Fundamentals of algorithm

1) Exchange of two values of two variables.

Statement - given two variables a and b, exchange the values assign to them.

→ to define the problem more clearly we examine a specific example

Consider, a ~~assign~~ Variables A and B ~~ag~~ assign a value as a = 721 and b = 463.

Development -

Starting Configuration

a	b
721	463

Distination or target Configaration :

a	b
463	721

to solve the exchange Problem we need to find a way of not distroging the old values of a or b.

To Do this we take a third Variable temparary Variable t & copy the original containt of a into t as

t := a. & a := b

& Now, After this Stage the contai-nts are

a	b	c
463	721	463

Now, we need to Assign the Value of a ~~with~~ which in t to b → b:=t.

Now, After this stage you are getting as

a	b	c
763	721	721

The algorithm can be written as

Algorithm Discription

1. Save the original value of a int
2. Assign a the " " of b
3. " to b the original Value of a that is stored int
4. Stop.

29-8-18

* Counting Algorithm

Q. Given a set of n students examination marks in the range of 0 to 100 make a Count of no. of students that pass the examination a pass is avoided for all marks of 50 and above.

classmate
Date 29-8-18
Page

Algorithm development –

Counting mechanism are very Friquently use in computer algorithms. Generally a Count must be made up of the no. of ~~utter~~ atoms in a set which posseses some perticular property or which satisfy some perticular condition.

For the abave problem suppose we are given a set of marks 55, 42, ~~75~~ 72, 47, 65, 89 to make a Count of passes For these said we can start at the left examine the First mark, see if it greater than equal to 50 if so remember one student has passed. than read the second mark examine it as marks are less than 50 No abjustment in count nexet third mark we see which is greater thah equal to 50 and so we add one to our previous Count. the process Continue until all marks have been tested. as shown below

Marks	Counting detail	
55	Previous Count = 0	Current Count = 1
42	Pre. Count = 1	Curr. Count = 1
72	Pre. Count = 1	Curr. Count = 2
47	Pre. Count = 2	Curr. Count = 2
65	Pre. Count = 2	Curr. Count = 3
89	Pre. count = 3	Curr. Count = 4

∴ no. of pass student = 4

The above description built the eqn as current value := pre. value + 1 —①
& pre. value := current value + 1 —②
these two steps can be repeatedly applyie to obtain the count required we exicute step ① followed by step ②, followed by step ①, followed by step ② & so on.

Hence the expresion become curre -nt value = current value + 1

Current value := prev. value + 1
Prev. value := current value

Current value := current value + 1
↑
new value := old value + 1

* Algorithm description

1) Prompt and Read number of marks to be processed.

2) Initialise count to zero.

3) While their are still marks tobe processed repetadlly do
   a) read marks

b) if marks >= 50 then add 1 to Count

4) Write total number of passes.

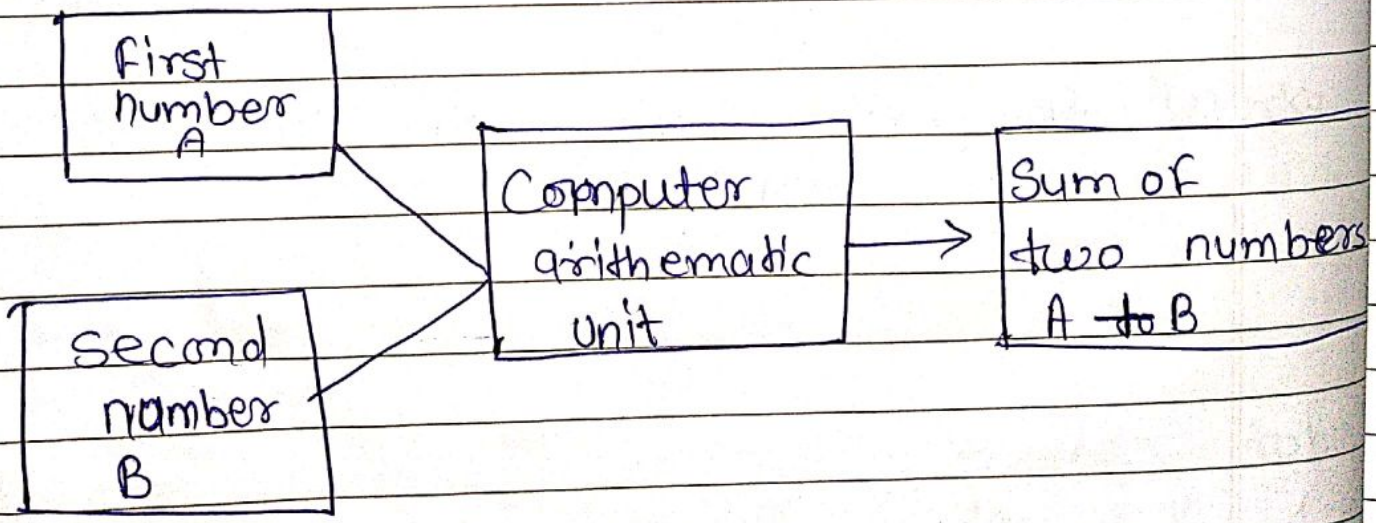
Q. Modify the above algorith to countit no. of pass & fail Student.

* Sumation of set of algorithm -

Given a set of n numbers design an algorithm that adds this numbers & writters the resultant sum. assume n is greaterthan or equal to zero ($n >= 0$).

Algorithm Deverlopement

This is an most fundamental think that we are likely to do add a set of numbers with Computer. In designing an algorithm for this task we must take the apporch that the computer device which acsept two nos to be added, perform the addition and writtens the sum of two no.s as shown in the figure.



classmate
Date 30-8-18
Page

For example we have to add 5 No.s one way to do these that takes note of the fact that computer adds two no. at a time hence $S := a_1 + a_2$ where, a_1 and a_2 are first two No's then add the third no. a_3 to the S computed in Step one & we get new sum as $S := S + a_3$

In Similar manner $S := S + a_4$

$S := S + a_5$

From step 2 on words we are actually repeting the same process over and over, the only difference is that Value of a & S changes with each Step For general i^{th} step $S := S + a_{i-1}$ ---(i)

this general Step can be place in the Loop to iteratively generate the sum of n numbers where 'n' is greater than equal to zero.

~~& Hence~~ Generally initially sum shall be zero and this Step is exicuted before Loop.

* Algorithm discription -

1. Prompt and read in the number how many numbers Summed.
2. Initialise Sum for zero numbers

3. While Less than n number have been summed repetedlly do. (while Counter <= n)

a) Read the next number (Read n).

b) Compute current sum by adding the number read to the most recent sum. (sum := sum + 1)

c) Count := Count + 1

4. Write out the sum of n numbers (write sum)

B.

* Factorial Algorithm -

Problem - given ~~the~~ a number n, Compute n factorial (n!) where n should be (n >= 0) greater ~~than~~ than zero.

Algorithm development -

We can start the developement OF this algorithm by examination of defination by n! given as $n! = 1 \times 2 \times 3 \times \dots (n-1) \times n$ (For $n >= 1$)

& by defination $0! = 1$

For developing the algorithm we need keep in mind that Computers

Arithmatic unit can multiply by two numbers at a time applying the factorial defination we get 0! = 1

1! = 1×1

2! = 1×2

3! = 1×2×3

4! = 1×2×3×4

⋮

we see that 4 factorial contains all the factors of three factorial the only difference is that the inclusion of no. 4.

with generlise $\boxed{n! = (n-1)! * n}$

using this defination we can write

1! = 1 * 0!

2! = 2 * 1!

3! = 3 * 2!

4! = 4 * 3!

⋮

if we start with P = 0! = 1 we can rewrite the few steps as P := 1

0! = 1 ⇒ P := 1

P := 1 * P ⇒ 1! = 1 * 0!

2! = 2 × 1! ⇒ P := P * 2

3! = 3 × 2! ⇒ P := P * 3

So, we get the general Statement as

$P := P * i$

this general step can be placed in a loop to iteratively generate n!

* The Complete algorithm description is as Follows.

1. Establish n, the ~~Functi~~ Factorial required. (Read n where $n >= 0$)
2. Set product P for 0! = 1
3. Also Set product Counter = 0
4. While Less than n product have been calculated repetadly do.
   (a) increment Product counter (counter := Counter + 1)
   (b) Compute the i^{th} product by multiplying i to the recent product.
   $(P := P * i)$
5. Write the result n! as P.

Date 8-9-18
Page

* Generation of Fibonacci (Numbers) Sequence.

Problem -

Generate & Print the first n terms of the Fibonacci sequence where n is greater than equal to 1. ~~The first~~

The first terms are 0, 1, 1, 2, 3, 5, 8, 13

each tearm beyeund to first to is derive from the sum of it's two ~~er~~ nearest predecessors

Algorithm devolepment -

From the defination we write that New term = Preceding term + term before proceeding term.

Let us define a as → ~~term~~ befor preceding term

& b as → preceding term.

& c as New term.

than to strat with we have

a := 0 First Fibonacci number

b := 1 Second FibonacciNumber

& c := a+b ← Third FibonacciNumber (From defin).

When the new term c has been generated we get the third Fibonacci Number. Similarly to generate the 4th or Next Fibonacci Number we need to apply same definition again. but before that we need to make some adjustment.

For example

The 4th Fibonacci Number is derived From sum of second & third Number.

With regard to defination the term before the preceding term is assign to 2nd Fibonacci Number or ~~New~~ A second Fibonaci No. & Third Fibonacci No. has the role of preceding no.

In Short we have to do

a) New term assumes the role of preceding terms & Currently preceding term assume the role of term before Preceding term.

In short rewrite

$a := 0$

$b := 1$

$c = a + b$ — New term

$a := b$ → term before preceding becomes preceding term

$b := c$ ← Preceding term become Newterm.

For Generation of no. of terms we have to follow same procedure again & again then they can be placed in a Loop to attratively generate fibonacci Number. The Loop will be execute N number of times.

Algorithm discription -

Algorithm for Fibonacci Number -

1. Prompt and read How many numbers to be generated. (Read n).
2. Assign First two Fibonacci Number a & b.
3. Initialise the counter [c:=0]
4. while Counter ↑ less than nos generated (c<=n).

c := a+b

[a = b, b = c] {
ⓐ write First two Fibonacci no.s
ⓑ generate next Fibonacci no. (Keeping a relevent)
ⓒ generate next fibonacci no. From more precent pair keeping be relevant For next competition.

ⓓ Update Counter by 1.

ⓔ Stop.

* Application -

This algorithm has practical application in botany, electrical network theory, Sorting & searching

* Reversing The digit of an Integer.

digit reversal is a technique that is some time use in computing to remove bias from set of nos. it is an important algorithm.

For example -

The Input is 1234

as we know that decimal no.s is multiples of 10^s power ↑ so we divide the no. by 10. we use two operator divisions and mod such that division gives quotient & mod operator gives remainder.

For reversing the digit every time we have to chop out the Last digit & which will we get by mod operator.

For example we can get the number

n = 1234 mod 10 = 4

1234 divide = 123

Here we drop out the ↓ number
Last

(digit) 4

apply this procedure as follows

r := n mod 10
n := n div 10

we get the digit 4
& The new no. 123

repeting the procedure till
n become zero.

12-9-18

* Some essential step in the algorithm are

1] Initialize the ~~quo~~ quoitent q to the decimal no. to be converted.

2) Derive the

Discription -

1) Establish n the positive integer to be reverse. (read n).

2) set the initial condition for reversed integer say variable ~~de~~ dreverse.
(Dreverse := 0)

3) While the integer is the being reversed is

greater than zero.

(While n>0) do

a) Use the remainder function to extract right most digit from integer.

digit := n mod 10

(b) Increase the previous reversed integer representation.

Dreverse by a factor of 10 and add to the most recently extracted digit

new dreverse = old dreverse * 10 + digit

~~de~~ dreverse = dreverse * 10 + digit.

c) Use the integer division by 10 to remove the Right most digit from the no. being reverse.

n := n div 10.

4) Write down the contents of dreverse

5) Write dreverse.

- Character to number Conversion -

Given the character representation of an integer ~~to~~ Convert it to it's Conventional decimal Format or vise-vica.

The no. of different characters needed to represent textual and String information is relatively small.

To represent the characters in decimal for a standred Code is adopted through out the world called ASCII Code

(american Standard Code For Information Interchange)

The ascII Code and their decimal equivalent are as below.

Character	8-bit code	(ASCII Code) Decimal Value
0	00010000	48
1	00010001	49
⋮	⋮	⋮
9		
A	01000001	65
B	01000010	66
⋮	⋮	⋮
Z		
a	01100001	97
b	01100010	98
⋮	⋮	⋮

Note that decimal digit 0, 1, 2, ---- are also represented in 8-bit character Code value.

As Every character has a ASCII ~~Code~~ Value and that will be converted into ~~bir~~ binary Value the following algorithm represent this ---.

- Algorithm :-

1) Establish the character string For Conversion to decimal and it's length 'n'.

2) Initialize the decimal Value to zero (dec := 0).

3) set base zero Value to the ASCII or (base 0).
ordinal value of '0' (zero)

4) While Less than n character have been examine to do.

a) Convert next character to the Corresponding ~~desi~~ decimal digit.
[string := "Dayanand"]

b) Shift Current decimal Value to the left one digit and add in digit For current Character.

5) Return decimal integer corresponding to input Character representation.