KATHMANDU UNIVERSITY

Department of Computer Engineering

A

Lab Report On

COMPUTER PROGRAMMING {COMP102}

Lab Sheet No: 5

Submitted by:
Ashraya Kadel
UNG: CE (I/I)
Roll No: 25

Submitted to:
Sameer Tamrakar
Department of
Computer Engineering.

WEEK 12: POINTERS

In week 12, we learnt about the use of pointers in C-programming. We passed pointers into functions.

<Q.1>: WAP that swaps two variables.

Ans:

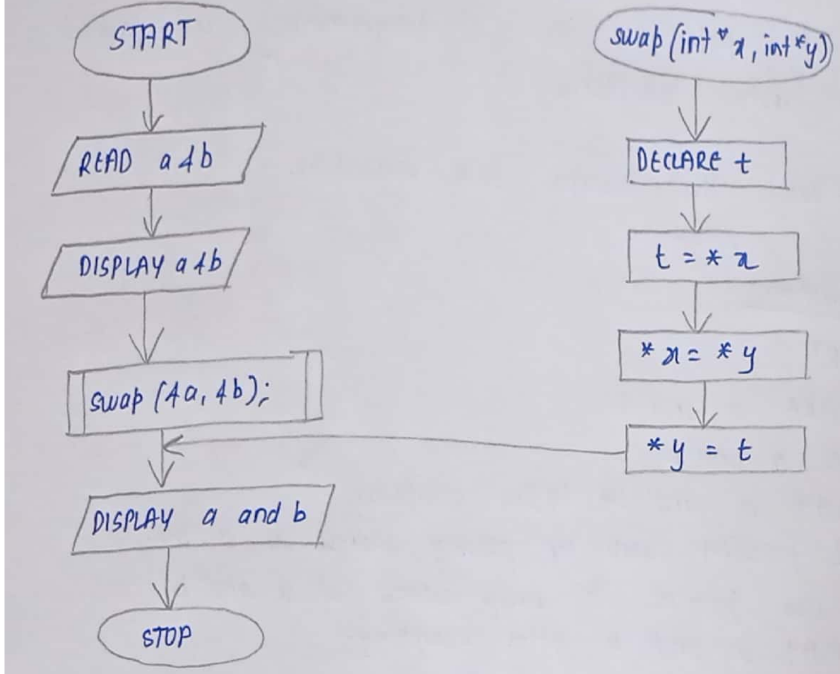
*) Algorithm:

```
i) START
ii) DECLARE a and b
iii) READ a and b
iv) DISPLAY a and b before swapping
v) CALL function swap by passing address on a and b
        a) Use pointers to swap values of a and b.
vi) DISPLAY a and b after swapping.
vii) STOP
```

*) Source Code

```
#include <stdio.h>
void swap (int *, int *);
void main ()
{
    int a, b;
    printf ("\n Input two numbers\n");
    scanf(" %d %d ", &a, &b);
    printf ("Before swapping: A = %d and B = %d \n", a, b);
    swap (&a, &b);
    printf ("After swapping: A = %d and B = %d \n", a, b);
}
```

*) Flowchart



```
void swap (int *x, int *y)
{
    int t;
    t= *x; *x = *y; *y = t;
}
```

*) Output:

```
Input 2 numbers
2
3
Before swapping: A=2 and B=3
After swapping: A=3 and B=2
```

* Description

This program reads two numbers a and b and displays it. The address of the two variablesis sent to function swap which switches the variables using pointers.

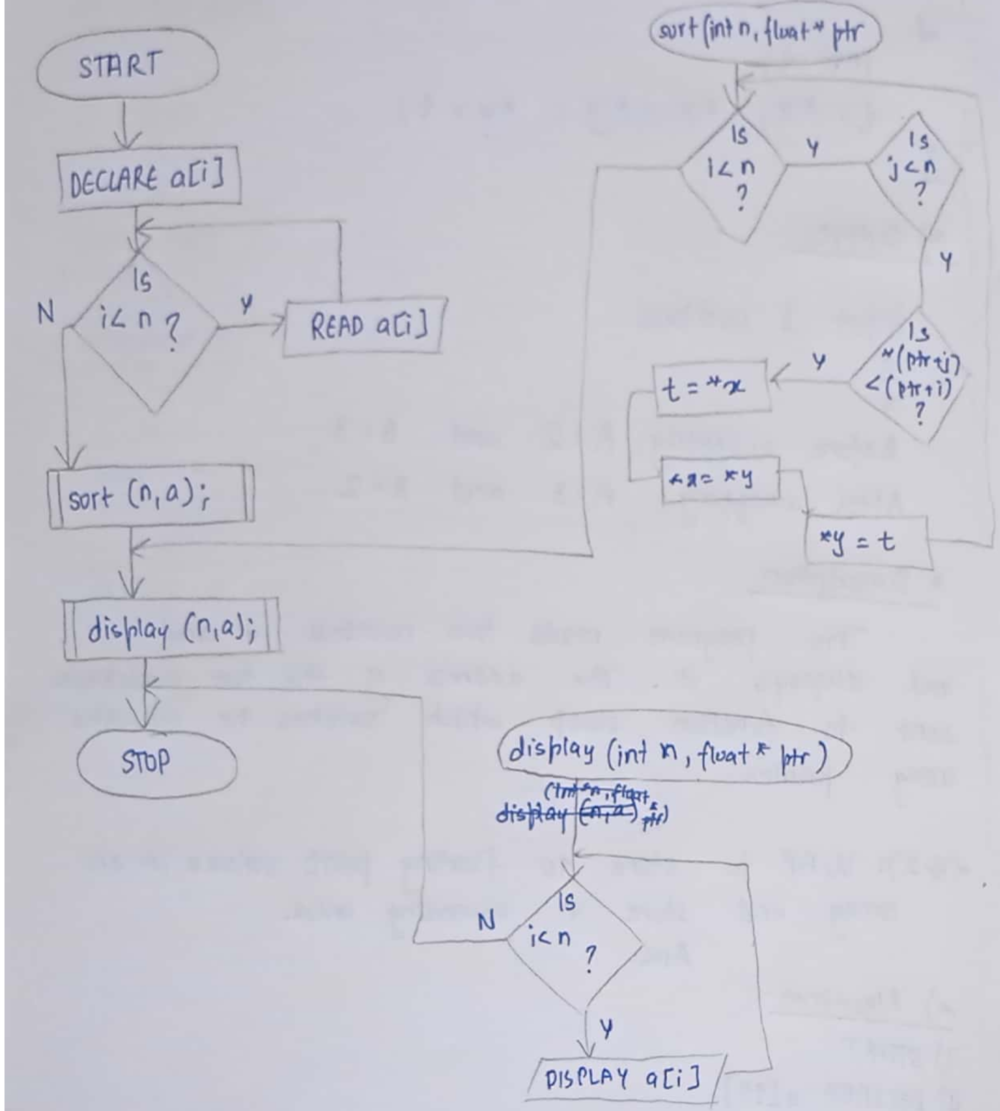
<Q.27: WAP to store 10 floating point values in an array and store in ascending order.

Ans:

*) Algorithm :

```
I) START
II) DECLARE a[15]
III) Is i<n ? If yes, READ a[i]
(iv) CALL function sort passing n and base ~~array~~ address of a
A) Is i<n? If yes, i) Is j<n? If yes, a) Is *(ptr+j)<*(ptr+i)?
    If yes, swap *(ptr+j)< *(ptr+i)
(v) CALL function display passing n and base address of a
~~a~~ i) Is i<n? DISPLAY a[i]
(vi) STOP
```

(*) Flowchart:



*) Source Code:

```
#include <stdio.h>
void display (int, float *);
void sort (int, float*);
void main()
{
int n=10, i,j; float a[15];
for (i=0; i<n; i++)  scanf ("%f", &a[i]);
Sort (n,a);   display (n,a);
}
void display (int n, float * ptr)
{
int i;
for (i=0; i<n; i++) ~~scanf ("%f~~ printf("%f\t", *(ptr+i));
}
void sort (int n, float * ptr)
{ int i, j; float t;
for (i=0; i<n; i++) {
for (j=~~0~~i+1; j<n; j++) {
if (*(ptr+j) < *(ptr+i)) { t= *(ptr+i); *(ptr+i) = *(ptr+j);
*(ptr+j) = t; }}}
}
```

*) Output

```
1.1     1.0
1.2     1.1
1.3     1.2
1.0     1.3
1.4     1.4
1.5     1.5
1.6     1.6
1.7     1.7
1.8     1.8
1.9     1.9
```

*) Description

This program reads 10 float numbers. Uses sort function to sort out them in ascending order.
The function display gives the output of the sorted float value.

WEEK 13: STRUCTURE

In week 13, we learnt the use of ~~functions~~ structures in C-programming.

<Q.17>: WAP defining structure called student with suitable attributes, reads 5 data. And ~~#~~ displays in ascending order of roll numbers.

Ans:

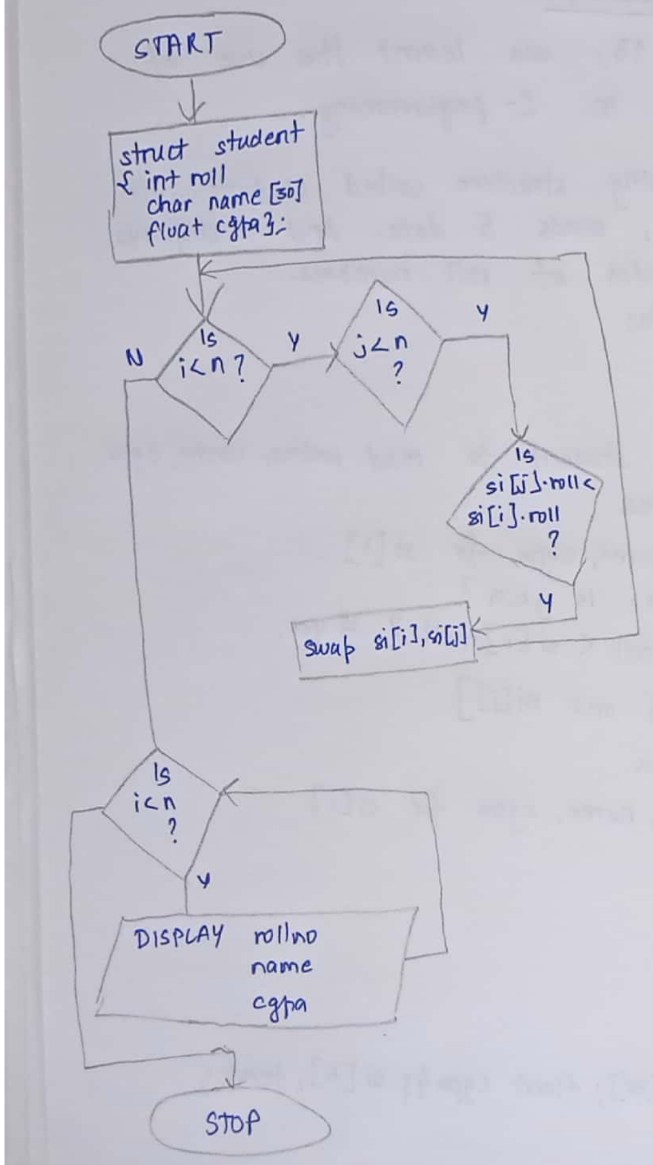
*) Algorithm

```
i) START
ii) DECLARE structure student to read rollno, name, cgpa
iii) Is i<n? If yes
    a) READ roll no, name, cgpa for si[i].
iv) Is i<n? If yes, is j<n?
        a) Is si[j].roll < si[i].roll? If yes,
        b) swap [si[i] and si[j]]
v) Is j<n? If yes.
        a) DISPLAY roll, name, cgpa for si[i]
vi) STOP
```

*) Source Code:

```
#include <stdio.h>
struct student
{ int roll; char name[30]; float cgpa }; si [6], temp ;
void main ()
{ int n=5,i,j;
for (i=0;i<n;i++)
{scanf("%d",& si[i].roll); scanf("~~$~~ %[^\n]",& si[i].name);
 scanf("%f",&si[i].float); printf("~~$~~ ----\n");
}
```

*) Flowchart:



```
for (i=0;i<n;i++) { for (j=i+1;j<n;j++) {
if (si[j].roll<si[i].roll) {
temp= si[i]; si [i] = si[j]; si[j] = temp; } } }
for (i=0; i<n; i++)
{ printf("Rollno: %d\n", si[i].roll); printf("Name: %s\n", si[i].name);
printf(" CGPA : %f\n ", si[i]. cgpa); } printf("---- \n"); }
}
```

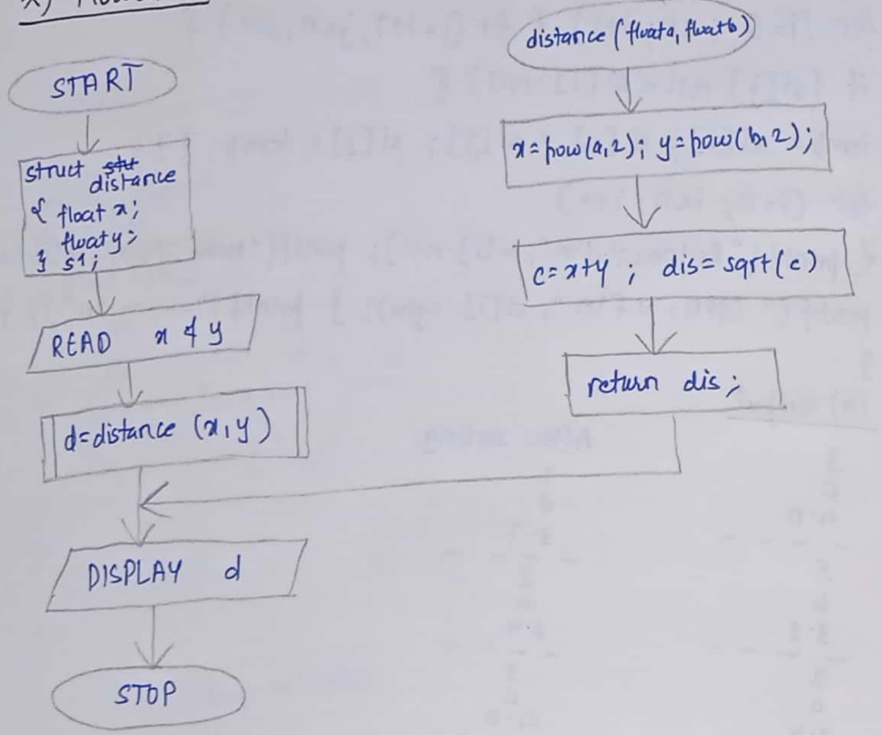
(*) Output

```
3                After sorting
c                   1
4.0                 d
- - - -             3.7
5                 - - - -
b                   2
3.8                 a
- - - -             3.4
2                 - - - -
a                   3
3.4                 c
- - - -             4.0
1                 - - - -
d                   4
3.7                 e
- - - -             4.0
4                 - - - -
e                   5
4.0                 b
- - - -             3.8
                  - - - -
```

*) Description:

This program reads the attributes of 5 students through structure student. If reads name, cgpa and roll no and sorts the data. The sorted data is displayed.

*) Flowchart:



<Q.27>: Consider a plane graph. Use function to return distance between given point and origin.

*) Algorithm:

i) START ii) DEFINE structure distance reading x and y.

iii) READ x and y iv) CALL function distance passing value of x and y.

a) calculate distance

b) return dis;

~~v) CALL~~ .

v) DISPLAY distance

vi) STOP

*) Source Code:

```
#include <stdio.h>  #include <math.h>
float distance (float , float);
struct distance { float x; float y; } s1;
void main ()
{ printf("Enter x and y\n"); float d;
  scanf ("%f%f", &s1.x ,&s1.y);
  d= distance (s1.x, s1.y); printf("Distance=%d\n", d); }
float distance (float a, float b)
{ float dis, x, y, c;
  x= pow(a,2);  y=pow(b,2);
  c=x+y;  dis= sqrt(c); return dis; }
```

*) Output:

```
Enter distance x and y
3.00
4.00
Distance= 5.0000
```