H/w: write a C program to generate all combinations of 1,2 and 3 using for loop.

2)

1

2 3

4 5 6

7 8 9 10.

Chapter 2

Arrays

collection of similar data types

Array of 'char's → String

Array of int/float → array.

ordered sequence of homogenous values.

Syntax

decl: datatype name [size];

↓

dimension

ex) int marks [30];

marks[0], marks[1] .... marks[29].

↓ Starts with zero. ↘ subscript.

ex)

```
int avg, sum=0
int i;
int marks [30];
for(i=0; i<=29; i++)
{
    Printf ("enter marks");
    Scanf ("%d", &marks[i]);
                    → store
```

Invalid

double x[];

int N;

double x[N]; ✓ allowed

```
for (i=0; i<=29; i++)
        Sum = Sum + marks[i];
    avg = sum/30;
    printf ("Avg", avg);
}
```

Initialization: auto: `int arr[5] = {0};`

`= {1,2}`

`int num[6] = {2,4,12,5,45,5};` `= {1,2,[4]=4}`

missed `int n[] = {2,4,12,5,45,5}`

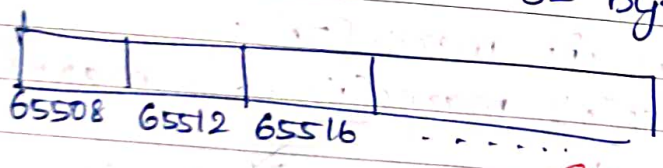
`float press[] = {12.3, 34.2, -23.4, -11.3};`

↓ just declare, static, all elements would be set to zero.

→ if it not initialise, it takes garbage values.

```
int arr[8];
```

↓ reserve 32 bytes



trap!:

```
int num[10]
    for(i=0; i<=99; i++){
        num[i] = i;
    }
```

↓ placed in memory loc, outside the array, if it is useful info it is lost

if garbage → no problem.

No error msg → programmer's duty

Lower & upper bound

scanf("%d%d", &arr[8], &arr[9]);

arr[5] = arr[1] + arr[3]

" = 3 + 7 % 2

Write a program that interchanges elements at odd position with elements at even position in an array of 10 elements.

```
int num[] = {12,4,5,1,9,13,11,19,54,34};
int i,t;
for (i=0; i<=9; i+=2)
{
    t = num[i];
    num[i] = num[i+1];
    num[i+1] = t;
}
for (i=0; i<=9; i++)
    printf("%d/t", num[i]);
}
```

H/w. write a program to copy the contents of a 5 element integer array into another array in reverse order.

```
int arr1[5], arr2[5], i, j;
Printf ("enter 5 elements");
for(i=0; i<=4; i++)
    scanf ("%d", &arr1[i]);
for(j=4, i=0; i<=4; i++, j--)
    arr2[j] = arr1[i];
Printf ("elements in reverse order")
for (j=0; i<=4; i++)
    printf ("%d\t", arr2[i]);
{
```

```
int a(25);                    X
int size = 10, b[size];       X
```

```
int num[5];   → array size
num[5] = 11;  → particular element
```

Array overflow → Leads to System malfunction

Array underflow → filled with garbage values or zero.

Subscript size → value overwritten.

Prg 3:

Sorting n numbers

```
Void main()
{
    int n, a[120];
Printf(" Enter no: of terms: ");
scanf("%d", &n);
for(i=0; i<n; i++)
{
    Printf(" Enter the array ele");
    scanf("%d", &a[i]);
}
for (j=0; j<n; j++)
{
    for (k= j+1; k<n; k++)
    {
        if a[j] > a[k]
        {
            temp = a[j]
```

45, 23, 1, 17, 35
j k
23, 45, 1, 17, 35

1, 45, 23, 17, 35
j k
1, 23, 45, 17, 35
j k
1, 17, 45, 23, 35

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} };
```