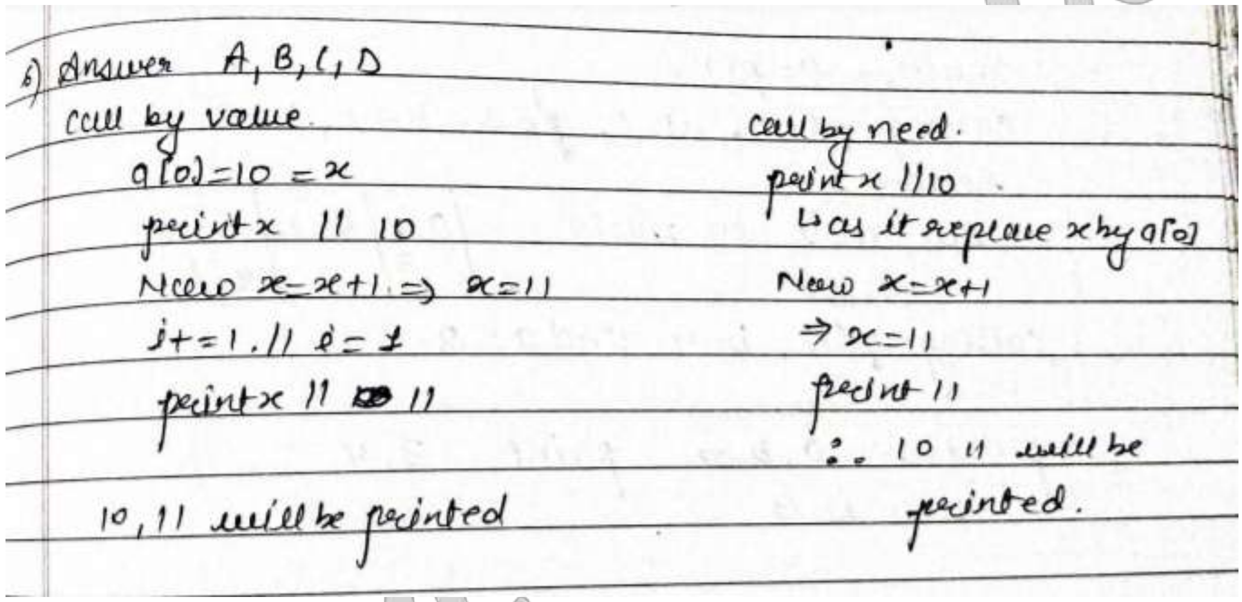
(B) If parameter passing mechanism is “call by name”, then the output of the given code: 10, 11.

(C) If parameter passing mechanism is “call by need”, then the output of the given code: 10, 11.

(D) If parameter passing mechanism is “call by value”, then the output of the given code: 10, 11.



6) Answer A, B, C, D

call by value

a[0]=10 = x

print x // 10

New x=x+1 ⇒ x=11

i+=1, // i = 1

print x // 11

10, 11 will be printed

call by need.

print x //10

↳ as it replace x by a[0]

New x=x+1

⇒ x=11

print 11

∴ 10 11 will be printed.

Q7. **[MSQ]**

Consider the following code

```
int n = 1;
void display (int x){
print  x + n;
}
void increment(){
n = n + 2;
print n ;
}
void main(){
int n =200;
display(7);
n = 50;
increment();
print n;
}
```

Which of the following statement is/are true?

(A) If the Static Scoping is used then printed output is: **8 52 50**

(B) If the Static Scoping is used then printed output is: **8 3 50**

(C) If the Dynamic Scoping is used then printed output is: **207 52 50**

(D) If the Dynamic Scoping is used then printed output is: **207 52 52**

Answer: b, d

Solution : **Static scoping**:

n=200→(local n of main)

display (7) will be called from main

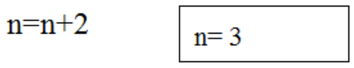
void display (7)

print x+n

(global) n = 8, and n=50 in main

increatment( ); will be called.

void increament: n=n+2

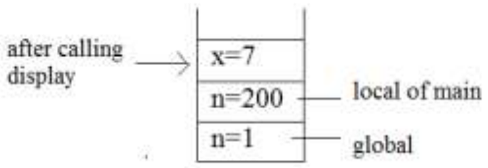


print n;

In main(), when print n will be execute so it will print its local n.

print n=50. Hence output: 8 3 5 0

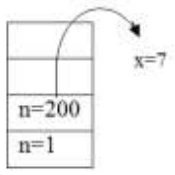
Dynamic scoping: hence create all variables in stack.



so display (x). It will print x+n

at stack top x=7 and n=200. So, x + n = 7+200=207

It prints 207.



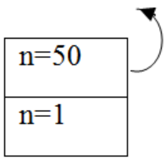
It will be pop from stack as display ( ) will terminate .

Now n=50

on calling increament ( )

n= n+2 → stack contain 50 on top.

52 =50+2



n= 52 will be printed .

n=52
n=1

stack top contains 52 hence in main again n=52 will be print.
Printed output : 207, 52, 52

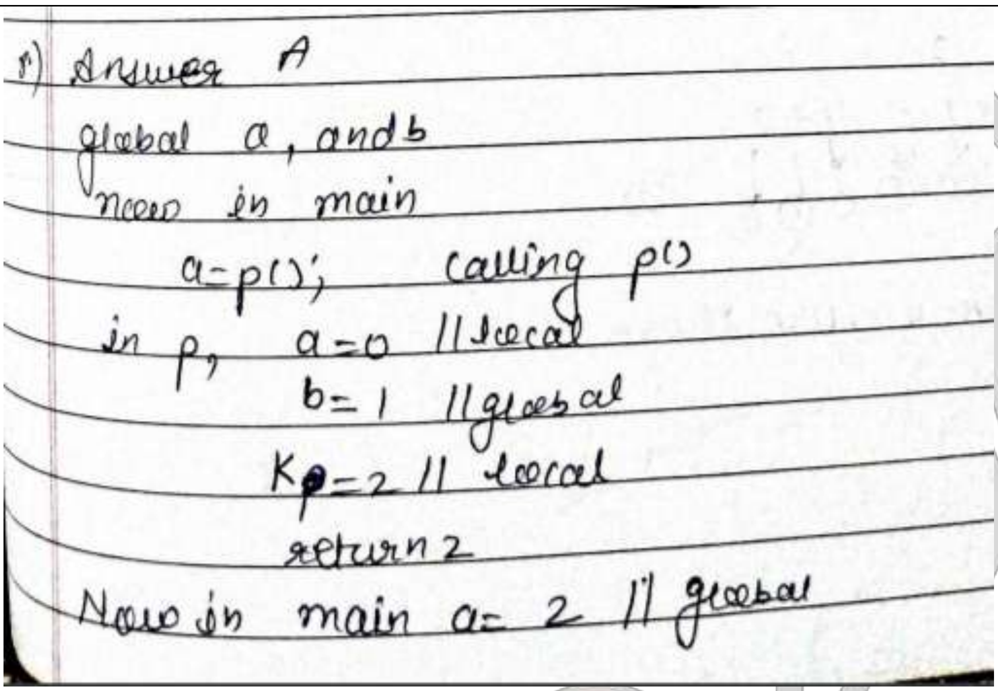
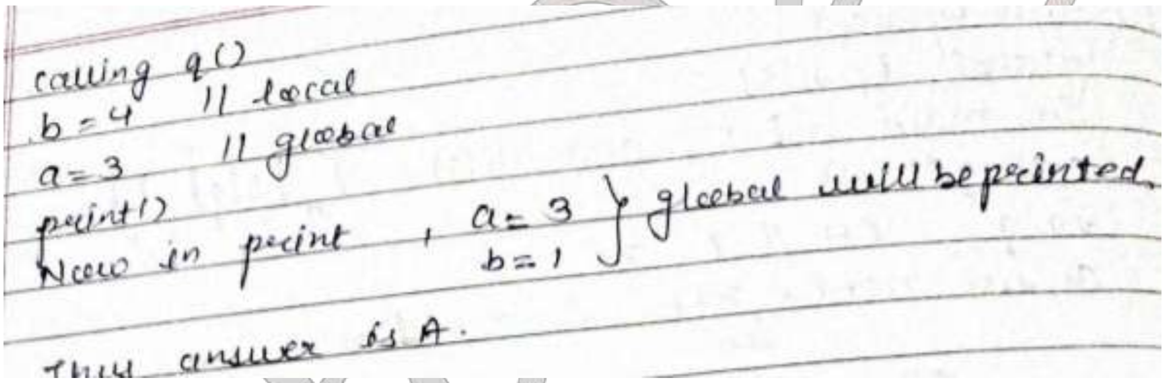
For Next Two Question Consider the following block of C code:

```
#include <stdio.h>
   int a, b;
   int p(void)
   {
       int a, k;
       a =0; b=1; k=2;
       return k;
   }
   void print( void)
   {
       printf("%d\n %d\n", a, b);
   }
   void q(void)
   {
       int b;
       a = 3; b=4;
       print();
   }
   Void main()
   {
       a=p();
       q();
   }
```

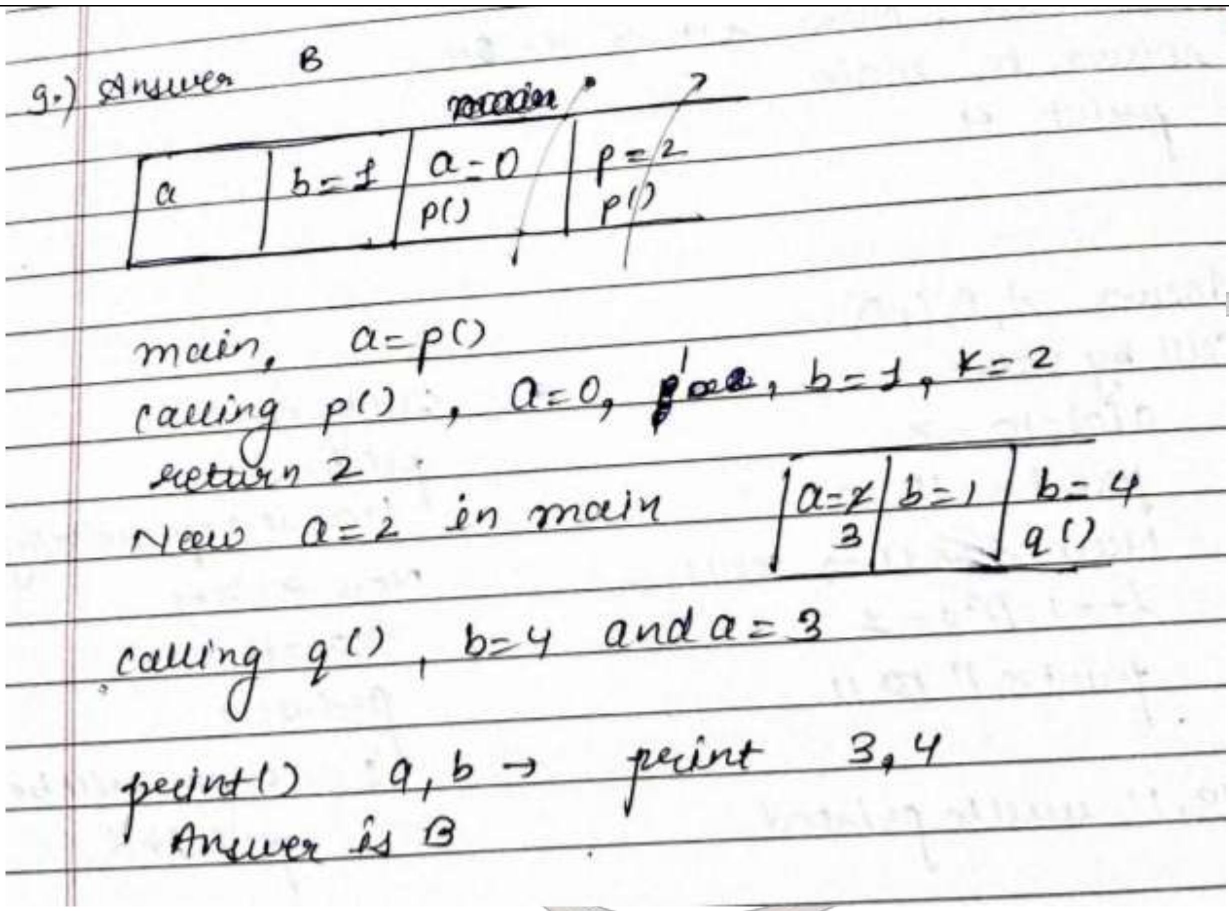
Q8. What values will be printed, when the program is parsed using Lexical (Static) scope?

(A) 3,1 (B) 3,4

(C) 4, 1 (D)none

	 1) Answer A
	global a, and b
	Now in main
	a=p(); calling p()
	in p, a=0 //local
	b=1 //global
	K~~p~~=2 // local
	return 2
	Now in main a= 2 // global
	 calling q()
	b=4 // local
	a=3 // global
	print()
	Now in print , a=3 b=1 } global will be printed.
	This answer is A.

Q9.	What values will be printed, when the program is parsed using Dynamic scope? (A) 3,1 (B) 3,4 (C) 4, 1 (D)none



9.) Answer B

~~main~~

a	b=1	a=0 p()	p=2 p()

main, a=p()

calling p(), a=0, ~~foo~~, b=1, k=2

return 2

Now a=2 in main

a=~~2~~ 3	b=1	b=4 q()

calling q(), b=4 and a=3

print() a, b → print 3, 4

Answer is B

Q10. **[MSQ]**

Consider the following code

```
int a = 2;
void foo(int b) {
b = b * a;
a = a - b;
}
void main(){
{
int a = 10;
foo(a);
print a;
}
```

Which of the following statement is / are true?

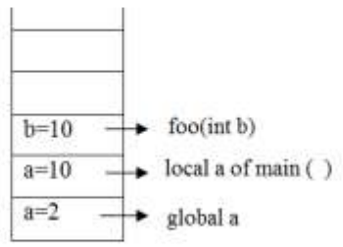
(A) The output of the above code under call-by-value and lexical scope is 10.

(B) The output of the above code under call-by-value and dynamic scope is -90.

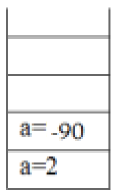
(C) The output of the above code under call-by-reference and dynamic scope is 0.

(D) The output of the above code under call-by-reference and lexical scope is 20.

Answer: a, b, c, d
Solution:
(a) **call by value and lexical scope**
Given function foo(int b) does not return anything and its call by value hence changes in formal do not reflect in actual and static scoping also does not do any changes hence print a in void main will 10 .
(b) **Call by value and dynamic scoping**
foo(a) ; from main
a=10 (actual)
this void foo(int b)
b=10 (formal)
No changes due to call by value but through dynamic scoping stack will be created.

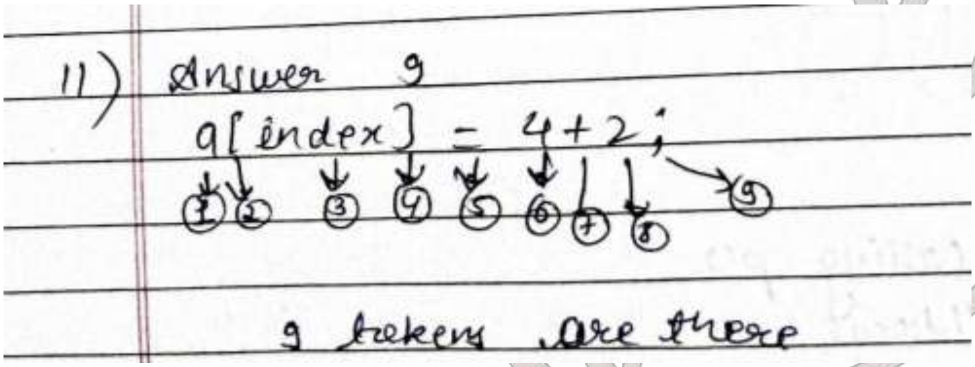


Now stack top b=10 and a=10. Hence a and b will be 10.
b = b*a
100= 10*10
a=a-b
-90 = 10-100
a=-90
after the termination of foo ( ) , now stack contain



Hence, print a; will print -90.
(c) **Call by reference dynamic scope**
foo(a) from main
a and b will share same memory location in call by reference.
void foo(b) $\Rightarrow$ b=b*a = 100=10*10
a=a-b =100-100=0
print a; then a=0
(d) **Call by reference and static scope**
foo(a);

	a=10 for main( ) void foo(b) b and a of main share location b=b*aa refers to global as it don't have local a of foo. 20=10*2 a=a-b =18 =2-20 a(global) In main a will be printed and that a is local of main which share memory location with b hence y = b=20 a(local)=20 hence 20 will be print.

Q11.	How many Tokens are there in the following statement? ________
	a[index] = 4 + 2;
	
	11) Answer 9
	a[index] = 4+2;
	$\textcircled{1}$ $\textcircled{2}$ $\textcircled{3}$ $\textcircled{4}$ $\textcircled{5}$ $\textcircled{6}$ $\textcircled{7}$ $\textcircled{8}$ $\textcircled{9}$
	9 tokens are there

Q12.	The number of tokens in the following C statement is ____________
	`int i = 10, j = 0;`
	`printf ("i = %d, &i = %x", i, &i);`