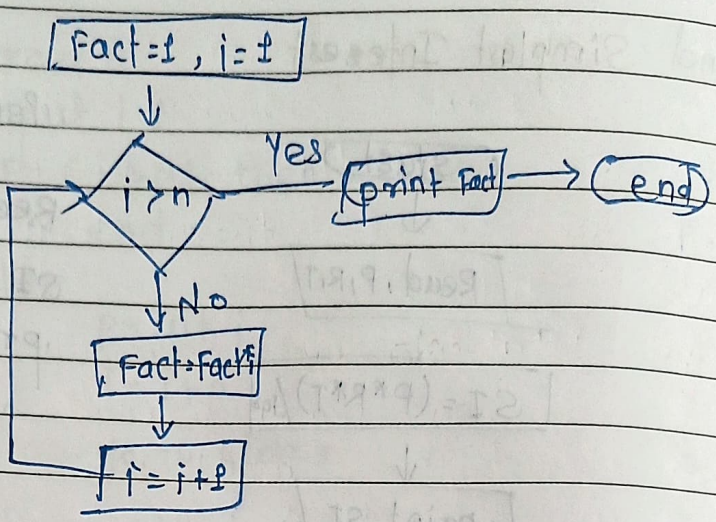


Pseudo:

1. Read N
2. input i=1 Fact=1
3. IF i>N then
   print fac
4. else then
   Fact = Fact * i
   i = i+1
   go to (3) ....

Day 2

Programming Language

Language in which we are giving instructions to computer.

Compiler

A ~~task~~ compiler takes a programe as whole & genreate intermediate machine code. This translate entire source code source code in the single run.

Inteepretee:
It's translate the programe line by line & also nevee genreates any Inteemediate machine code.

IDE :- (Integrated Oevelopment Enviorment)
It's provides an enviorment to write our own codes and also provides various Featuees to increase our productivity.

Writing our First programe.

```
#include <iostream>        ← File having impl of cout
using namespace std;       ← standard name space
int main() {               ← compilee staet their prog.
    cout << "Hello Microsoft" << endl;
}                          ↑ print or show some on screen
```

Adding a new line :- endl , '\n'

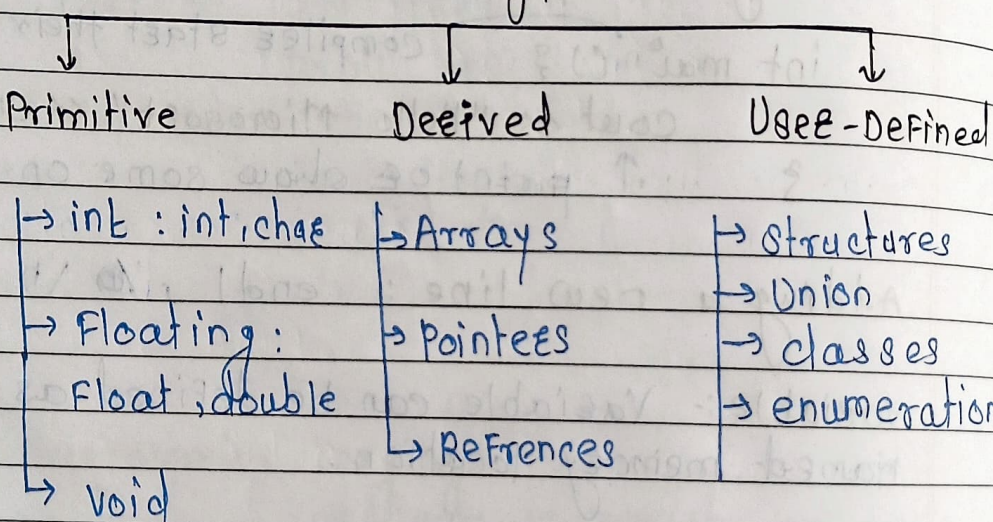
vaeiable :- Vaeiable can be defined as named memoey location.

Data type :- This tells what type of data is stored.

Data Types:-

Basic Data Types	32-bit CPU byte	Range	64-bit CPU byte
Char	1	-128 to 127	1
Short	2		2
int	4		4
long	4		8
long long	8		8
float	4		4
double	8		8

C++ Data Types.



Primitive :- These data types are already defined.

Derived :- These data types are made with using primitive data types

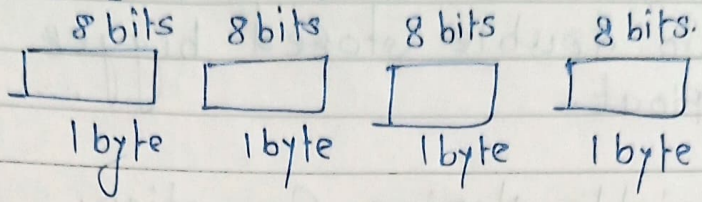
User-Defined :- We make our own datatypes.

- Minimum space in memory is 1byte

int data type :-

Takes up 4 byte of memory

4 × 8 = 32 bits.



Each 1 bit has either 0 or 1.

∴ 2 combination for each bit & hence 2^{32} are possible.

Char data type:-

Takes up 1 byte of memory.

1 × 8 = 8 bits.

∴ 2^8 possible.

Note:- Character a is stored in memory.

∴ No instead an character number is stored in memory & that mapped value is ASCII values.

Boolean Data types

It's have only two value true or false.

true → 1

false → 0

Float & Double data type:-

These is a representing decimal numbers.

Float - 4 bytes - 32 bits
double - 8 bytes - 64 bits.

So double stored bigger no than Float.

Variable Naming Convention:-
1) Can contain letters, digits & underscores
2) Names begins with underscores or letters.
3) Names are case sensitive.
4) Names can not have spaces & no special characters other than underscored.

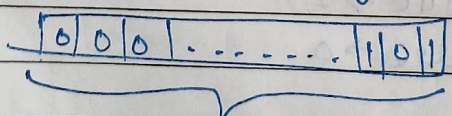
sizeof() :- It's gives an sizeof data type.
ex:- sizeof(a); int a;
size is 4 byte.

Storage of Data:-

Positive Number:-

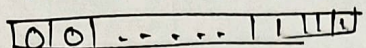
ex:- int a=5

5 → binary → 101



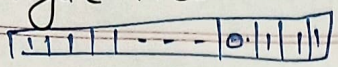
32 bit All left side bit's are zero

Note:- Agar First bit From Eight side is zero then no is +ve.



→ +7

Agar First bit From Eight side is



→ -7.

Negative Number:-

It's will be stored in the form of 2's compliment.

ex:- int a=-7

7 → binary ~~0111~~ 00000111

1. one's complement

11111000

2. 2's complement

```
  11111000
+        1
----------
  11111001   2's
```

∴ Find two's complement & stored it.

- (i) Ignore Negative sign
- (ii) Binary Representation
- (iii) 1's complement
- (iv) 2's complement

For Reading an number From memory?

Calculate an 2's complement.

ex:- ~~000 .... 0100~~

-7 :- 1111 001

2's 1's 0000 110

2's + 1

0000 ..... 111 ✓

-7

Interesting Problem :-

22	31	40	7

Suppose in Memory the value are given so how we know these is 4 byte or 1 byte.

∴ So We know it's using Datatype.

Signed vs Unsigned :-

Signed Data :-

We stored an negative or positive no. ex. -2^{31} to $-2^{31}-1$ For 4 byte

-2^{n-1} to $-2^{n-1}-1$

Unsigned Data :-

Only positive no will stored.

ex. 0 to $2^{32}-1$ if 4 byte.

0 to $2^{n*}-1$

$\frac{2^{32}}{2^1} = 2^{31}$ ← devide two equal part.

Type Casting :-

Converting one data type into another data type.

ex :- Implicit Data Conversion :- Automatically conversion done.

```
char ch = 97;
cout << ch << endl
∴ ⟶ a
```

Note :- $\frac{int}{int} = int$ $\frac{double}{int} = double$

$\frac{float}{int} = float$ ∴ Biggee Data type always dominate

```
int num = 'b';
cout << num;
∴ ⟶ 98
```

Explicit Data Conveesion :-

This is done manually.
This use () opeeatoe.

```
Float a = (Float) 2;
cout << a;
∴ ⟶ a = 2.0
```

Opeeatoes :-

Arithmetic	Relational	Assignment	Logical.
+	>	=	&& — AND
-	<		|| — OR
*	>=		! — Not.
/ - numerator	<=		
% - eem	!=		
	==		

Quotent.

Note :- cond1 && cond2 && cond3

↑

if cond1 become false then ans is false don't need to check next conditions.