arr | 10 | 20 | 20 | 40 | 50 |
(labels above: 5̶0̶ 4̶0̶ __ 2̶0̶ 1̶0̶)
indices: 0 1 2 3 4
i i i j j

arr | 10 | 20 | 30 | 40 | 50 | 60 |
(labels above: 6̶0̶ 5̶0̶ 4̶0̶ 3̶0̶ 2̶0̶ 1̶0̶)
indices: 0 1 2 3 4 5
i i i j j j
int a[5]

$$\text{for } (i=0,\ j=n-1;\ \boxed{i<j}\ ;\ i++,\ j--)$$

```
{
    temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}
```

Searching :→   Linear   Binary

Linear →

#define TRUE 1

n | 9

arr | 5 | 3 | 7 | 8 | 6 | (9) | 11 | 4 | 1 |
indices: 0 1 2 3 4 5 6 7 8
i i i i i (i) TRUE

num (9)

```
int flag = FALSE;   // TRUE
for (i=0; i<n; i++)
{
    if (num == arr[i])
    {
        flag = TRUE;
        break;
    }
}
if (flag == TRUE)
    number found
else
    number not found.
```

Binary Search :→

Sorted Array

n $\boxed{8}$

arr | 5 | 7 | 10 | 15 | 18 | 21 | 25 | 28 |
index: 0, 1, 2, 3, 4, 5, 6, 7

low   high   mid   high

high → low

high → 1
low → 2

num $\boxed{8}$

```
flag = FALSE TRUE
low = 0 ;
high = n-1;
while ( low <= high)
{
    mid = (low + high)/2;
 →  if (num == arr[mid])
    {
        flag = TRUE;
        break.
    }
 →  if ( num < arr[mid])
        high = mid-1;
    else
        low = mid+1;
}
```

if (flag == TRUE)
    number found at mid

else
    number not found

Sorting ⇒

① Selection :-

arr | 5 1 | 2 5 | 4 | 3 | 1 2 |
index: 0, 1, 2, 3, 4

if (arr[i] > arr[j])
    swap

pass-1
arr | 5 1 | 1 5 | 4 | 3 | 1 2 |
index: 0, 1, 2, 3, 4

i = 0 | j = 1 - 4

pass-2
arr | 1 | 5 2 | 5 4 | 1 4 | 1 3 |
index: 0, 1, 2, 3, 4

i = 1 | j = 2 - 4

pass-3
arr | 1 | 2 | 1 5 3 | 5 4 | 1 4 |
index: 0, 1, 2, 3, 4

i = 2 | j = 3 - 4

arr | 1 | 2 | 3 | 5 4 | 5 4 |

i = 3 | j = 4 - 4

arr | 1 | 2 | 3 | $\cancel{4}$ 5 | $\cancel{5}$ 4 |

0 1 2 3 4

i=3 | j=4-4

i j α

arr | 1 | 2 | 3 | 4 | 5 | ← Sorted Array

i = 0 to 3
―――――――――
j = i+1 to 4

```
for( i=0 ;  i < n-1 ;  i++)
{
        for (j = i+1 ;  j < n; j++)
        {
                if (arr[i] > arr[j])
                {
                        temp = arr[i];
                        arr[i] = arr[j];
                        arr[j] = temp;
                }
        }
}
```

```c
#include<stdio.h>
#define FALSE 0
#define TRUE 1
int main()
{
    int a[10];
    int n,i,j,temp;
    printf("Enter number of elements:");
    scanf("%d",&n);
    //input
    for(i=0;i<n;i++)
    {
        printf("Enter value of %d element:",i+1);
        scanf("%d",&a[i]);
    }

    //selection sort
    for(i=0;i<n-1;i++)
    {
        for(j=i+1 ; j<n ; j++)
        {
            if(a[i]>a[j])
            {
                temp = a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    //output
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    return 0;
}
```

Bubble :—

Pass-1

i=1

arr | 5 | 1 | 7 | 4 | 5 |
     | 0 | 1 | 2 | 3 | 4 |

j+1   if(arr[j] > arr[j+1])
          swap

j = 0 to 3

i=2

arr | 3 | 2 | 4 | 4 | 5 |
     | 0 | 1 | 2 | 3 | 4 |

j = 0 to 2

i=3

arr | 2 | 7 | 3 | 4 | 5 |
     | 0 | 1 | 2 | 3 | 4 |

j = 0 to 1

i=4

arr | 7 | 1 | 3 | 4 | 5 |
     | 0 | 1 | 2 | 3 | 4 |

j = 0 to 0

arr | 1 | 2 | 3 | 4 | 5 | ← Sorted

$$i = 1 \text{ to } 4$$
$$j = 0 \text{ to } < n-i$$

```c
#include<stdio.h>
#define FALSE 0
#define TRUE 1
int main()
{
    int a[10];
    int n,i,j,temp;
    printf("Enter number of elements:");
    scanf("%d",&n);
    //input
    for(i=0;i<n;i++)
    {
        printf("Enter value of %d element:",i+1);
        scanf("%d",&a[i]);
    }

    //bubble sort
    for(i=1;i<n;i++)
    {
        for(j=0 ; j<n-i ; j++)
```

```c
        {
            if(a[j]>a[j+1])
            {
                temp = a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    //output
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    return 0;
}
```
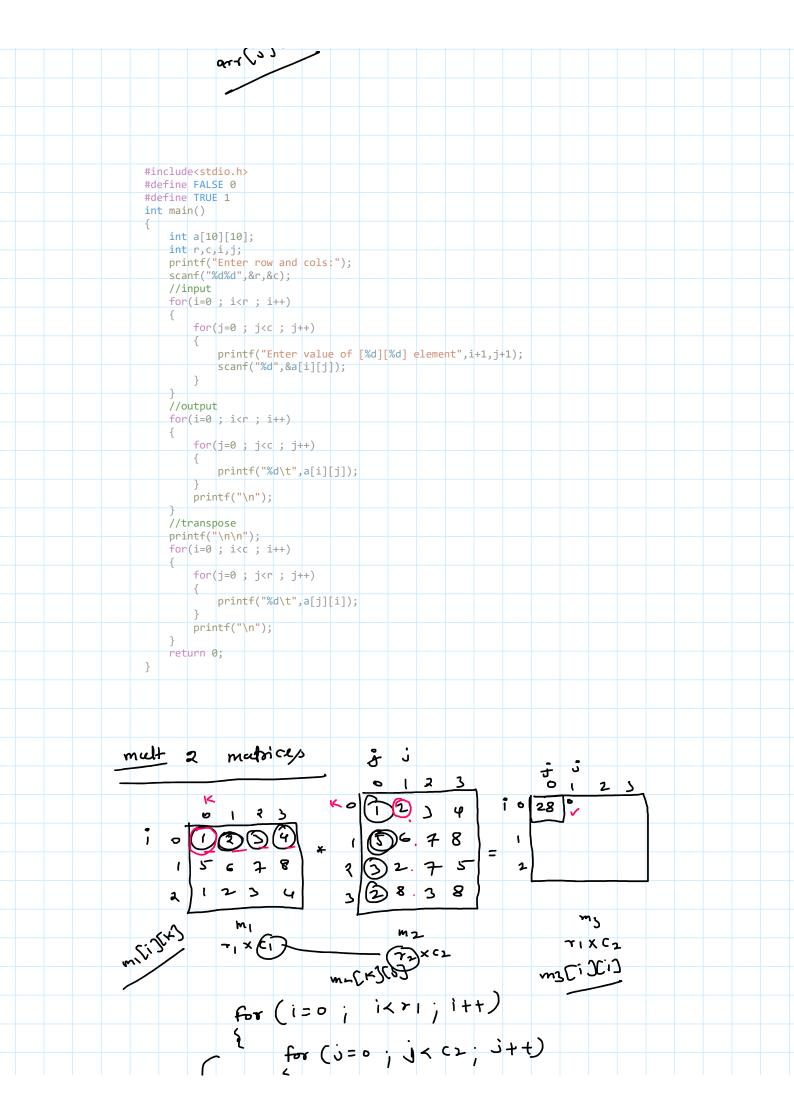
Insertion :→

num $\boxed{4}$

n $\boxed{6}$ arr $\boxed{5}_0$

arr $\boxed{1\ \ 2\ \ 3\ \ 45\ 5\ \ 6}$

$_{-1}$ 0 1 2 3 4 5
        j j i i

```c
for (i = 1 ; i < n ; i++)
{
    num = arr[i];
    for (j = i-1 ; j >= 0 && arr[j] > num ; j--)
    {
        arr[j+1] = arr[j];
    }
    arr[j+1] = num;
}
```

2-D Array :→

Row   col
 ↓   ↓

int arr [3] [4];

      0   1   2   3

   0
   1   10
   2

       arr

arr[1][1] = 10;

int arr[3][4] = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };

	0	1	2	3
0	10	20	30	40
1	50	60	70	80
2	90	0	0	0

arr

$$int\ arr[3][4] = \{\ \{10,20,30\},\ \{40,50,60\},\ \{70,80,90\}\ \};$$

100

10	20	30	0
40	50	60	0
70	80	90	0

$$int\ arr[5] = \{1,2,3,4,5,6\};\quad \longleftarrow\ \text{Compile time Error}$$

```c
#include<stdio.h>
#define FALSE 0
#define TRUE 1
int main()
{
    int a[10][10];
    int r,c,i,j;
    printf("Enter row and cols:");
    scanf("%d%d",&r,&c);
    //input
    for(i=0 ; i<r ; i++)
    {
        for(j=0 ; j<c ; j++)
        {
            printf("Enter value of [%d][%d] element",i+1,j+1);
            scanf("%d",&a[i][j]);
        }
    }
    //output
    for(i=0 ; i<r ; i++)
    {
        for(j=0 ; j<c ; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

col   c

i

i	1	2	3	4	\n
	5	6	7	8	
	9	10	11	12	

output
Trans

1	5	9
2	6	10
3	7	11
4	8	12

r   row

arr[j][i]

arr(u)

```c
#include<stdio.h>
#define FALSE 0
#define TRUE 1
int main()
{
    int a[10][10];
    int r,c,i,j;
    printf("Enter row and cols:");
    scanf("%d%d",&r,&c);
    //input
    for(i=0 ; i<r ; i++)
    {
        for(j=0 ; j<c ; j++)
        {
            printf("Enter value of [%d][%d] element",i+1,j+1);
            scanf("%d",&a[i][j]);
        }
    }
    //output
    for(i=0 ; i<r ; i++)
    {
        for(j=0 ; j<c ; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    //transpose
    printf("\n\n");
    for(i=0 ; i<c ; i++)
    {
        for(j=0 ; j<r ; j++)
        {
            printf("%d\t",a[j][i]);
        }
        printf("\n");
    }
    return 0;
}
```

mult 2 matrices

$m_1[i][k]$

m_1
$r_1 \times c_1$

m_2
$r_2 \times c_2$
$m = [k][0]$

m_3
$r_1 \times c_2$
$m_3[i][j]$

for (i=0 ; i<r1 ; i++)
{
   for (j=0 ; j<c2 ; j++)

$$
\begin{array}{l}
\{ \\
\quad for\ (j=0\ ;\ j < c_2\ ;\ j++) \\
\quad \{ \\
\qquad m_3[i][j] = 0; \\
\qquad for(k=0\ ;\ k < c_1\ \overset{or}{r_2}\ ;\ k++) \\
\qquad \{ \\
\qquad\quad m_3[i][j] += m_1[i][k] * m_2[k][j]; \\
\qquad \} \\
\quad \} \\
\}
\end{array}
$$

String :→ char Array.

%c   char   ch = `A' ;

int a;

int a[10] ✓

%s   char   s[10] = " ABCD" ;
              ↑
            string

null char
(EOS)
↓

S	A	B	C	D	\0					
	0	1	2	3	4	5	6	7	8	9

printf ("%s", s); → ABCD

Input :→

char str[20];

scanf ("%s", &str); → Enter ✗
         ↑   → space ✗
        str
       & str

```c
#include<stdio.h>
#define FALSE 0
#define TRUE 1
int main()
{
    char str[20];
    printf("Enter your name:");
    //scanf("%s",str);
    //gets(str);
    scanf("%[0-9]",str);
    printf("Your name = %s",str);
    return 0;
}
```

string.h :→   — strlen
                — strcpy
                — strcat
                — strcmp
                — strrev
                — strstr

strlen →   char s₁[10] = "ABCD";

                int n = strlen (s₁);

                printf ("%d", n); → 4

Strcpy :—   char s₁[10] "ABCD";

                char s₂[10];

                strcpy (s₂ , s₁);
                          ↑   ↑ source
                         dest

char s₁[10];

.
.
.

x   s₁ = "ABCD";   ← Error
BA

                strcpy (s₁, "ABCD");

BA				
A	B	C	D	10

strcat :—

                char s₁[20] = "ABCD";

                char s₂[20] = "xyz";

strcat (s_1, s_2)

$+$

printf ("%s", s_1); → ABCDxyz

---

strcmp :→

s_1 = "Amit"

✓ s_2 = "Amit"

strcmp (s_1, s_2);

$s_1 > s_2$ → 1 ✓

$s_1 < s_2$ → -1

$s_1 == s_2$ → 0

---

strrev :→

char $s_1[10]$ = "ABCD";

strrev (s_1);

printf ("%s", s_1); → DCBA

---

strstr :→

s_1 = "ABCDEF GHIJK";

s_2 = "GHI";

&G

T

NULL

F

strstr (s_1, s_2);

0

Count frequency of digits

49

s_1 = abcd|i|5 8309 rp 3625 0 rsp 2975...

i   i

48-57   48   0 —   int $a[10] = \{0\}$;

1 49   1 —

2 50   2 —

3 51   3 —   a

Handwritten notes:

```
2  50      2  —
3  51      3  —
4  52      4  —
5 [53]     5  —
6  54      6  —
7  55      7  —
8  56      8  —
9  57      9  —
```

a

```
 o o o  o  o o o o  o o
 0 1 2 3 4 5 6 7 8 9
   _
```

(1) ← I − 48
int   49

char → int
digit

(5) ← 53 − 48

if (s₁[i] >= 48 && s₁[i] <= 57)
{
    a[s₁[i] − 48]++
}

```c
#include<stdio.h>
#define FALSE 0
#define TRUE 1
int main()
{
    char str[100]="dgkjdfgdf2g4dfg78dfg2dfb4d5f7g78g9dfgdfg7dfg8df431dg42dfg7862100012574g1df2g5dfg7df4g";
    int a[10]={0},i;
    for(i=0 ; str[i]!='\0' ; i++)
    {
        if(str[i]>=48 && str[i]<=57)
        {
            a[str[i]-48]++;
        }
    }
    for(i=0;i<10;i++)
    {
        printf("%d->%d\n",i,a[i]);
    }
    return 0;
}
```