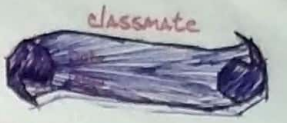


C Programming

- Middle level
- Fundamental lang.
- Base of other langs.
- Linux & Android mostly developed in C.
- User friendly.

Need of computation.
Industrial revolution.
Census calc.
Space race.

Q (Head & leg problem.)
A farmer is having few birds & cattles in his farm. The head count is 32 & the leg count is 100. The farmer exchanged few animals & now the new head count is still 32 but leg count is 104. Find no. of birds & cattle.

Birds $x + y = 32$.

$2x + 4y = 100$
$2(32 - y) + 4y$
$64 + 2y = 100$
$2y = 36$
$y = 18$
$x = 14$

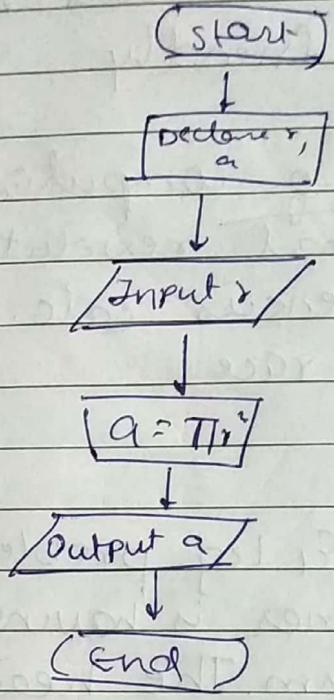
$2x + 4y = 104$.
$64 + 2y = 104$
$2y = 40$
$y = 20$.
$x = 12$

Cattle. Bird

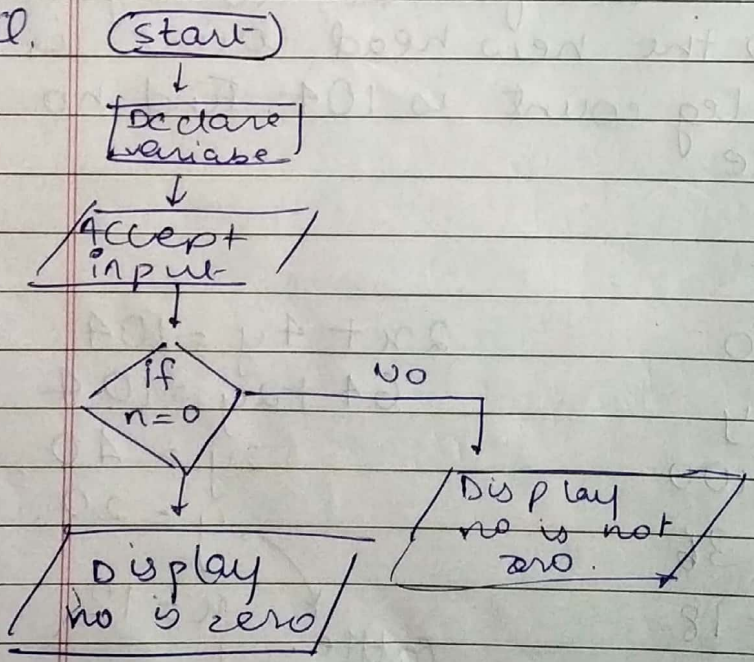
conio.h → header
↓ ↓
console input/output.

Q. write the algorithm to find area of circle.

1) Start
2) Declare r & a
3) Input r
4) $a = \pi r^2 (3.14)$
5) Display output a.
6) End

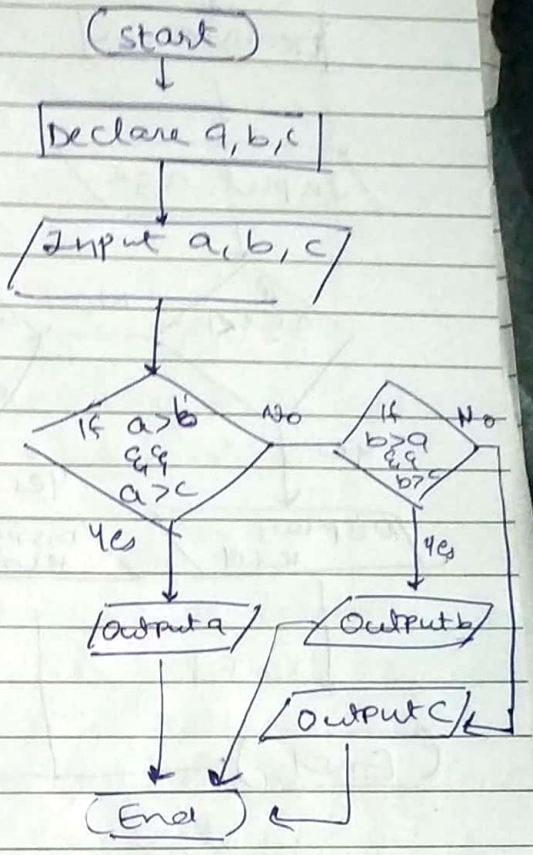


Q.



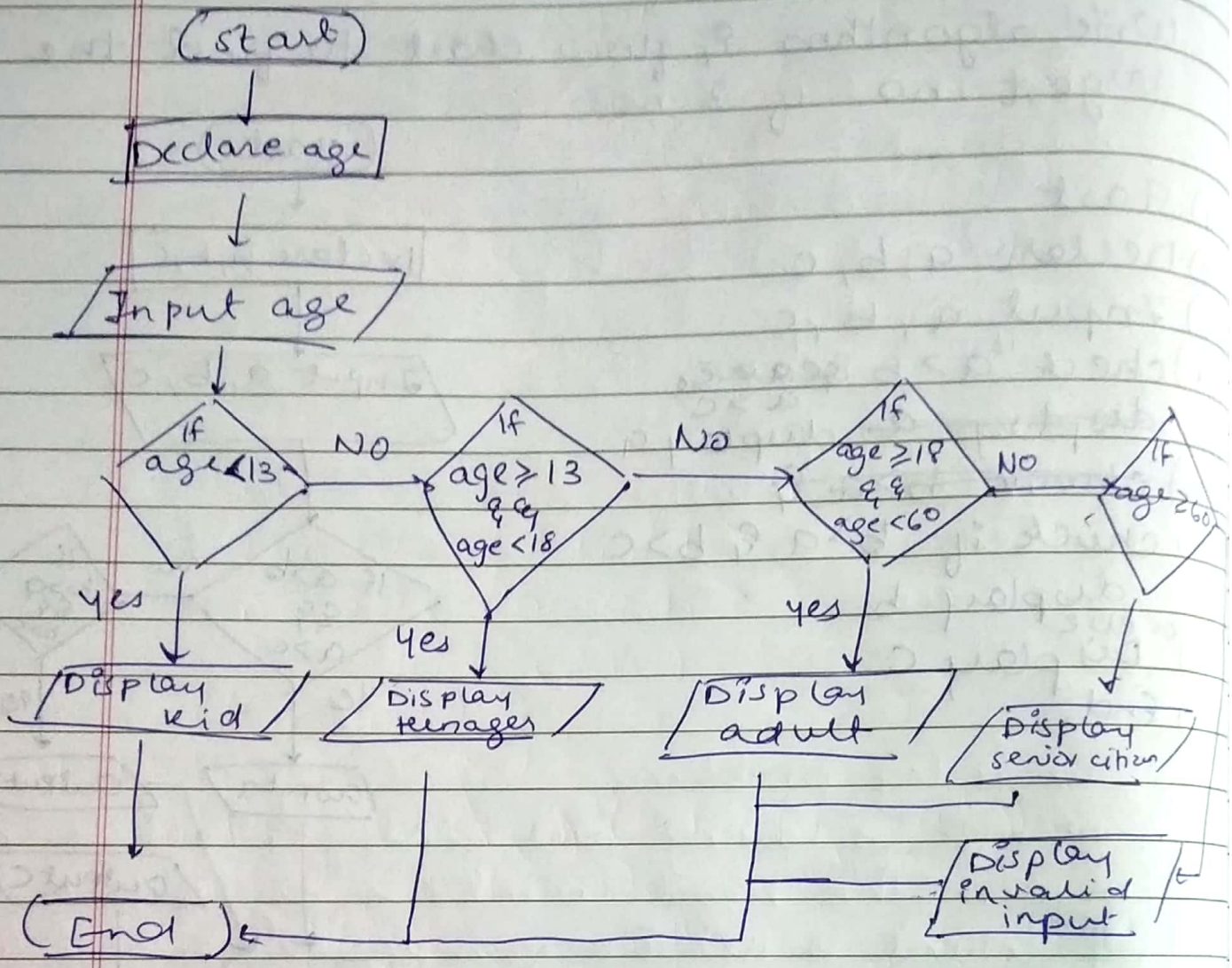
Write algorithm & flow chart to find the largest no. of 3 nos.

1) Start
2) Declare a, b, c
3) Input a, b, c
4) check if a > b && a > c, display a
5) check if b > a && b > c display b
6) else Display c
7) End.



Q. Write algorithm & flowchart to accept the age from user. If age is below 13, then display the user is kid. B/w 13 to 18 display teenager. Greater than 18 & less than 60, adult. Greater than 60 senior citizen.

1) start
2) Declare age.
3) Input age
4) Check if age ≤ 13, display kid
5) Check if age ≥ 13 && age ≤ 18, display teen
6) Check if age ≥ 18 && age < 60, display adult
7) Check if age ≥ 60, display senior citizen.



Q. Write algorithm & flowchart to accept single character code from user & display weather its vowel or consonant.

1) Start
2) Declare

Q2 Write algorithm & flow chart to accept marks for 3 subjects & find out avg marks of each subject

Q3 Write alogithm & flow chart to convert ~~farenheit~~ ferenheit temp to celcius.

Q4 Write alogrithm & flow chart to display if entered no. is +ve or -ve.

Q. What are the advantages of an algorithm.

Algorithms

step-by step procedure to achive some goal.

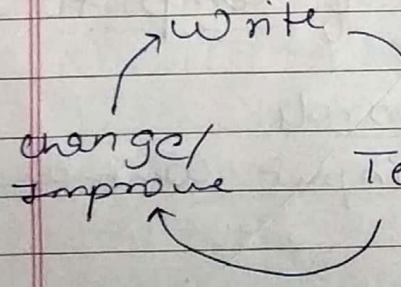
The relationship b/w input & output is called as cardinality.

There can be several inputs to one output & wise versa.

4 types of cardinality:
1 to many
many to 1
1 to 1
many to many

Algorithm is very precise & accurate while procedure is general steps.
Procedure is a generalized way of explanation while algorithm is specified way.

Writing an algorithm is a cyclic process.



Procedure V/s Algorithm.
- Algo is more time efficient
- leads to more consistency.
- Wastage of materials is less, so production cost decreases
- More profit by using algo.

→ where does the word Algorithm come from?
Al-khwarizmi - Persian mathematician, astronomer, astrologer, scholar

- An algorithm is a finite sequence of step by step discrete, unambiguous instructions for solving a particular problem.

- has input data & is expected to produce output data.
- Each instruction can be carried out in a finite amount of time in a deterministic way.

- Instructions in algorithm can be input, store, process & output.
store, process, output, input are used to deal with data processing problems

Algorithmic Representation of Comp ft's

Functions	Keywords
- Input Get info.	Get (input command)
- Storage store info	Given/Result Intermediates/set

- Process
  - Arithmetic — Let (assignment command)
  - Repete instructions — Loop
  - Branch conditionals — If

- Output
  - Give info — Give (output command)

→ Given - Used to describe (store) values given by user.

→ Result - Used to describe (store) the final output.

→ Intermediate - Used to describe (store) those outputs which are not "results" but are necessary as an input for further processing.

→ Set - Used to describe (store) a constant

"go" — Description
— Definition.

Algorithm Description.

1. Identify & name each Input / Givens
2. Identify & name each Output / Result.
3. Assign a name to algorithm (Name)
4. Combine the previous 3 pieces of info into a formal statement (Definition)

Results = Name (Givens).

Q. Write an algorithm to find the sum of 3 given numbers.

→ Let a, b, c be the given nos.
→ Let sum be the output
→ Let name of algorithm be addition.

Ans: Name: Sum3
Givens: N_1, N_2, N_3
Results: Total
Definition: Total = Sum3 (N_1, N_2, N_3)

METHOD
Get N1
Get N2
Get N3
Let total = N1 + N2 + N3
Give Total

Q. Write an algorithm to find result of division of X & Y.

Name: Division
Givens: X, Y
Result: Ans
Definition: Ans = Division (X, Y)

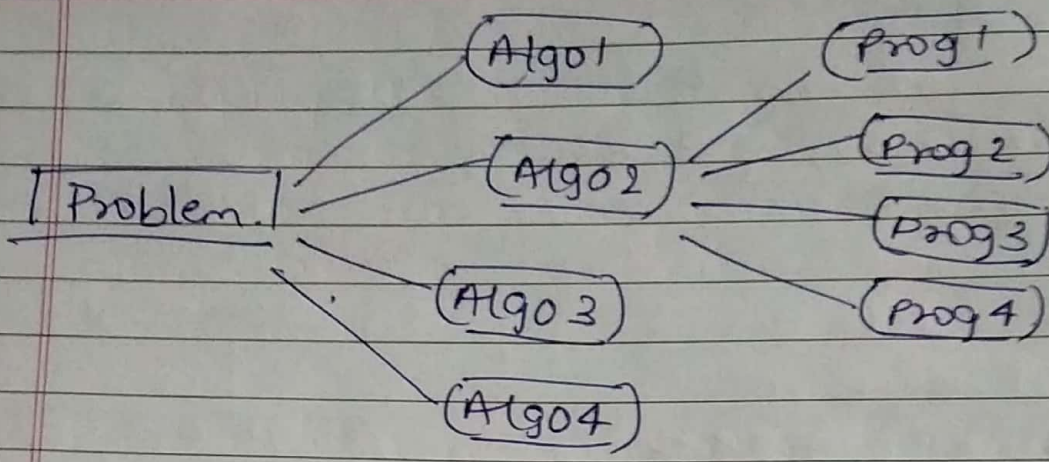
METHOD
Get X
Get Y
Let Ans = X / Y
Give Ans

given

Q. Find the sum & avg of 3 numbers.
Name: Calculation
Let: a, b, c
Results: sum, avg
Definition: sum & avg = calculation (a, b, c)
~~avg = calculation (a, b, c)~~

METHOD.
Get a
Get b
Get c
Let sum = a+b+c
Let avg = sum/3
Give sum
Give avg

Q. Write an algorithm to find the sum of product of 2 nos.
Name: Calculation
Let: a, b
Result: sum, product
Definition: sum & product = calculation (a, b)
METHOD:
Get a
Get b
Let sum = a+b
Let product = a * b
Give sum
Give product.



Q. WAP for area of circle

```
#include<stdio.h>
#include<conio.h>

void main() {
float r;
Printf("Enter r to find area");
scanf("%f", &r);
Prinf("Area of circle is %f", 3.14*r*r
}
```

→ 16 bit processor
2 bytes — int
→ 2^{16} bits.

char → 1 byte
→ $2^8 = 256$
0 — 255 ASCII value

float → 1 bit - sign
25 - fraction
last 8 — exponential

Iterative loop - for
do while
while

selective loop - switch case

for loop

```
for (initial; final; step)
{
}
```

→ fixed iteration

do while → exit controlled
executes atleast one
dynamic iteration.

```
do
{
} while (condition)
```

while → entry controlled
for sensitive data.
dynamic iteration

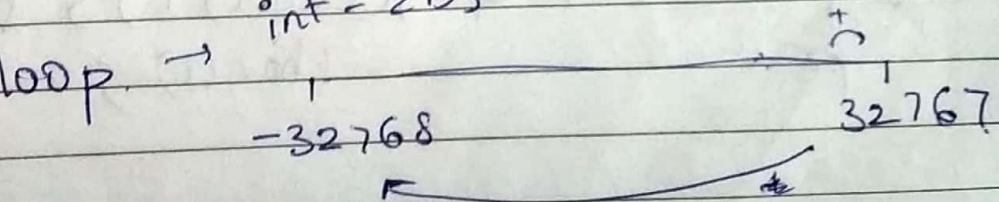
```
while ( ) {
}
```

infinite loop → for (; ;){ }

```
for ( i= 32766 ; i< 32768 ; i++ )
{ printf ("%d", &i);
```

infinite loop →

int - 2 bytes



complier has
- memory & time used

Q WAP to read non-zero signed int & display the followeing;
total no of positive nos, total no -ve nos, summation of +ve nos & summation of -ve nos.
→

Q Read 50 signed integers. Replace all +ve odd nos as 1 & all even -ve integers as 0.

```
int i, n;
for (i=0; i<50 ; i++)
{ scanf ("%d", &n);
 if (n%2==0 & n<0)
   printf ("0");
 else if (n%2==1 & n>0)
   printf ("1");
 else
   printf ("%d", n);
}
```

char i

Q Display all ASCII characters

```
for (i=0; i<256; i++)
{ printf ("%d", (char) i); }
```

char(i);

.dll — dynamic link library

watch ?

#include < >
public
pre processor directory

If header file is missing, error missing object module

.c → .obj → .exe
linked with lib.

#include "stdio.h" → search only current ~~all directories~~ directory
private

void → not necessary in c
necessary in c++ cause ft°

compile — f9
compile & run — Alt+f9

Array — storing group of similar data types.

\0 → null character generated by pressing enter at end of string.

&a → memory location of variable.

2d Array

```
for (int i=0; i<3; i++)
 {
  for (int j=0; j<3; j++)
  { scanf("%d", &a[i][j]);
  }
 }
```

matrix addition, sub
→ multiplication ?

program counter - points at which instruction is to be executed by compiler

Functions

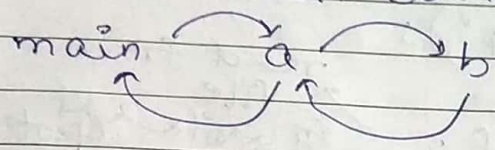
Starts execution from main pc
when it encounters a fn # moves to the location of fn & executes it. when it reaches return statement, pc will come back to memory location of main & fn.

Function callback.

Undeclared variable - A lookup table of all declared variables is used by compiler. If a variable is not found in this table, error is shown.

ftn types
- with input parameter & return value
- with input parameter & no return value
- with no input & return value
- with no input & no return value.

Function call cannot overlap



File management in C

File datat type
File pointer
File operation
(File x-tp)

Saving a file — continuous
— non continuous

Sequential file & Random Access file.
- read & write in sequence.
  - easy to track & implement
- read & write anywhere

~~File~~ (Records ~~in terms of blocks~~ stored ? ?.
Files are stored as blocks known as record.

data → file → record.

components
data
record (eg structure)
file

FAT - ~~File~~ File Allocation Table.

diff modes → r - read
→ w - write
→ a - append
→ r+ , w+
↓ ↓
read & write & write & read

file pointer

```
fp = fopen("abc.txt", "w+");
```

↑
non-negative integer

fscanf → read from file
fprintf → add input to the file

fclose(fp);

```
fscanf(*fp, "%d %d", &a, &b
```

getc(fp) → get character from file pointer.

getw(fp) → get word (integer - 2 bytes)

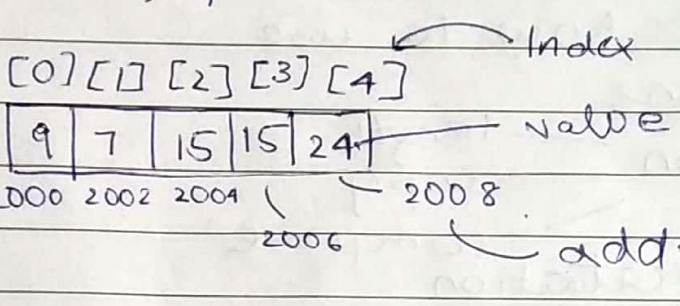
*

```
gedic abc.c
  ctrl+s
  ctrl+q
gcc abc.c -o abc
./abc
```

Write a code to identify entered no is divisible by 2 or 3 if it is not divisible then print "Enter the no which is divisible by 2 or 3."

Array is a collection of multiple homogenous elements which are stored in consecutive blocks of memory and can be accessed using an index.

3 things are needed
value, space, index.



MSB - most significant Bit

LSB - Least significant bit.

Syntax:
<type> <array name> [<array_size>]
int a[10];

struct - global ?
↳ dynamic memory ?
**
↳

malloc

structure of C
6 unique features
data types
variables / constants
Preprocessor directives → use → purpose
operators & precedence rule
control structures (if, if else, nested)
↳ syntax
iterative loops (for, while, do while)