KATHMANDU UNIVERSITY

Department of Computer Engineering.

A

Lab Report On

Computer Programming {COMP 102}

Lab Sheet No: 4

Submitted by:

Ashraya Kadel

UNG CE 1/1

R.N:25

Submitted to:

Sameer Tamrakar

Department of

Computer Engineering

Submission date: 23/04/2023

WEEK 9: ARRAY

In week 9, we learnt about the use of arrays in C-programming. We did some programs using arrays.

<Q.1>: Write a C-program to store N-numbers in 1-d array and calculate with the help of function.

Sol^n

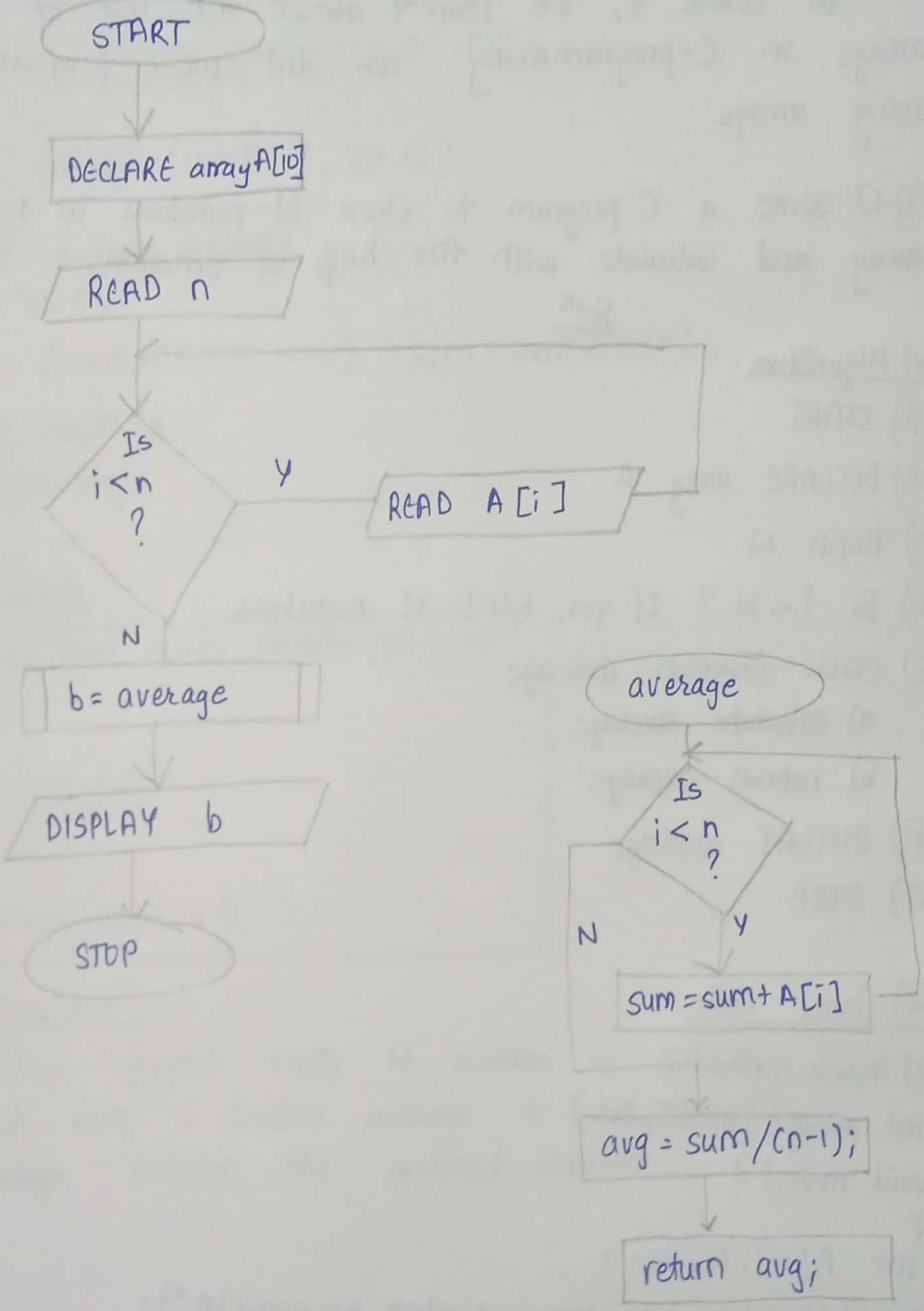
*) Algorithm

```
A) START
B) DECLARE array A
C) READ N
D) Is i<N ? If yes, READ N numbers
E) CALL function average
    a) Calculate average
    b) return average
F) DISPLAY average
G) STOP
```

*) Source Code

```
#include <stdio.h>
int average (int[], int)
void main()
{
int A[10], N, b,i ;
printf("Enter how many numbers you want\n");
scanf("%d",&N);
```

*) Flowchart


START
DECLARE array A[10]
READ n
Is i<n ?
Y
READ A[i]
N
b = average
DISPLAY b
STOP
average
Is i<n ?
N
Y
sum = sum + A[i]
avg = sum/(n-1);
return avg;

```
for (i=0; i<N; i++) scanf("%d", &A[i]);
ab= average (A,N)
printf("Average value = %d\n", b);
}
int average (int A[ ], int N)
{
int a, sum, n, avg;
a= N-1;
for (n=0;n<=a ,n++)  sum = sum + A[n];
avg =sum /a;
return avg;
}
```

*) Output

```
Enter how many numbes you want
6
2
4
6
8
1
3
Average value = 4
```

*) Description:

This program reads N numbes of numbers and calls a function average to calculate the average between the entered numbers.

<Q.2>: Write a C-program to convert a binary number to decimal with the help of the function todecimal (char bits [20], int length).

Soln

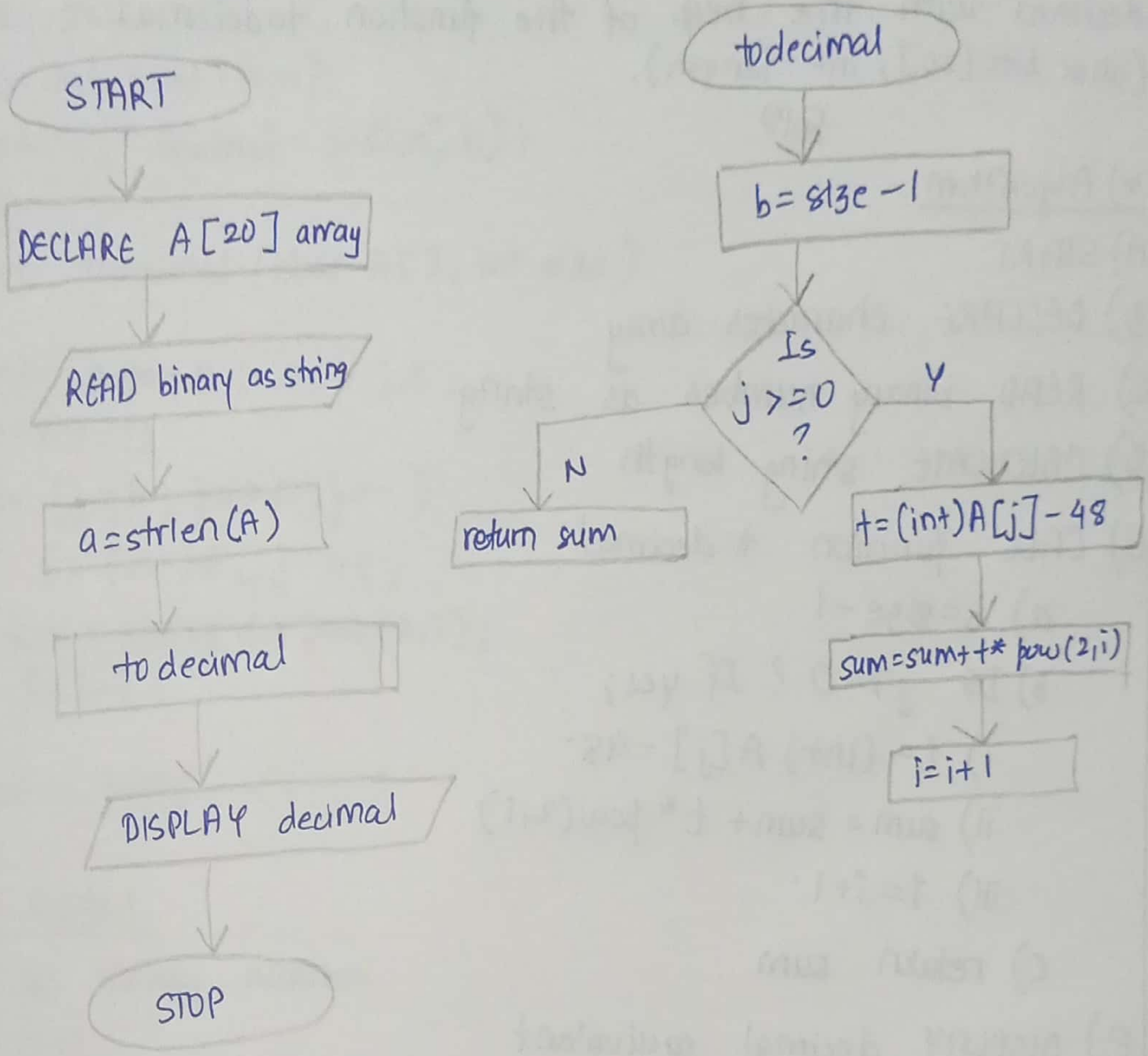
*) Algorithm

```
A) START
B) DECLARE character array
C) READ binary number as string
D) CALCULATE string length
E) CALL function todecimal
    a) b=size-1
    b) IS j>=0 ? If yes;
        i) t= (int) A[j]-48.
        ii) sum = sum+ t* pow(2,i)
        iii) i= i+1.
    c) return sum
F) DISPLAY decimal equivalent
G) STOP
```

*) Source Code

```
#include <string.h>
#include < math.h>
#include <stdio.h>
int todecimal (char[],int );
void main( )
{
char A [20];
int b,a;
printf ("Enter binary number\n");
```

*) Flowchart:


START
DECLARE A[20] array
READ binary as string
a=strlen(A)
to decimal
DISPLAY decimal
STOP
todecimal
b= size -1
Is
j>=0
?
Y
N
return sum
t=(int)A[j]-48
sum=sum+t* pow(2,i)
i=i+1

```
scanf("%s", A);
 a= strlen(A);
 b= todecimal (A,a);
 printf(" Decimal = %d\n", b);
 }
int todecimal (char A[], int size)
{
 int sum=0,i=0, b,j,t;
 b= size-1;
for (j=b; j>=0;j--)
{ t= (int) A[j]-48;
  sum= sum+ t* pow (2,i);
   i=i+1;
 }
 return sum;
 }
```

*) Output

```
Enter binary number
11011
Decimal = 27
```

*) Description:

This program reads a binary number as string and uses todecimal function to change it into the decimal equivalent.

WEEK 10: One-D Array and Functions:

In week 10, we learnt about using one-d array and passing them into functions.

<Q.1>: WAP to arrange the numbers in ascending order using bubble sort.

Soln:

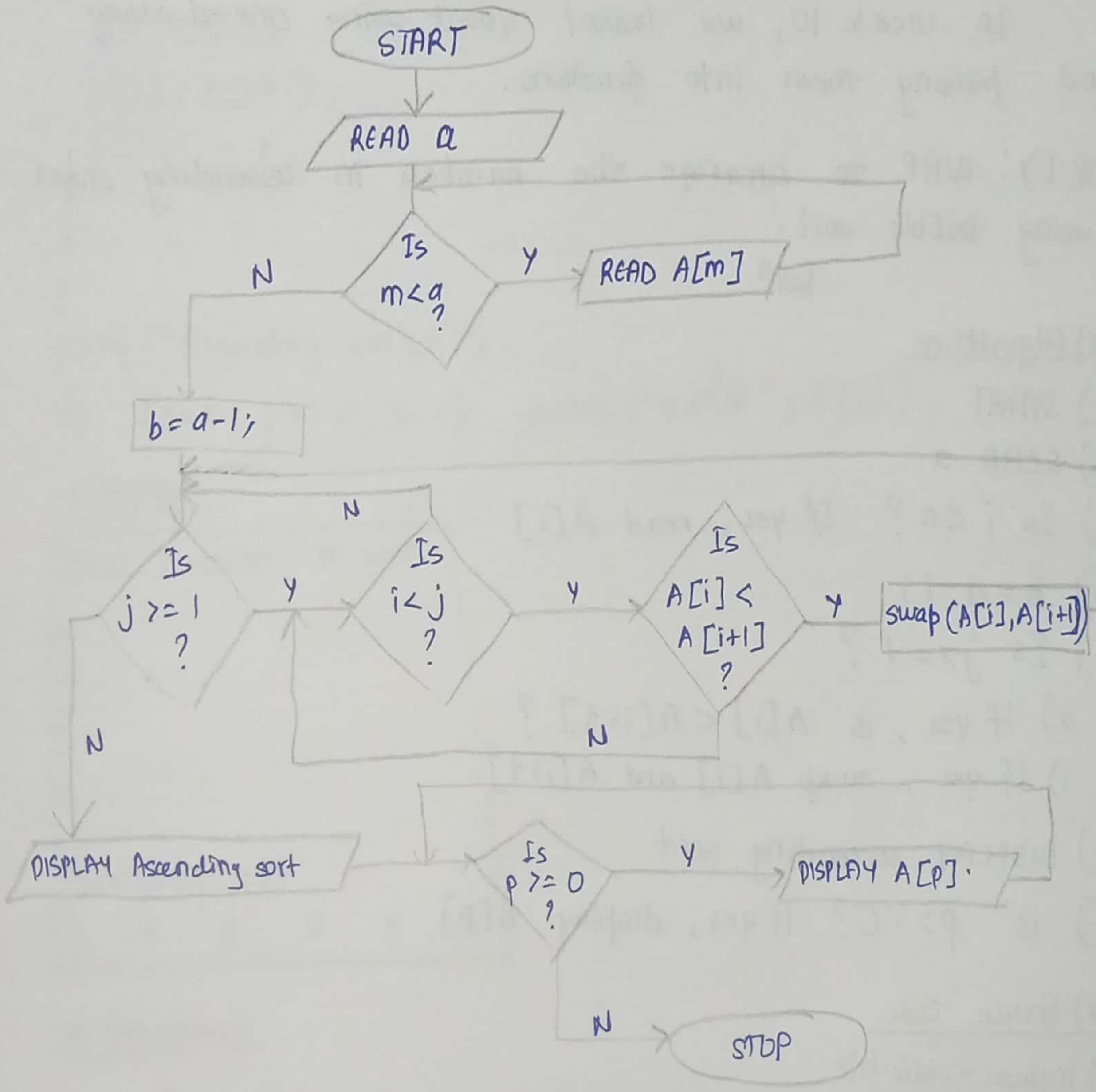
*) Algorithm

```
A) START
B) READ a
C) Is i<a? If yes, read A[i]
D) b=a-1;
E) Is j>=1?
   a) If yes, is A[i]<A[i+1]?
      i) If yes, swap A[i] and A[i+1]
F) DISPLAY ascending sort
a) Is p>=0? If yes, display A[p]
```

*) Source Code

```
#include <stdio.h>
void main()
{
int A[15],d,b,m,i,j,t,n,p,q;
printf("Enter how many numbers\n");
scanf("%d",&a);
for (m=0;m<a;m++) scanf("%d",&A[m]);
b=a-1;
for (j=b;j>=1;j--)
{
for (i=0;i<j;i++)
{
```

*) Flowchart


START
READ a
Is m<a ?
Y
READ A[m]
N
b=a-1;
Is j>=1 ?
Y
N
Is i<j ?
Y
N
Is A[i] < A[i+1] ?
Y
swap(A[i],A[i+1])
N
N
DISPLAY Ascending sort
Is p>=0 ?
Y
DISPLAY A[p].
N
STOP

```
if (A[i] < A[i+1])
{
  t= A[i];
  A[i] = A[i+1];
  A[i+1] = t;
}
}
}
printf("Ascending sort\n");
for (p=b; p>=0; p--)  printf("%d\t", A[p]);
```

*) Output

```
Enter numbers to sort
11
8
9
3
4
5
Ascending sort
3   4   5   8   9   11
```

*) Description

This program reads N numbers & uses bubble sorting method to sort the numbers in ascending order. We did this ~~using~~ without using function.

<Q.27: WAP to find decimal equivalent of a binary number.

Soln:

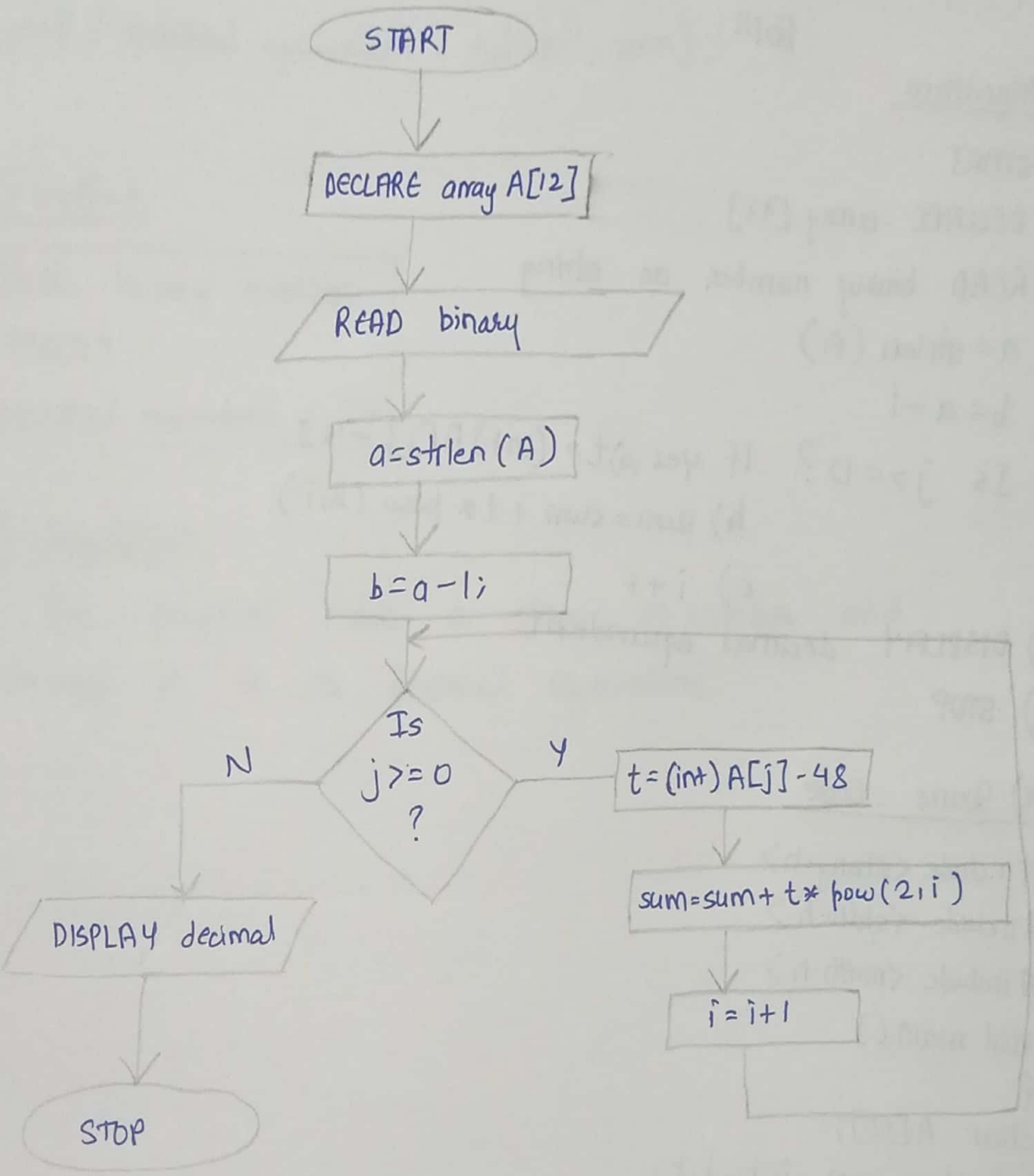
*) Algorithm

```
A) START
B) DECLARE array [12]
C) READ binary number as string
D) a = strlen (A)
E) b = a-1
F) Is j>=0? If yes, a) t = (int) A[j] - 48
                    b) sum = sum + t* pow (2,i)
                    c) i++
G) DISPLAY decimal equivalent
H) STOP
```

*) Source Code

```
#include <string.h>
#include <stdio.h>
#include <math.h>
void main()
{
char A[12];
int b,a,sum, i, b,j,t;
printf("Enter binary number\n");
scanf("%s", &A);
a=strlen(A);
b=a-1;
for (j=b;j>=0;j--)
{
t=(int) A[j]-48;
sum= sum + t* pow(2,i);
```

*) Flowchart


START
DECLARE array A[12]
READ binary
a=strlen (A)
b=a-1;
Is j>=0 ?
N
Y
t=(int) A[j]-48
sum=sum+ t* pow(2,i)
i=i+1
DISPLAY decimal
STOP

```
i=i+1;
}
printf ("Decimal equivalent = %d\n", sum);
}
```

*) Output

```
Enter binary number
11011
Decimal equivalent = 27
```

*) Description:

This program reads a binary ~~of~~ number and converts it to its decimal equivalent.

WEEK 11: MULTI-D ARRAY AND FUNCTIONS

In week 11, we learnt using matrix operations with the help of the function.

<Q1>: WAP to evaluate transpose of n by n matrix with the help of function transpose (int matrix [][20], int n).

Soln:

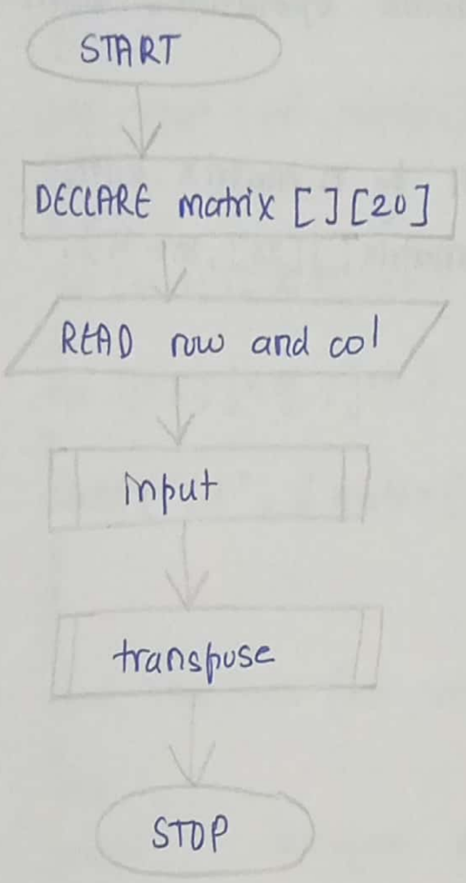
*) Algorithm

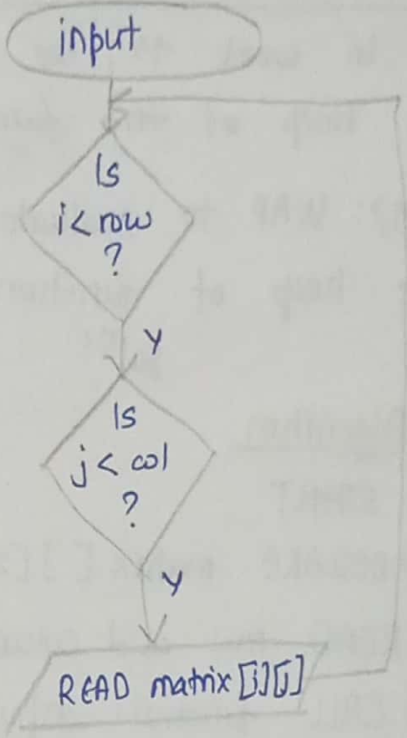
```
A) START
B) DECLARE matrix [][20]
C) READ row and column
D) CALL function input
    a) Is i<row?
        If yes,
       i) j<col?
       If yes, read matrix [i][j]
E) CALL function transpose
    a) If Is i<row?
        If yes,
       i) j<col?
       If yes, display transpose [j][i]
```

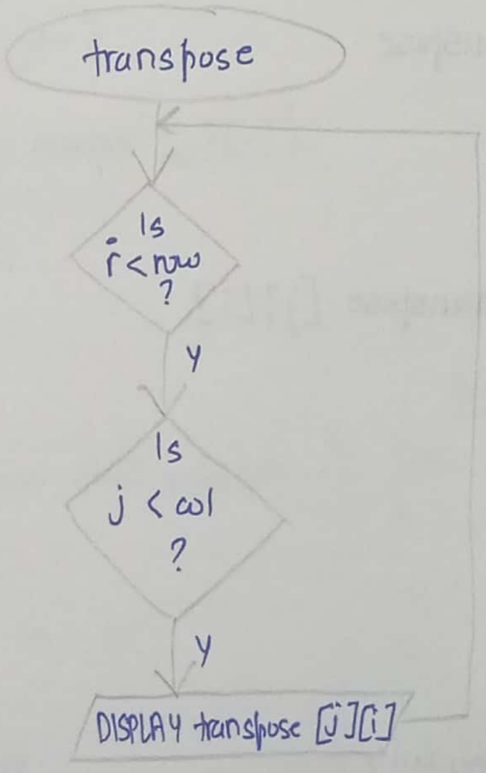
*) Source Code

```
#include <stdio.h>
void input ();
void transpose ();
void main()
{
int matrix [][20], row, col;
printf ("Enter size in row and column \n");
scanf ("%d", row);
scanf ("%d", col);
```

*) Flowchart:


START
DECLARE matrix [ ] [20]
READ row and col
input
transpose
STOP


input
Is i< row ?
Y
Is j< col ?
Y
READ matrix [i][j]


transpose
Is i< row ?
Y
Is j < col ?
Y
DISPLAY transpose [j][i]

```
input (matrix, row, col);
transpose (matrix, row, col);
}
void input (int matrix [][20], int x1 , int y1 )
{
int i,j;
for (i=0; i<x1 ;i++)
{
for (j=0; j<y1 ; j++ )
{
scanf ("%d", & matrix [i][j] );
}
}
}
void transpose (int matrix [][20], int x1 , int y1)
{
int i,j;
printf ("\n Transpose matrix\n");
for (i=0; i<x1 ; i++ )
{
for (j=0; j<y1 ; j++ )
{
printf ("%d\t", matrix [j][i]);
}
printf ("\n\n");
}
}
```

* Output

```
Enter size in row and column
2
2
1
2
4
5
 Transpose matrix
    1     4
    2     5
```

*) Description

This program reads the elements of a matrix through function input.

The function transpose inverts the row and column.

<Q27: WAP ~~to~~ in C for matrix addition with help of function int add (int a[][20], int b[][20], intc[][20], int row, int col).

Soln:

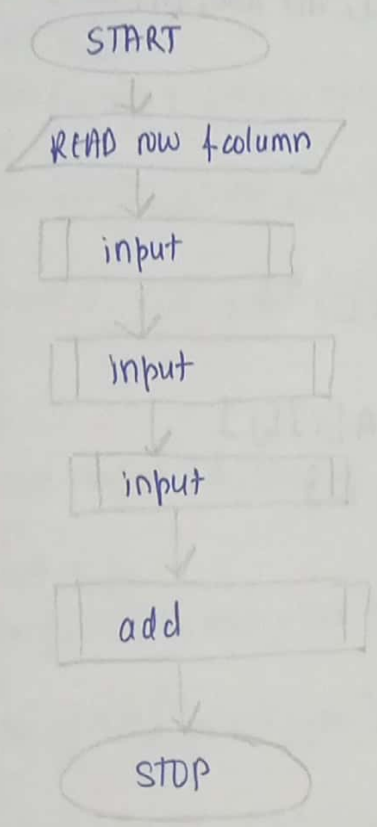
*) Algorithm

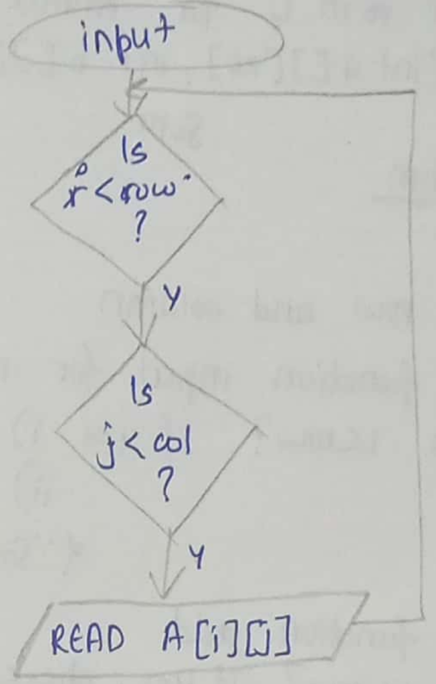
```
A) START
B) READ row and column
C) CALL function input for matrix A, B and C.
    a) Is i<row? If yes i) Is j<col?
                        ii) If yes, READ A[i][j]
                        {∵ Similar for B and C}
D) CALL function add
    a) Is i<row? If yes, check j<column.
    If yes,
  i) D[i][j] = A[i][j] + B[i][j] + C[i][j]
  ii) ~~Sum~~ DISPLAY "Sum of matrices"
  - If i<row? If yes, Is j<col? If yes, display D[i][j]
E) STOP
```

*) Source Code

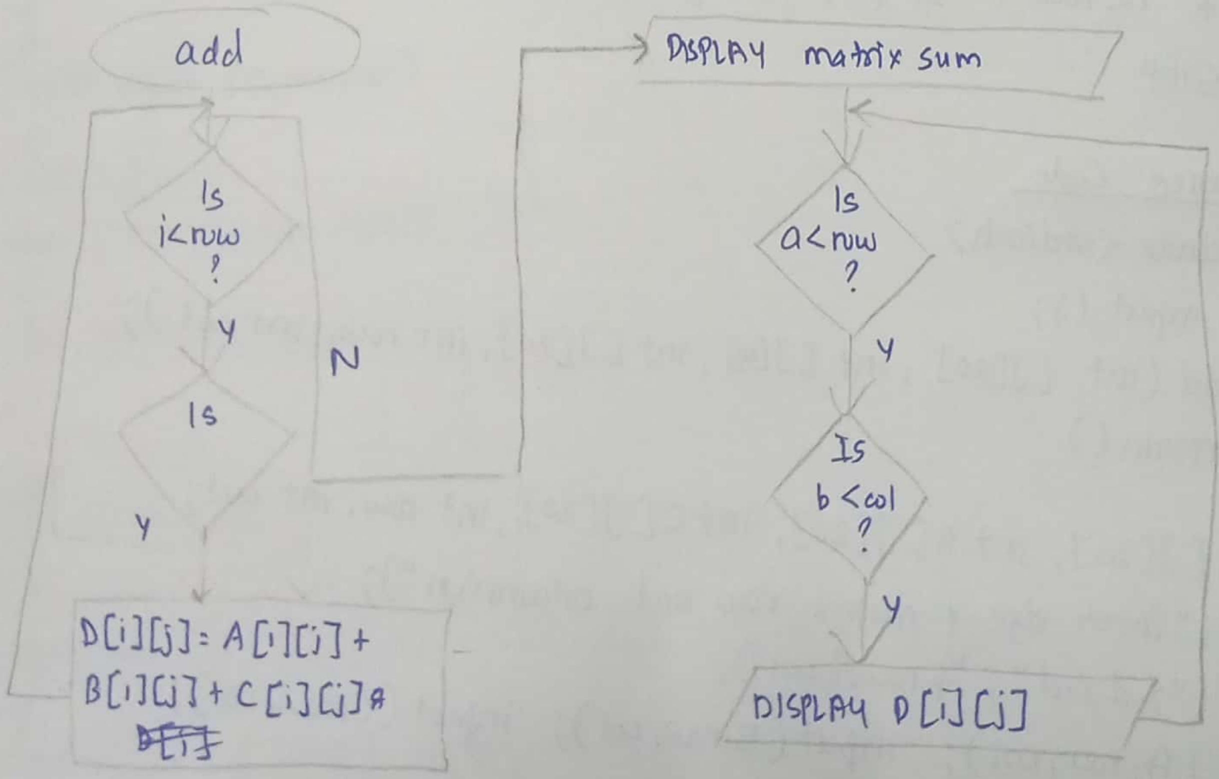
```
#include <stdio.h>
void input ();
int add (int [][20], int [][20], int [][20], int row, int col);
void main()
{
int A[][20], int B[][20], int C[][20], int row, int col;
printf("Enter size of matrix row and column\n");
scanf("%d %d", &row,&col);
input (A, row, col); input (B, row, col); input (C, row, col);
add (A, B, C, row, col);
}
```

*) Flowchart





{Similar for B and C}



```
void input (A, row, col)
{
int i,j;
for (i=0; i<row; i++)
{
for (j=0; j<col; j++)
{
scanf("%d", 4A[i][j]);
}
}
}
void input (B, row, col)
{
int i,j;
for (i=0; i<row; i++)
{
for (j=0; j<col; j++)
{
scanf("%d", &B[i][j]);
}
}
}
void input (c, row, col)
{
int i,j;
for (i=0; i<row, i++)
{
for (j=0; j<col; j++)
{
scanf("%d", 4c[i][j]);
}
}
}
int add (int A[][20], int B[][20], int C[][20], int row, int col)
{
int i,j, D[][20], a,b;
for (i=0; i<row; i++)
{
for (j=0; j<col, j++)
{
```

```
D[i][j] = A[i][j] + B[i][j] + C[i][j];
}
}
printf(" Sum of matrices\n");
for (a=0 ; a<row ; a++ )
{
for (b=0; b<col; b++)
{
~~scanf~~ printf("%d \t", D[i][j]);
}
printf("\n\n"); return 0;
}
```

*) Output

```
Enter size of matrix
2
2
1
4
6
1
2
5
6
2
3
4
3
4
Sum of matrices
 6        13
15         7
```

*) Description

This programs calls input function three times to read matrix A, B and C.
and
it ~~returns~~ uses function add to add the elements of these matrices and ~~conver~~ display its sum.