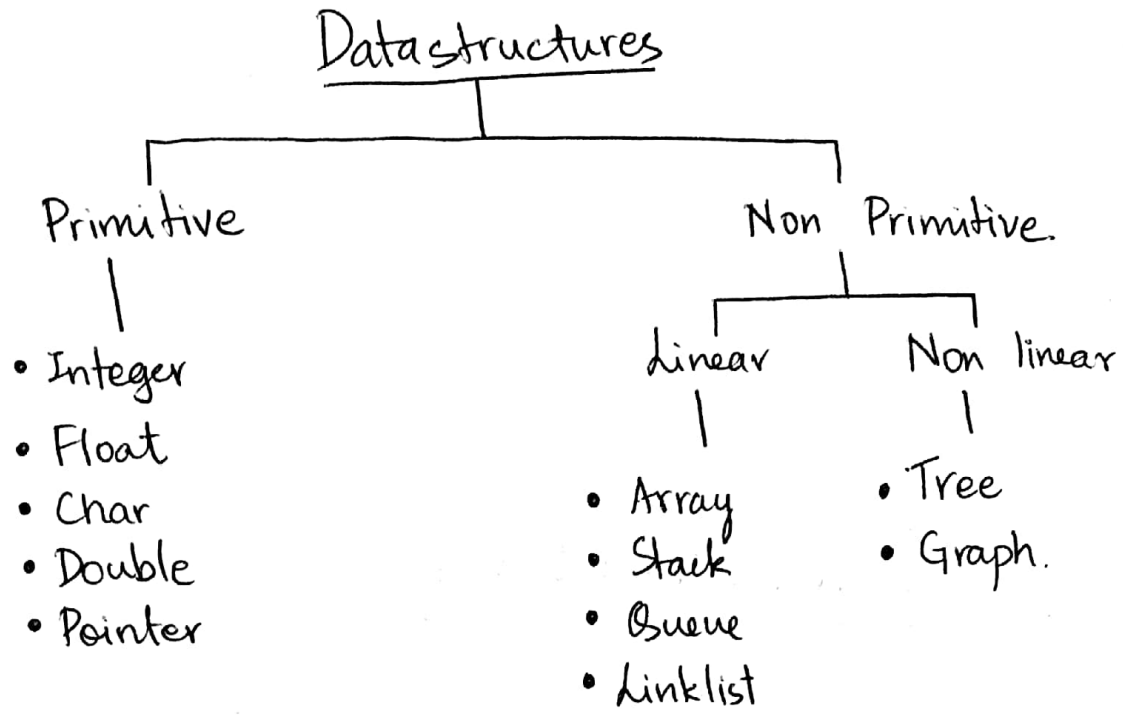

Recursion vs Iteration ★★★ (less code, slower / more code, fast)

Classification of data structures ★★★

Structure is a derived or constructed datatype which holds either similar or dissimilar data items.

The syntax of defining a structure is as follows

```
struct tagname
{
    datatype1  m1;
    datatype2  m2;          /*members*/
        :
    datatype n  mn;
};
```

Write a program to read and print a student info - name, sem, dept. using structure.

```
#include <stdio.h>

struct student
{
    char name[30];
    int  sem;
    char dept[15];
};

main()
{
    struct student s;
    printf("Read name:\n");
    scanf("%s", s.name);
```

```
    printf ("Read sem: \n");
    scanf ("%d", &s.sem);
    printf ("Read dept: \n");
    scanf ("%s", s.dept);
    printf (" %s \n %d \n %s", s.name, s.sem, s.dept);

    return 0;
}
```

WAP to read n student's info then print.

```
#include <stdio.h>

struct student
{
    char name[30];
    int sem
    char dept[15];
};

int main()
{   int n, i;
    struct student s[20];
    printf("Enter no. of students: \n");
    scanf ("%d", &n);

    for (i=0; i<n; i++)
    {
        pf ....
        sf .... s[i]. ...
    }
    for (i=0; i<n; i++)
    {
        pf
    }
```

WAP to read student info and to display it using a pointer

```
struct student
{
    char name[30];
    int sem;
    char dept[15];
};
int main()
{
    struct student s={"Pari", 3, "CSE"};
    struct student *ptr;
    ptr = &s;
    pf("Name %s\n sem %d\n Dept %s\n", ptr->name,
        ptr->sem, ptr->dept);
    return 0;
}
```

WAP n students using pointers.

```
struct student
{ ...
}
int main()
{
    int n, i;
    struct student s[20];
    struct student *ptr;
    read 'n';
    for (i=0; i<n; i++)
    {
        scanf("%s", s[i].name);
    }
```

```
for (ptr = &s ; ptr < &s[n] ; ptr++)
{
    print.....
}
```

WAP to display the content of array in reverse using ptr

```
int   s[5] , * = [1, 2, 3, 4, 5];
int   *ptr ;
   ptr = &s[4]
for ( ptr = &s[4] ; ptr > &s[0] ; ptr-- )
{
    printf(" %d\n", *ptr );
}
```

```
{
    printf("%d\n", *(ptr--));
}
```