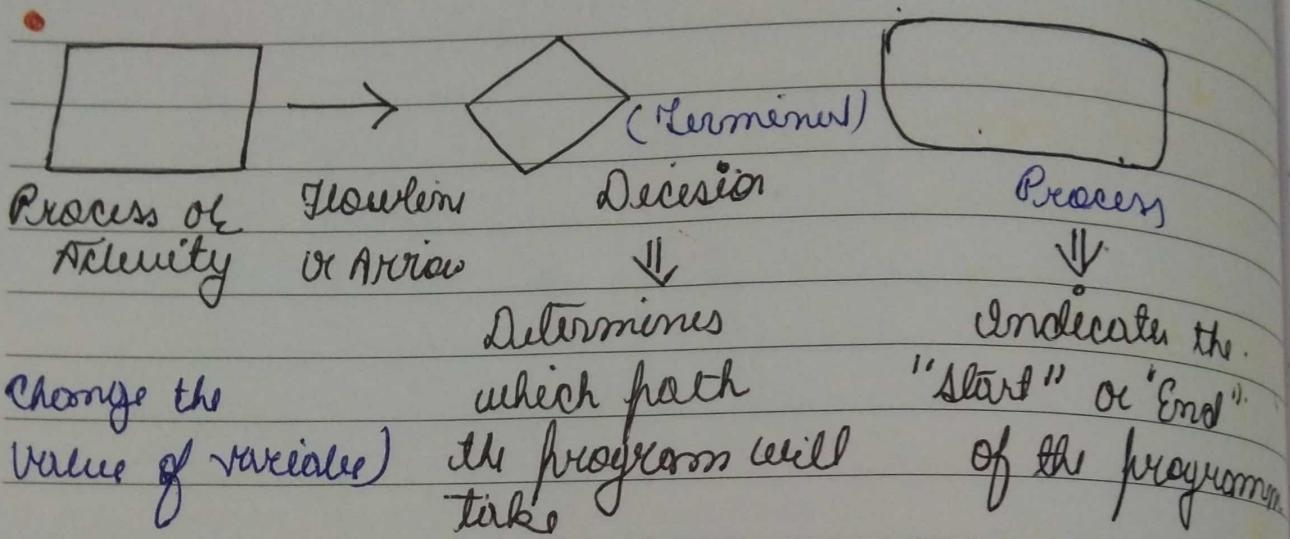


DATE ___/___/___

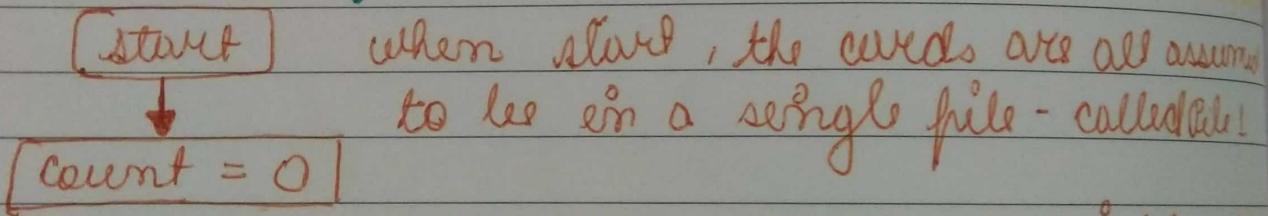
Notes:

- :- Computational Thinking :-

Introduction to flowchart :-



• Flowchart for counting cards.



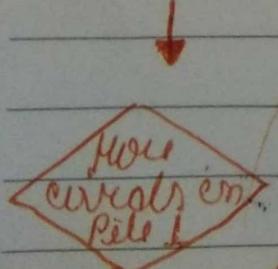
We initialize the count variable to 0

This is where the iteration starts.

We need to repeat the following steps.

- Pick a card from file 1
- Move it into a different file (say file 2)
- Increment the value of count

The iteration stops when file 1 is empty



We need to pick a card from file 1. But we can do that only if there are more cards in file 1 to pick.

Two outcomes are possible here:
either there are no more cards (condition = false) or there are more cards (condition = true)

→ NO (False) → End

If there are no more cards we can end the program. The variable count Yes (True) carries the required value.

Pick a card & from
file 1

If there are more cards in file 1
we pick a card from the file.
lets call this card x.

Move x to file 2

To make sure that we don't revisit
card x again, we move it to file 2.

(Increment count)

(After increment count)

we go back to the beginning of the
iteration +

General flowchart

Iterative steps :-

- Pick a card
- Move card to file 2
- Increment count

start

Initialise

unvisited
element

End

Yes (True)

Pick an unvisited element

Mark element as visited

Update variables

DATE ___/___/___

Notes:

S	M	T	W	T	F	S
---	---	---	---	---	---	---

Validity of Data :-

- some data

Basic Data Types :-

Boolean

Has only two values
True, False

Operation Result type:
AND, OR Boolean

Integer

Range of values is :-
..., -3, -2, -1, 0, 1, 2, 3

Operation Result type:
+, -, ×, ÷
<, >, = Boolean

Character type :-

Character :-

Values - alphameric

A B ... Z a b ... z 0 1 ... 9

Special character

, ; , : * / & % \$ # @ ! ---

Operation Result type
= Boolean

Subtype Data :-

Subtypes of basic datatypes:-

Integer :-

Range of values is

..., -3, -2, -1, 0, 1, 2, 3, ...

Operations

+, -, ×, ÷
<, >, =

Result type

Integer
Boolean

DATE - / /

Notes:

Reg No.: - Range of value is : ?
0, 1, 2, ..., Max.

S	M	T	W	T	F	S
---	---	---	---	---	---	---

Marks :- Range of value is ?
0, 1, 2, 3, ..., 100

Operations
??

Result type:
Marks, Boolean.

\times , \div do not make sense for marks.

+,-

Count :- Range value is :-

0, 1, 2, 3, ...

Result type .

Operations

Count
Boolean .

+,-
<,>, =

Character Data Type and its subtypes.

Character :- Value alphabets.

A B ... Z a b ... z o t ... g.

Special character.

! , ; : * / & ^ % \$ # @ ! ---

Operations
=

Result type
Boolean .

Gender :- Value? (M or F)

Operations
=

Result type .
Boolean .

DATE ___/___/___

S M T W T F S

Notes:-

String datatype and its subtypes :-

String :- Values - any sequence of characters
Operation Result.
char in string? Boolean
= Boolean

Names :- Values string with no special characters.

City :- Values :- String with no special character.

Words :- Values :- string with alphanumeric and ', ; ---

Category :- Values : Can take only one of the following values:
"Noun", "Verb"
"Preposition", "Adjective", ---

Transformation of sub-datatypes :-

Integer to Dates :-

RANGE of values

0, 1, 2, 3, --- 365

Operation
+, -

<, >, =

Result Type
Marks, Boolean

S M T W T F S

Amount :- Range of values is 0, 1, 2, 3.

Operator :-

present
4, -

<, >, =

Result type :-

String -

Amount

Boolean

Quantity :-

Range of values is 0, 1, 2, 3.

Operations :-

present
+, -

<, >, =

Result type :-

String -

Quantity -

Boolean

• Record Data Type :-

Records And lists

Data type with multiple fields - each of which has a name and a value.

record (also called struct or tuple)

(// Marks card Address Para.)

• List :-

A sequence of data elements (for example a sequence of records)

Marks card list - Marks card Record Data type.

Paragraph card list - Worden Record Data type

Shopping Bill list -

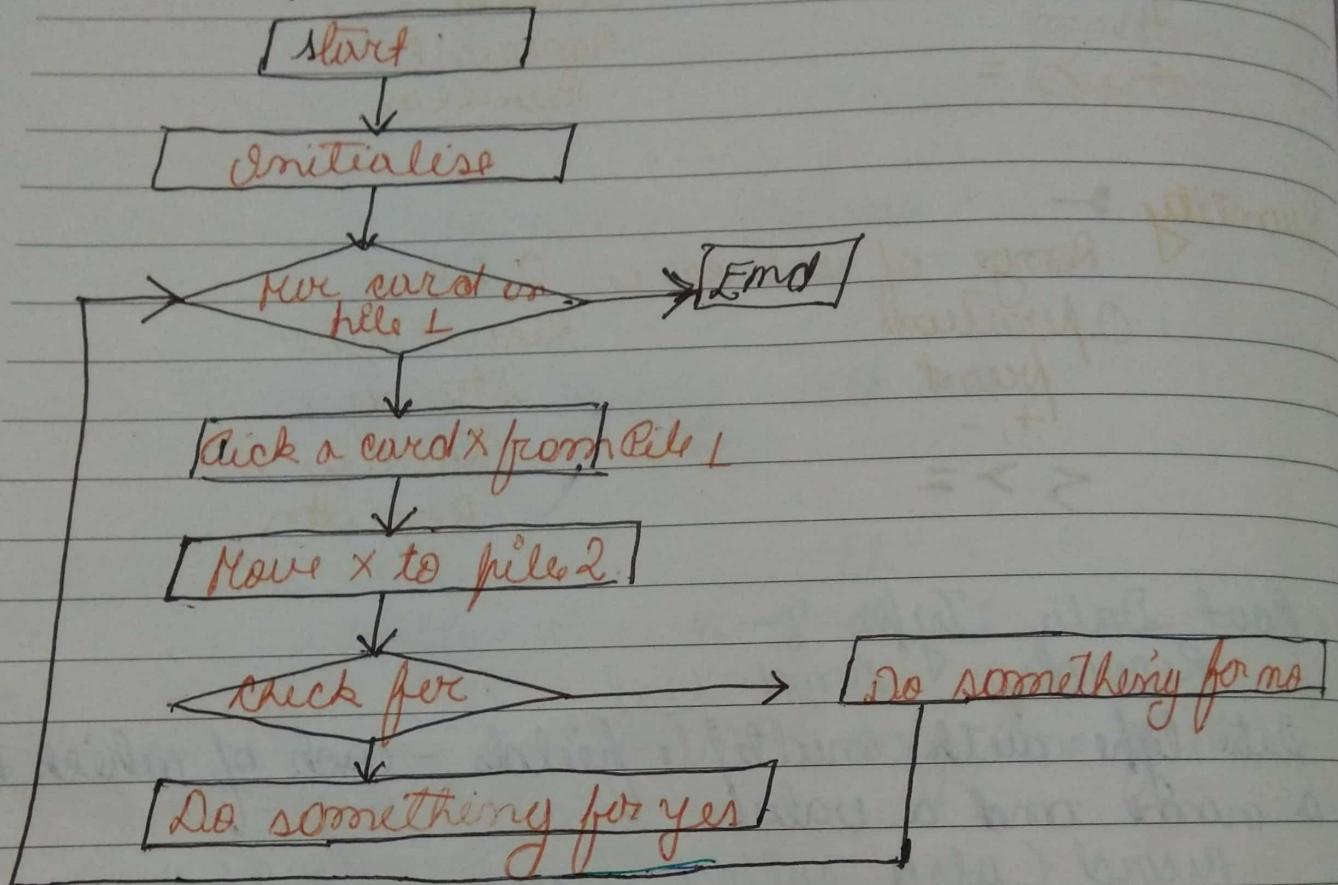
DATE 1/1

Notes:

S	M	T	W	T	F	S
---	---	---	---	---	---	---

Iteration with filtering

General formula.



S M T W T F S

[WEEK-3]

Presentation of Data sets in the form of a table :-

- Extracting Data from cards :-
 - Each card is a unit of information.
 - Diff. attributes and fields.
Card Id, Name, Gender --- Total
 - Organise as a table.
 - All the grade cards in a single table.

Summary :-

- Data on cards can be naturally represented using tables.
- Each attributes is a column on the table.
 - Each card is a row in the table.
 - Difficulty & if the card has a variable no. of attributes.
 - Items in shopping lists.
 - Multiple rows - duplication of data.
 - split as a separate tables need to link via unique attribute.

PROCEDURE (Functions).

- A procedure to sum up Maths marks.

DATE ___/___/___

Notes:

S M T W T F S

PSEUDOCODE :-

- Procedure SumMaths (gen)

Sum = 0

while (Pile 1 has more cards).

Pick a card x from file 1.

Move x to Pile 2

if ($x \cdot \text{Gender} == \text{gen}$) { }

 sum = sum + x Maths.

}

g.

Return (sum)

end sumMaths.

Procedure name : SumMaths.

Argument receives value : gen.

Procedure call SumMaths, implicitly with gen

Call procedure with a parameter sumMaths (F)

Argument variable is assigned parameter value.

Procedure returns the value stored in sum.

- A procedure to sum up any type of mark

Procedure sumMark (gen, f (d))

Sum = 0

while (Pile 1 has more cards) {

Pick a card x from file 1

Move x to Pile 2

if ($x \cdot \text{Gender} == \text{gen}$) { }

T M T W T F S

$$\text{sum} = \text{sum} + x \cdot \text{feld}$$

}
return (sum).
end summarks

- # Two parameter, gender (gen) and field (feld)
- # gen is assigned a value M or F to check against x.Gender
- # feld is assigned a field name to extract appropriate card entry x.feld
- # Single Procedure sumMarks to handle different requirement

→ SumMarks (F, Chemistry)
sum of Girls Chemistry marks.
→ SumMarks (M, Physics)
sum of Boys Physics marks

→ Calling A Procedure :-

• Use procedure names like math function, as part of an expression.

• Assign the # return value to a variable.

girlChemSum = SumMarks (F, Chemistry)

BoyChemSum = SumMarks (M, Chemistry)

if (girlChemSum) > (BoysChemSum) {

 "congratulate the girl".

use "congratulate the Boys".

DATE ___/___/___

Notes:

SUMMARY

- # A procedure may not return a value
- # Correct marks for one subject on a page
 - Procedure update marks (Card Id, sub, mark)
- # Procedure call is a separate statement.
 - update marks (17 Physics, 88.)

Summary :-

- > Procedure are pseudocode template that work in different situation.
- Delegate the work by calling a procedure with appropriate parameter
 - Parameter can be a value, or a field name
 - SumMarks (M, total)
- Calling a procedure
 - Procedure call is an expression, assign return value to the variable.
 $\text{girlChemMarks} = \text{SumMarks} (\text{F}, \text{chemistry})$
 - No useful return value, procedure call is separate statement I
 - Update Marks (17 Physics, 88.)
- > Procedure help to modularise pseudocode.
 - Avoid describing the same process repeatedly
 - If we improve the code in a procedure benefit automatically applies to all procedure calls

S M T W T F S

SIDE EFFECT OF PROCEDURES :-

- Side effect of Procedure modifies some data during its computation
- Sequence of cards may be disturbed.
 - Does it matter?
 - not in this case - adding marks does not depend on how the cards are arranged
- Sometimes the side effect is the send effect is the end good procedure to wrong card in decreasing order of Total marks
 - A side effect would be undesirable
 - We pass a deck arranged in decreasing order of Total Marks
 - After the procedure, the deck is randomly rearranged

* Interface Vs Implementation.

Each procedure come with a contract.

Functionality

- what parameter will be passed
- what is expected in return.

Data Integrity.

- can the procedure have side effects

DATE ___/___/___

Notes: _____

S M T W T F S

→ Is the nature of the side effect predictable?
For instance clock is reversed contract specifies interface.

- Con change procedure implementation (code) provides interface is unaffected.

Summary:-

- Need to separate interface & implementation
- Interface describes a contract
 - Parameters to be passed
 - Value to be returned
 - What side effects are possible
- Con change the implementation provided we preserve the interface.
- Side effects are important to be aware of
 - Sometimes no guarantee is needed (adding marks)
- Sometimes no side effect is tolerated (pronoun matching)
- Sometimes the side effect is the goal (sort the data)

DATE 1/1

Notes:

S	M	T	W	T	F	S
---	---	---	---	---	---	---

[WEEK - 4]

-° BINNING :-

- Importance of binning to reduce no. of comparison in nested iteration.

REDUCING COMPARISON :-

- Some computation seem to require comparison of each card with all other cards in the file
 - for example, choosing a study partner for each student.
 - the no. of comparison required can be very large.
- we observed that if we can organize the cards into bins based on some heuristic.
 - then we only need to compare ^{cards} within one bin.
 - this seems to significantly reduce the no. of comparison required.
- is there a formal way of determining the reduction in comparison?
 - calculate the no. of comparison w/o binning.
 - calculate the no. of comparisons with binning.
 - Use these calculations to determine the reduction factor.

DATE ___/___/___

Notes: _____

S M T W T F S

Comparing each element with all other elements

For element A, B, C, D, E

The comparisons required are :-

A with B, A with C, A with D, A with E (4)

B with C, B with D, B with E. (3)

C with D, C with E. (2)

D with E (1)

$$\text{No. of comparisons} = 4 + 3 + 2 + 1 = 10.$$

• For N objects, the no. of comparisons required

$$(N-1) + (N-2) + \dots + 1$$

$$\text{which is } \frac{N(N-1)}{2}$$

• This is same as the no. of ways of choosing 2 objects from N objects.

$$N_C = \frac{N \times (N-1)}{2}$$