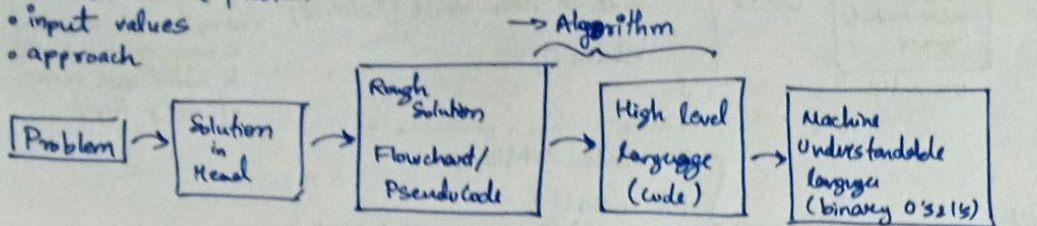
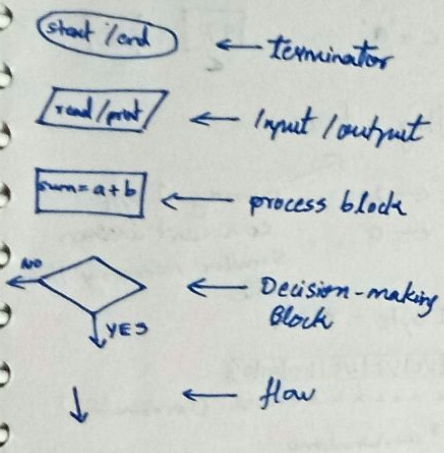
Programming Fundamentals

→ Thought Process to solve a Problem
- understand the problem
- input values
- approach



Architect → Design
Labourers → Work
similarly
Programmer → Algorithms
& Majdoor → code

→ Flowchart

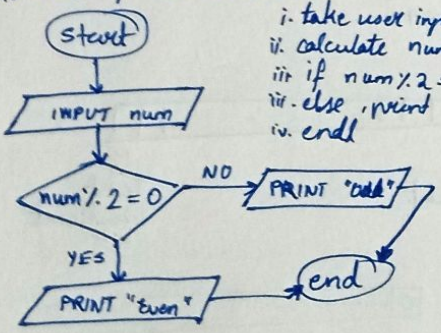


→ Pseudo Code (write anyhow)

Q) difference of 2 numbers
i. read a & b
ii. calculate diff = a-b
iii. print diff

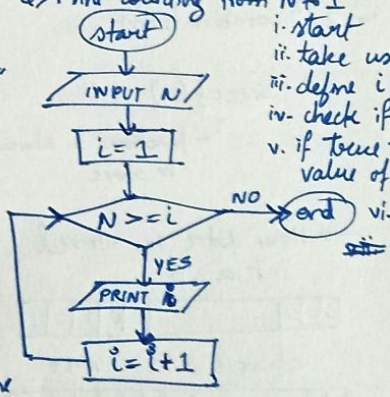
Q) avg of 2 no's
i. start
ii. cin >> a >> b;
iii. int integer value of avg variable be equal to $\frac{a+b}{2}$
iii printf avg
iv. end

Q) check if even or odd



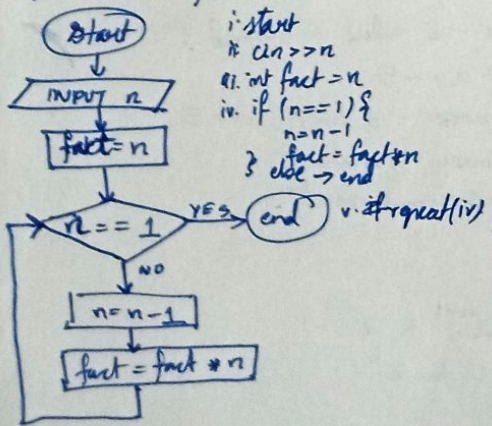
i. take user input for num
ii. calculate num % 2 -
iii. if num % 2 == 0, then print "Even"
iii. else, print "Odd"
iv. endl

Q) Print counting from N to 1



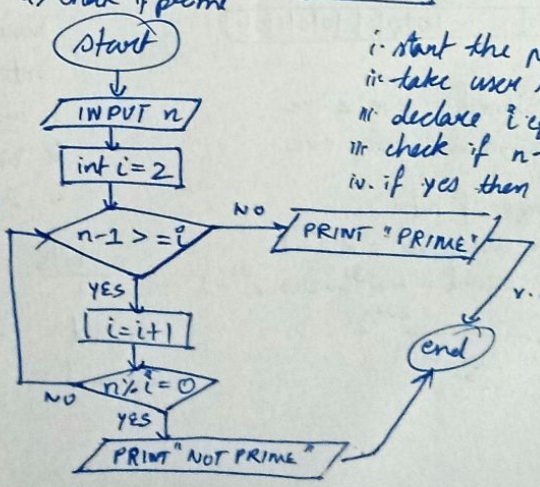
i. start
ii. take user input N
iii. define i = 1
iv. check if N >= i
v. if true then print i & increase value of i by 1, again check v (repeat (v))
vi. else end

Q) Factorial of number



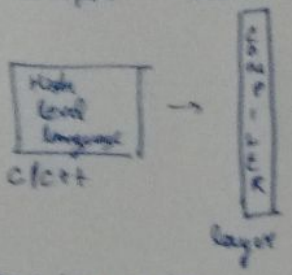
i. start
ii. cin >> n
iii. int fact = n
iv. if (n==1){
n = n-1
fact = fact * n
} else → end
v. if repeat (iv)

Q) Check if prime



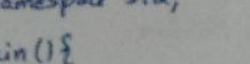
i. start the programm
ii. take user input n
iii. declare i equal to 2
iv. check if n-1 >= i
v. if yes then do i = i+1 & check n%i = 0
if this term then repeat (iii)
else print "NOT prime" & end
v. else if (iv). print "PRIME" & end

→ Compilation Process



C++
```
#include <iostream>
using namespace std;
int main(){
    cout<<"Hello World !"<<endl;
    return 0;
}
```

C
```
#include <stdio.h>
int main(){
    printf("Hello world ! \n");
    return 0;
}
```



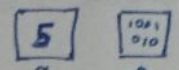
- compiler knows which & how much data to access with help of data type

- IDE / Code Editor
  ex: VS code, sublime text, replit

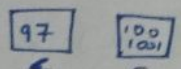
→ Datatypes (helps → type of data stored; → memory space)

- built-in / primitive → int, float, void, etc
- Derived → arrays, pointers, references
- user-Defined → structures, unions, classes, enumerations

int a = 5
- variable
- datatype
- value



char c = 'a'

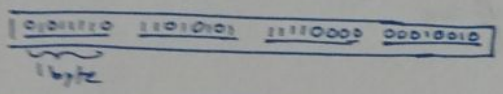


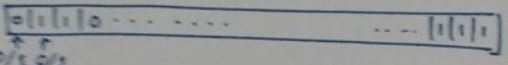
char → 1 byte = 2^8 - 1 values
short → 2 bytes = 2^16 - 1 "
int → 4 bytes = 2^32 - 1 "
long → 8 bytes = 2^64 - 1 "
float → 4 bytes = 2^32 - 1 "
double → 8 bytes = 2^64 - 1 "

1 byte = 8 bits

bits: 0 1

int → 4 bytes = 32 bits

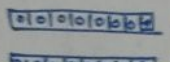




↳ 2 → 2 ..... (32 times)

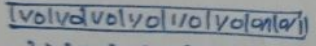
↳ 2^{32} combinations

bool flag = true;



memory 1 byte coz can't assign smaller memory than this

char → 1 byte = 8 bits

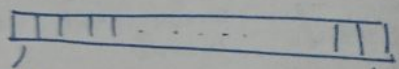


↳ 2^8 combinations

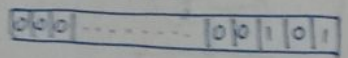
char → Ascii value
a → 97
b → 98

1 - 1
2 - 10
3 - 11 ← 2 bit max no
4 - 100
5 - 101
6 - 110
7 - 111 ← 3 bit max no
8 - 1000
9 - 1001
10 - 1010

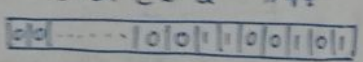
sizeof(a)
↳ function to show bytes taken to store



→ How data is stored
int a = 5



char c = 'a' //97



1's complement
- flip the bits 0's & 1's
  0 = 1
  1 = 0

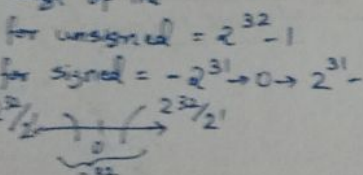
```
  0000 .... 0101
= 1111 .... 1010
```

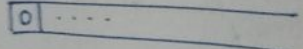
2's complement
- add 1

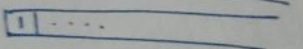
```
  1111 .... 1010
           + 1
-------------------
= 1111 ..... 1011
```

Signed = +ve & -ve
unsigned = only +ve

- Range of int
  for unsigned = $2^{32} - 1$
  for signed = $-2^{31} \to 0 \to 2^{31} - 1$



for +ve 

for -ve 

how -ve value is stored ?
int a = -5
i. ignore -ve sign; a = 5
ii. binary equivalent
iii. 2's complement

n bits

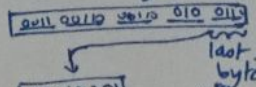
signed = -2^{n-1} to $2^{n-1} - 1$
unsigned = 0 to $2^n - 1$

→ Type Casting / Type Conversation
- conversation of one data type to another in a program either automatically by compiler (implicit type conversion) or manually by programmer (explicit type casting)

implicit : char c=97;
```
cout<<c;  // a
int c='b';
cout<<c;  // 98
```

explicit : float a=(float)2; //2.000
// variable = (type) expression;

Char ch = 234342
// will not give error



last 1 byte will be considered

10010101 → ~~character~~ x (no) char equivalent

int / int = int

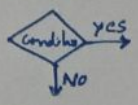
float / int = float

double / int = double

int / float = float

→ Operators
- Arithmetic → +, -, *, /, %
- Relational → >, <, >=, <=, !=, ==
- Assignment → =
- logical → &&, ||, !
- Miscellaneous → Ternary operator → condition ? x : y (if — then — else —)

→ Conditionals → hoops




```
for (int i=1; i<=n; i++){
    //body
}
```

~~while (i<=n)~~

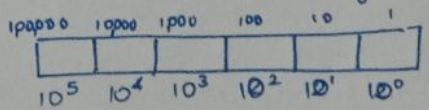
```
i=1
while(i<=n){
    //body
    i++;
}
```

```
i=1
do{
  //body
  i++;
}while(i<=n)
```

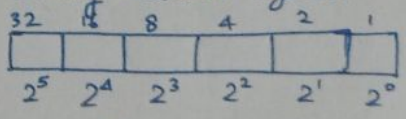
→ Break & Continue
- terminate the loop
- skip to the next iteration of the loop

→ switch (condition){
```
case 'a':
   cout<< "A"<<endl; break;
case 'b':
   cout<< "B"<<endl; break;
case 'c':
   cout<< "C"<<endl; break;
default:
   cout<< "zzz"<<endl; break;
}
```

Decimal Number System ⟷ Binary Number System



$1234 = 1\times10^3 + 2\times10^2 + 3\times10^1 + 4\times10^0$



$45 = 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

↳ 101101 (in binary)