H/w: write a c program to search a number in the array, if exists display the number and its index.

```
        a[j] = a[k]
            a[k] = temp;
        }
    }
}
```

VLA

```
int n;
int arr[n];
scanf("%d", &n);
```

Procedure for creating VLA.

Multidimensional arrays

2D array → matrix

Syntax:

```
datatype  array_name [size1][size2];
```

ex) int a[3][3]; → represents matrix.

Memory allocation:

Row major order

(0,0)	1	2	3
(1,0)	4	5	6
(2,0)	7	8	9

col n " "

1 2 3 4 5 6 7 8 9

1 4 7 2 5 8 3 6 9.

Initialization:-

```
int stud[4][2] = { 1,2,3,4,5,6,7,8 };
```

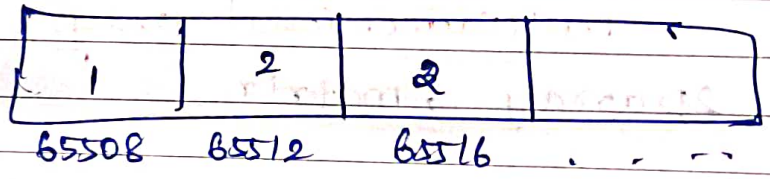
readability

```
{ {1,2}, {2,2}, {3,2}, {4,2} };
```

```
int a[2][3] = { 1,2,3,4,5,6 }  ✓
int a[ ][3] = { 1,2,3,4,5,6 }  ✓
int a[2][ ] = { 1,2,3,4,5,6 }  ✗
int a[ ][ ] = { 1,2,3,4,5,6 }  ✗
```

missing elements are initialized to zero or garbage value.

$$[4][2] = \begin{bmatrix} 00 & 01 \\ 1 & 2 \\ 2 & 2 \\ 3 & 2 \\ 4 & 2 \end{bmatrix}$$



↓ 2D is called array of array

✓ int m[3][5] = {{0}, {0}, {0}}

```
int a[3][3];
  for(i=0 ; i<3; i++)
   {
     for(j=0; j<3; j++)
      {
        Printf (" Enter numbers");
        Scanf (" %d , &a[i][j] ;
        Printf (" %d, a[i][j]);
      }
   }
```

separate for loop for input & output

Program

Write a C program to obtain transpose of a 3×5 matrix.

```
main()
{
    int mat[3][5] = {{1,2,3,4,5},
                     {6,7,8,9,10},
                     {11,12,13,14,15}};

    int Tran[5][3];
    int i,j;
    for(i=0; i<3; i++)
        {
        for(j=0; j<5; j++)
            {
            Tran[j][i] = mat[i][j];
        } }

    for(i=0; i<5; i++)
    {
        for (j=0; j<3; j++)
            Printf("%d \t", Trans[i][j]);
            Printf("\n");

        }

}.
```

24, 35

Arrays

```
int arr[5]; = {1,2} → 2nd
```

1st

3rd

{1,2,[4]=4}

4th arr[2]=arr[0]

arr[1]

arr[3]=317x2

```
{
for(i=0; i<=4; i++)
  Printf("%d \n", arr[i]);
}
```

O/P:

	2	3rd	4th
all garbage value	1	1	1
	2	2	2
	0	0	3
	0	0	4
	0	4	4

```
int a[5] = : 20
float a[5] = 20
double a[5] = 40 bytes
Char: a[1] = 5 "
```

```
Char name [10] = "WELL DONE";
```

Name(0)	'W'
(1)	'E'
(2)	'L'
(3)	'L'
(4)	' '
(5)	'D'
(6)	'O'
(7)	'N'
(8)	'E'
(9)	'\0' → null Char

5, 9, 10, 11, 12, 14, 21, 26, 28, 30, 31, 33, 35, 39, 40, 41, 42, 44, 45, 47, 48, 50, 52, 55, 58, 62, 65, 66, 69, 73, 74, 75, 76, 81

Classmate Date

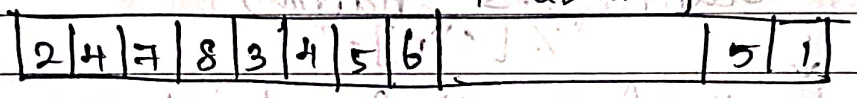
Bunch-4

3D arrays → array of 2D arrays.

syntax

datatype array name [1][2][3];

int arr [3][4][2] = {
{ {2,4}, {7,8}, {3,4}, {5,6} },
{ {7,6}, {3,4}, {5,3}, {2,3} },
{ {8,9}, {7,2}, {3,4}, {5,1} } };



arr[2][3][1] ?

Strings. 1 Dimensional

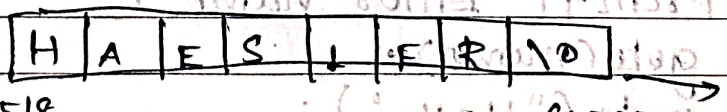
↳ group of chars (Char array).

1D array of chars terminated by a '\0'.

ex: Char name[] = { 'H', 'A', 'E', 'S', 'L', 'E', 'R', '\0' };

missed

teach a[i], &a[i] → missed.



for functions working on string.

Char name[] = "HAESLER";