ------
}
main()
{
    i=10; *b, **a, t;
    b = &i;
    a = &b;
    t = f(i, b, a);
    printf("%d", t);
}
```

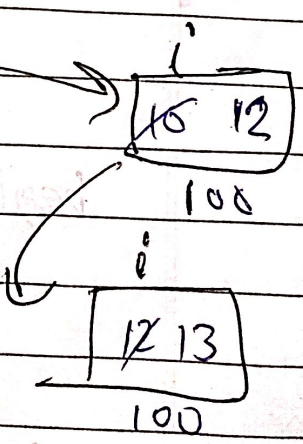


t = f(i, b, a);

f(i, b, a) ⟶ f(10, 100, 200)

```
{
    int k, m;
    **z = **z + 2 = 10 + 2 = 12
    m = **z = 12
    *y = *y + 1 = 12 + 1 = 13
    k = *y = 13
    x = x + 5 = 10 + 5 = 15
    return (15 + 13 + 12) = 40
}
```

x = 10
*y = b
**z = a



Ans: (c) 40

Teacher's Signature .....

45. Program A;

```
X: procedure (int x)
    {
        int y;
        y = x + 1;
        return;
    }
    main( )
    {
        X (3);
        print x, y;
    }
```

The above program has the output, when x, y called by value

(a) 3, 4

(b) 4, 3

(c) undefined

(d) none of the above

(c) 45) Program A;

```
x: Procedure (int x)
{
    int y;
    y = x+1;
    return;
}
main()
{
    X(3);
    printf(x, y);
}
```

Call by Value

```
{X(3);
 y = 3+1 = 4
```

Ans - (c) Undefined

46. What would be the output of the following program?

```
Unsigned int n=20,i;
main( )
{
        for(i=0;i<n;i--)
        printf("b");
}
```

(a) infinite number of b's
(b) 20 b's are printed
(c) 65537 b's are printed
(d) none

(a) 46 ⇒

```
int n = 20; i;
main()
{ for (i=0; i<n; i--)
    print (b)
}
```

n = 20

i = 0 < 20 → b

i = -1 < 20 → False

Only one (b) printed

Ans → (a) → None

47. Which of the following is an advantage of static scoping?

(a) Readability
(b) Predictability
(c) Type checking
(d) All of the above

(d) 47→ An advantage of static scoping }

(a) Readibility
(b) Predictability
(c) Type checking
(d) All

48. What is the output printed by following program?

```
int main ()
{
    if ( 0 < 7 < 5)
            printf ( "I am printed") ;
    else
            printf ( "I am not printed ") ;
}
```

(a) I am printed

(b) I am not printed

(c) Compiler error

(d) None of these

(a) 488

```
int main()
{
    if (0 < 7 < 5)
        printf("I am printed");
    else
        printf("I am not printed");
}
```

Ans | (a) → I am printed

49.
```
void main( )
    {
      int A[10][20];
      int i,j , *p;
      p=&A[0][0];
      for(i=0;i<20;i++)
      for(j=0;j<10;j++)
      *(p+20*j+i)=i*20+j;
      printf("%d", A[5][10] – A[5][9]);
}
```

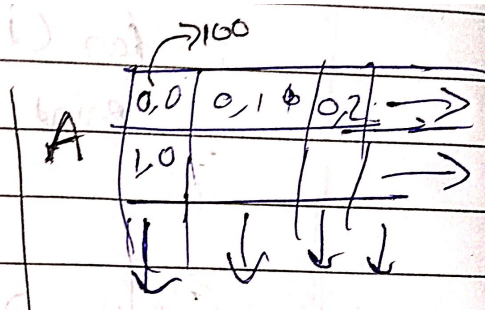
value printed by the program is

(a) 1 (b) 10

(c) 20 (d) 100

(c) 49. void main()
{ int A[10][20];
int i, j, *p;
p = &A[0][0];
for (i=0; i<20; i++)
for (j=0; j<10; j++)
*(p + 20*j + i) = i*20+j;
printf (A[5][10] - A[5][9]);
}
205 - 185 = 5

Ans → (c) 20



p = 100

*(100 + 0 + 0) = 0

*(100 + 20 x 10 + 5) = 20x5+10

*(p + 205) = 110

*(p + 20 x 9 + 5) = 20x5+9

*(p + 185) = 109

50. Which of the following statements is false about recursion?

(a) Simplifies the logic

(b) Every recursive program can have an alternative iterative program

(c) Generally recursive version of programs take less space over iterative counter parts

(d) None of these

(c) 50⇒ False about recursion —

(a) Simplified the logic
(b) Every recursive program can have an alternative iterative program
(c) Generally recursive version of programs take less space over iterative counter parts.
(d) None

Ans → (c) Recursion takes more space.