*	Pointer to an array -
	Pointer to an array is a pointer that points to
10 Paris	Pointer to an array is a pointer that points to first element of an array.
Code-	#include < bits stdc ++·h>
Lette Akarin	Using namespace std;
	int main () of microlina planta entrance
	int nums[]={1,2,3};
	int (*ptx)[3] = &nums //pointer to array of 3 integers
Lat & Comment	cout << (*ptn)[1]<< end];
	int * ptx2=nums; // pointer to 1st index of nums
	cout
ker (1999) in the second	}
× Neter	CAT 3 i Ciril C
0/P=	wing it is the third house the series of the
1. 46. 15.	13 miles motor superings to Lie in the
*	Array of pointers -
	in Contracting to
	An array of pointers is an array where each element is a pointer to a memory location.
	a portato lo a merriorgi lo action.
Code-	#include < bits Stdc++·h>
	Using namespare std;
1	
	int main(){
14111	int nums[5]= &1,2,3,4,54; O/P=1
3 1 2 2 2 2	int * ovr.[5];
M	for (int i=0; i<5; i++){
	artij=&numstij
	cout << *arrtij << end!;
	44
	4.5

Note > While making pointer to an array, int nums[3] = {1,2,3}; int (*ptx)[3] = & nums; the parentheses around *ptr is necessary because the de-reference operator * has lower pricedence than the array subscript operator []. What happens when an Array is passed to a function #include < bits/stdc++.h> using namespace std; > This is pointer int solve (int aux [7, int size) { cout << "size of are inside solve: "<< size of (are) (cend); int main () { int over.[] = {1,2,3,4}; cout << "size of arr inside main: "<< size of (arr) << endl; solve (auc, 4); OIP= size of arr inside main: 16 Size of arm inside solve: 8 / Pointer address is architecture dependent Note- When an array is passed to a function, not the whole array is passed. Actually, the base address of array is passed to the function. We can also write int aux [] in solve function as int xavr.

The second second								
Q1-	mo	in()	15 mm (11.5)	solve (in	it *aur, int	size) {		
	S							
	int	ave[]={1,	2,33;	11 P	sint			
	Solve (avr, 3); avr → 104 &avr → 216							
	1/ Printing standary of a standard no							
		vuz -> 104		LOW - Cap How Kills 163				
	<u>&c</u>	104 -	and in	int our]={1,2,3};			
	5			104	<u></u>			
				0902 1	2 3			
	Pag	151	rue	1 5	216			
Lt	SW P	1121	- (rost	Int *a	ver > 104			
			1.47			100		
	<u>q</u>	- 1	29	4	J) 5	baile 11 to 1		
(D) (1)		24-8	- 1-4	工程式	4. 4.	Collingua		
92-	mai	h()	2444	-14450	olve (int aux	[], int size)		
	9	ww.[]= {10			-1×××			
	int o	k(avz+1)+=	5;					
			,	7				
		re (our, 3)	, ,	}				
	Solv	re (arr, 3)		<u></u>				
(p. 5	Solv	re (arr, 3)		<u></u>	. p. ini 💠			
(F =	Solv 1/Pa	re (aur, 3) sint/cura		0 25;30 ; 28	LD ini s			
(+=	Solv	re (arr, 3)	y ~> 1	<u></u>	LD ini s			
(F =	Solv 1/Pa	re (aur, 3) sint/cura	y ~> 1	0 25 30	operation			
/ H=	Solv 1/Pa	re (aur, 3) sint/cura	y ~> 1	0 25 30	LD ini s			
(F =	Solv 1/Pa	re (aur, 3) sint/cura	1 2 1	0 25 30	operation			
	Solv 1/Pa	re (aur, 3) sint/cura	y = 1	0 25 30	Ap ini s			
	Solv 1/Pa	re (aur, 3) sint/cura	y = 1	0 25 30	operation			
	Solv 1/Pa	re (avr, 3)		0 25 30	Ap ini s			
/ K=	Solv 1/Pa	re (avr, 3)		0 25 30	Ap ini s			
	Solv 1/Pa	re (avr, 3)		0 25 30	Ap ini s			

```
Pointer to Pointer ->
      *
           int a = 5;
           int*p=&a;
           int ** p1 = & p; //double Pointer
           int ** * p2 = &p1; // triple Pointer
int ** * * p3 = &p2; // multi Pointer
         104
                  204
                                    (304)K
                                             1404
                            204)
                   7104
                            ρ<u>1</u>
                                    ρž
 Canbe
        \ a
                                    *P3
                            *P2
accessed by
                  *P1
         * P
                 **P2
                           **P3
          **p1
                  ***P3
        * ** P2
                                 object love to
        * * * * P3
            int a = 5; = 0/104. 104.
     0-
            int *p= &a; 5 k
                                    104K
                                              204
            int **q = &p;
           // Print ->
         a:5 p:104
                                9; 204
        &a: 104 &p:204
                                &q: 304
                   *p:5
                                *9: 104
                                **q: 5
```

