

## \* Problem statement 1 [ARRAYS]

Write a Program to take input from a user in one dimensional array. Take 5 input from user. find Max and min values from it.

### Algorithm:

- Start
1. Take Integer: array [5], min, max
2. Print all values of the array
3. Print all values of Array
4. let min = array [0]
5. If  $\min > a[i]$        $\min = a[i]$
7. Repeat step 6 for all values
8. let max = a [0]
9. If  $\max < a[i]$ ,  $\max = a[i]$
10. Repeat step (9) for all values
11. Print min And max
12. Stop.

### \* Program: -

```
#include <stdio.h>
void main ()  
{  
    int a[5]; min, max;  
    printf (" Enter 5 numbers");  
    for (i=0; i<5; i++)  
        scanf ("%d", &a[i]);  
    for (i=0; i<5; i++)  
        printf ("%d\n", a[i]);
```

min = a[0]  
for (i=0; i < 5; i++)

{

if (min > a[i])

min = a[i];

}

printf ("minimum value is : %d\n", min);

max = a[0];

for (i=0; i < 5; i++)

{

if (max < a[i])

max = a[i];

}

printf ("maximum value is : %d ", max);

y

### Output:

enter 5 numbers: 1 2 3 4 5

1

2

3

4

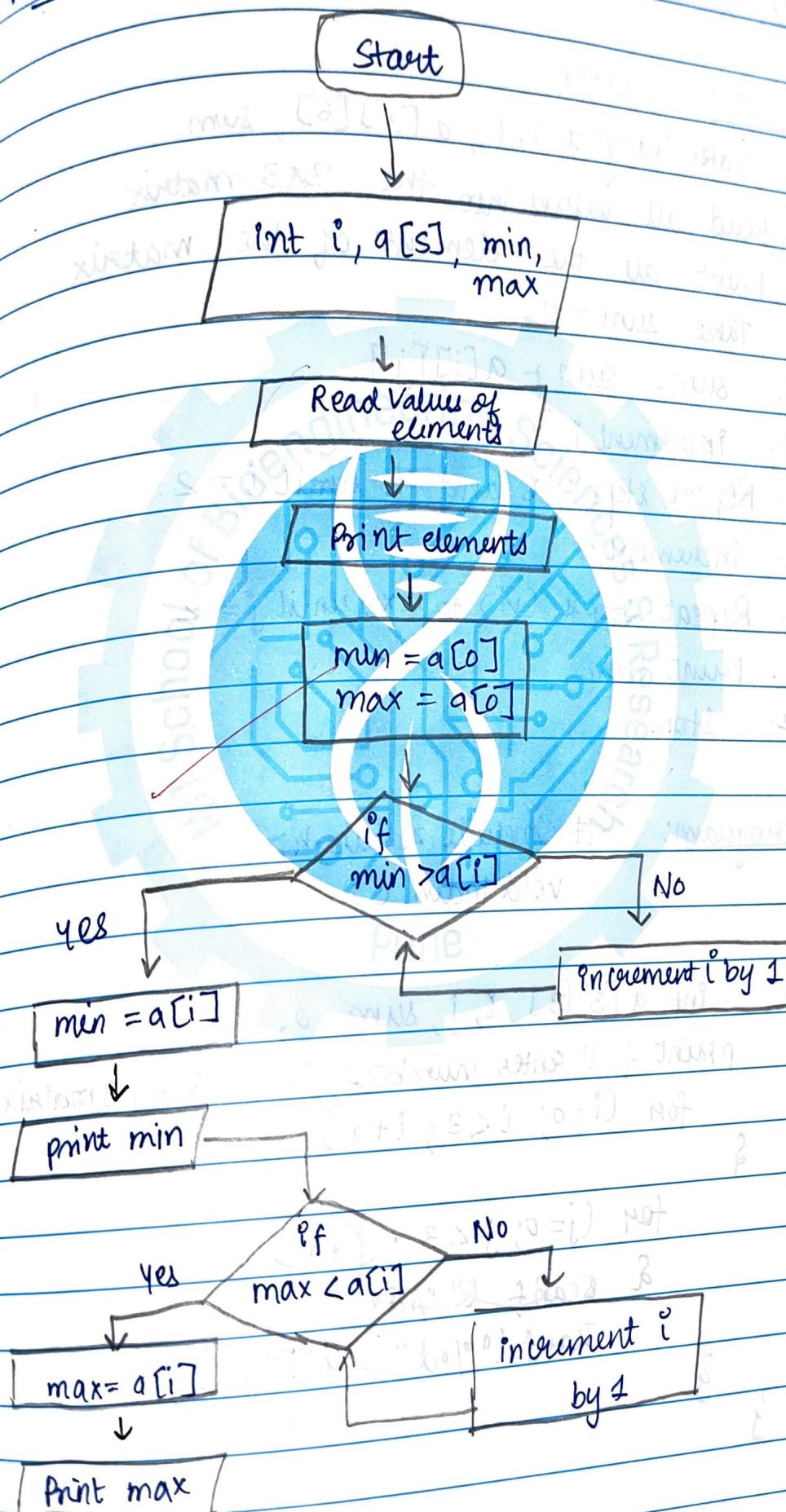
5

minimum value is : 1

maximum value is : 5

(P.T.O)

Flowchart:





# Problem Statement 2: find sum of all elements of  $3 \times 3$  matrix

- \* Algorithm :
  1. Start
  2. Take integers,  $i, j, a[3][3], \text{sum}$
  3. Read all values into the  $3 \times 3$  matrix
  4. Print all the elements of the matrix
  5. Take  $\text{sum} = 0$
  6.  $\text{sum} = \text{sum} + a[i][j]$
  7. Increment  $i$
  8. Repeat steps 6 and 7 until  $i = 2$ .
  9. increment;
  10. Repeat steps (vi) to (ix) until  $j = 2$
  11. Print Sum
  12. Stop.

\* Program: ~~# include <stdio.h>~~  
~~void main ()~~

\$

```
int a[3][3], i, j, sum = 0,  
print f ("enter numbers in a 3 by 3 matrix \n");  
for (i=0; i<3; i++)
```

\$

```
for (j=0; j<3; j++)
```

{ ~~scanf (" %d ", &a[i][j]);~~

```
scanf (" %d ", &a[i][j]);
```

g

g

```
for (i = 0; i < 3; i++)
```

```
{
```

```
    for (j = 0; j < 3; j++)
```

```
{
```

```
        print f ("%d \t", a[i][j]);
```

```
}
```

```
    print f ("\n");
```

```
}
```

```
for (i = 0; i < 3; i++)
```

```
{
```

```
    for (j = 0; j < 3; j++)
```

```
{
```

```
        sum = sum + a[i][j];
```

```
}
```

```
}
```

```
print f ("\n sum of all elements is %d", sum);
```

```
}
```

\* output:-

enter numbers in a 3 by 3 matrix

1 2 3 4 5 6 7 8 9

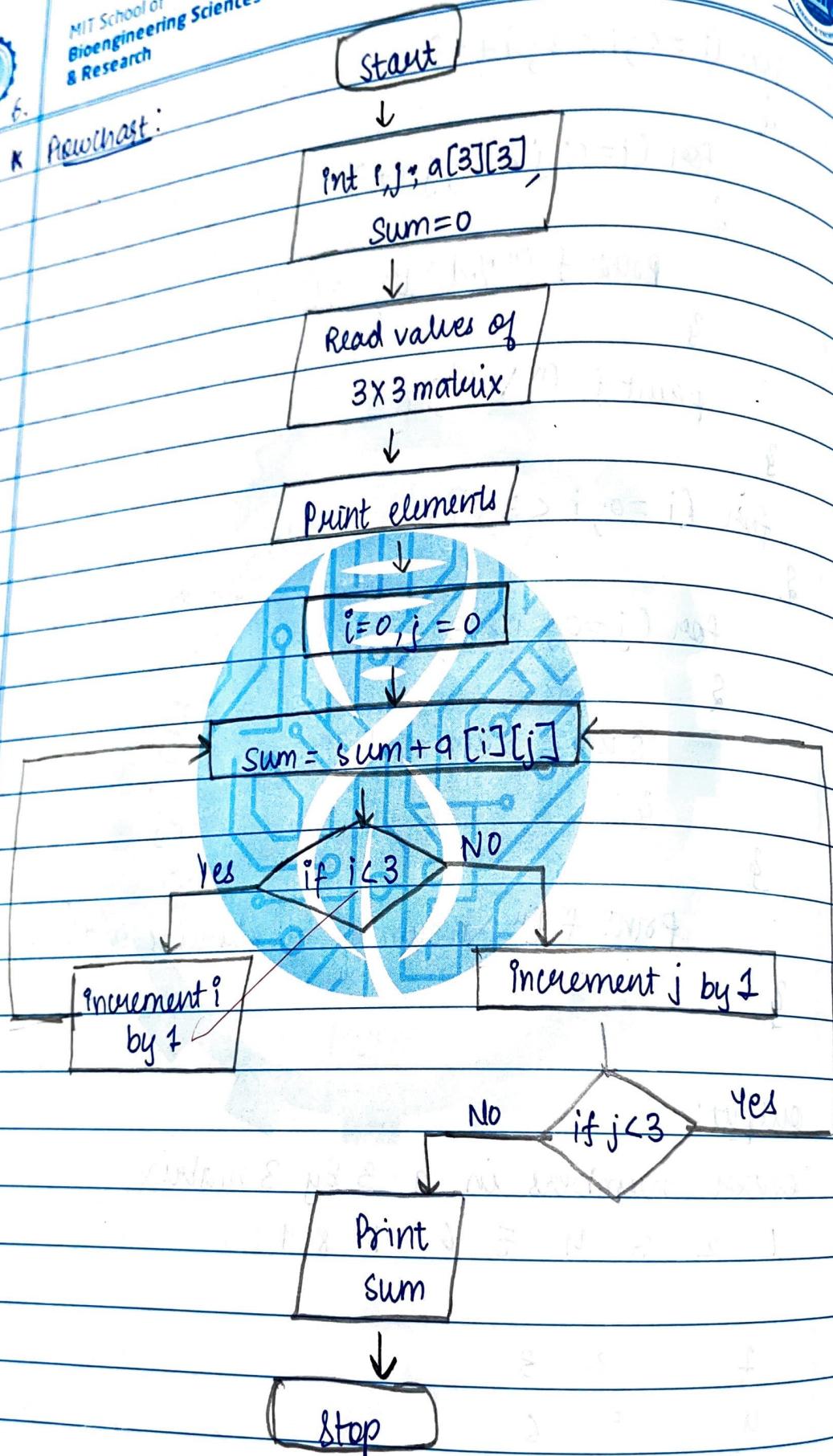
1 2 3

4 5 6

7 8 9

Sum of all numbers is 45.

6. K Flowchart:



\* Problem statement 3: To print Transpose of  $3 \times 3$  matrix.

- \* Algorithm :-
1. Start
  2. Take integers  $i, j, a[3][3]$
  3. Read all the values into  $3 \times 3$  matrix
  4. Print all elements of matrix
  5. Interchange the value of  $i$  and  $j$  for an element.
  6. Repeat Step (5) for all the elements
  7. Print new values of Matrix
  8. Stop.

\* Program: #include <stdio.h>

Void main ()

{

int a[3][3], i, j,

printf ("enter numbers in a 3 by 3 matrix \n");

for (i=0; i<3; i++)

{

    for (j=0; j<3; j++)

{

        scanf ("%d", &a[i][j]);

}

3

    for (i=0; i<3; i++)

{

        for (j=0; j<3; j++)

{

            printf ("%d\t", a[i][j])

2 3



8

print f (" \n ");

3

print f (" transpose of given matrix is \n ");

for (i=0; i<3; i++)

3

for (j=0; j<3; j++)

3

print f (" %d\t", a[j][i]);

3

print f ("\n");

3

3

→ Output:-

enter Numbers in a 3 by 3 matrix

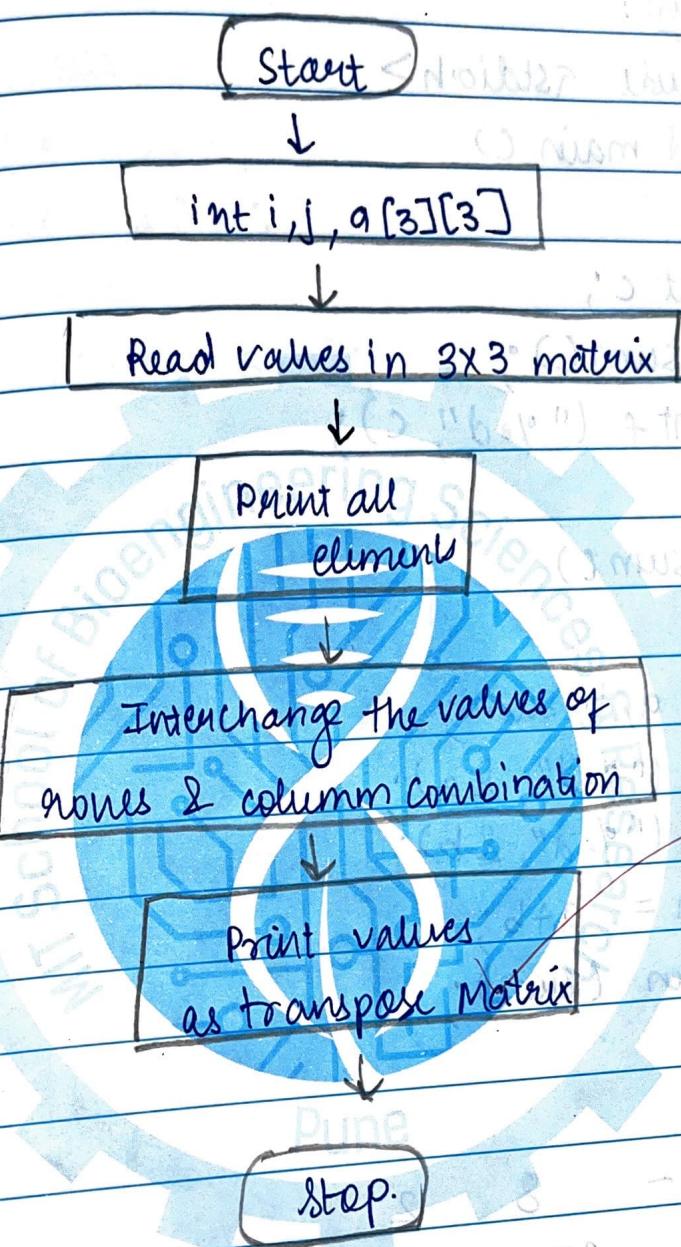
1 2 3 4 5 6 7 8 9

1 2 3  
4 5 6  
7 8 9

given  
Transpose of Matrix is:

1 4 7  
2 5 8  
3 6 9.

Flowchart:



\* Problem Statement 4.. (User Defined Functions)

→ Write a Program to find sum, difference, product and Division of two numbers using User defined functions.



1D.

1) Sum:

Program:

```
# include <stdio.h>
```

```
void main ()
```

```
{
```

```
    int c;
```

```
    int sum();
```

```
    printf ("%d", c);
```

```
}
```

```
int sum()
```

```
{
```

```
    int a, b, sum;
```

```
    scanf ("%d", &a);
```

```
    scanf ("%d", &b);
```

```
    sum = a+b;
```

```
    return (sum);
```

```
}
```

Output: - 8 2 10

2)

Subtraction:

Program:

```
# include <stdio.h>
```

```
void main ()
```

```
{
```

```
    void sub();
```

```
    sub();
```

```
}
```

void sub()

{

int e, f, sub;

scanf ("%d", &e);

scanf ("%d", &f);

sub = e - f

printf ("%d", sub);

}

output  $\rightarrow$  5 2

3.

### 3) Multiplication:

Program:

# include <stdio.h>

void main ()

{

int d;

d = mul();

printf ("%d", d);

}

int mul()

{

int p, k, mul;

scanf ("%d", &p);

scanf ("%d", &k);

mul = p \* k;

return (mul);



12

Output:-

4 7  
28

4)

Division: Program:

# include <stdio.h>

void main ()

{

    div ();

}

void div ()

{

    int w,r,div;

    scanf ("%d", &w);

    scanf ("%d", &r);

    div = w/r

    printf ("%d", div);

}

→ output : 25 5

5

\* Problem Statement 5: To find sum & product  
using user defined function with  
Arguments.

(P.T.O.)

Program:

```
# include <stdio.h>
void main ()
```

{

```
int a, b, c;
```

```
int sum (int a, int b);
```

```
a = 100, b = 200;
```

```
c = sum (a, b);
```

*Declaration*

*Calling*

```
printf ("%d\n", c);
mul (a, b);
```

3  
{  
int sum (int a, int b)

```
int s;
```

```
s = a + b;
```

```
return (s);
```

3  
void mul (int a, int b)

{

```
int m;
```

```
m = a * b;
```

```
printf ("%d", m);
```

3

Output:

300

20000.



## \* Problem statement 6 :- Strings

Program:

```
#include <stdio.h>
#include <string.h>
void main ()
{
    int i, j, k, l, m;
    char a[11] = "Hello World";
    i = sizeof(a);
    printf ("%d\n", i);
    char b[] = {'a', 'b', 'c', 'd', 'e', '\0'};

    printf ("%s\n", b);
    j = sizeof(b);
    k = strlen(b);

    printf ("%d\n", j);
    printf ("%d\n", k);
    char c[] = "abcdef";
    printf ("%s\n", c);
    l = sizeof(c);
    m = strlen(c);
    printf ("%d\n", l);
    printf ("%d", m);
```

2

\* Output: 11

abcde

6

5

abcdef

7

8.

\* Problem Statement 7: Program:

```
# include <stdio.h>
# include <string.h>
void main ()
{
    char a[] = "Hello";
    char b[] = "World";
    printf ("%s", strcat (a,b));
}
```

→ Output: Hello World.

\* Problem Statement 8: Program:

```
# include <stdio.h>
void main ()
{
    int i;
    char a[8] = "Nakshatram@192.168.1.10";
    Nakshatra";
```

```
printf ("%\n", a[9]);  
printf ("%\n", a[5]);  
printf ("%\n", a);  
a[3] = 'a';  
printf ("%\n", a);  
for (i=0; i<9; i++)  
    printf ("%\n", a[i]);
```

3.

\* Output: (null)

a

Nakshatra

Nakshatra.

