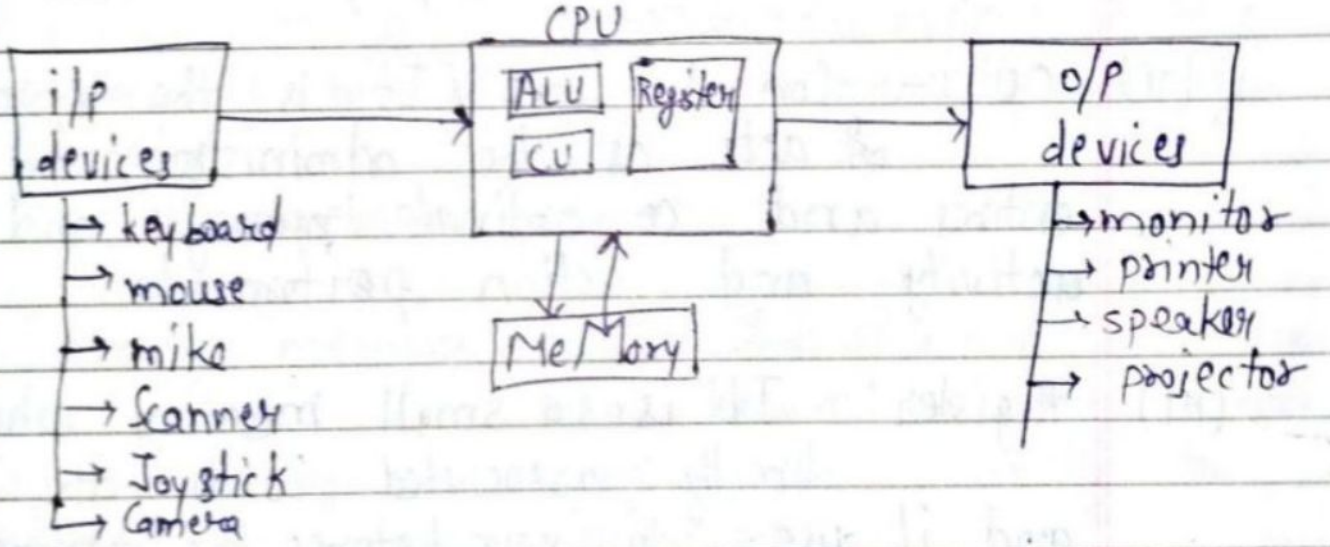
Programming for problem Solving

- Fundamentals ✓
- Numeral System ✓
- Programming ✓

Computer is electronic device which is processing & storing of the data.

Data :- Data is a raw facts & figures where as information is meaningful processed data.



1. Input devices :- These are the devices which are used to take the input (data) from user & pass into central unit. The Analog input such as press or mouse click generate the electric signal which then transform to digital form.
   eg- keyboard, mouse, scanner etc.

2. Output devices :- These are the devices which are used to received process data

by using a special device which emits ultra voilet rays.

3.) EEPROM ⇒ In this case the contains are eraser by using ~~bust~~ bursts of electric current their by making a more ~~reliable~~ reliable and enhances longetivity.

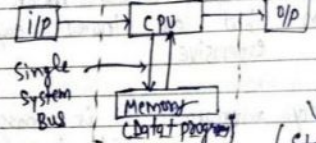
Cache memory ⇒ Cache memory is a type of memory which exist b/w RAM and Register and is used to store data and programs that are frequently used by the cpu. It is much faster than RAM but as lower capacity and is also expensive

Secondary memory ⇒ It is mass store device which is used to store user data and programs.
It is non-voletile and as a higher capacity compare to primary memory but is also slow and is inexpensive.
eg- hard disk, SD Card, Pen drive, CD/DVD, Magnetic tape/disk

very Imp ★ STORED PROGRAM ARCHITECTURE (SPA) ⇒

Also known as von Neumann architecture ~~des~~ is a architecture modal based on stored program Concept states that the data and the instruction which operate ~~of~~ on the data are store together in the memory.
Connected to the cpu through up single system Bus, the cpu would sequencly acces the data and program from the memory.
Due to a single system bus the cpu would only acces data and program one at the time. This problem is known as von-neumann Bottleneck and the solution to the problem is to use to seprate Bus for each data and program.

inwhich an instruction & data cannot occur at same time because they shared a common bus system.



Von-neumann (Stored pro. concept)

Algorithms :- Algorithum is a list of step in sequence to solve a problem. There are two ways to write a algorithms.

1) Pseudo code
2) Flow chart

1) Pseudo code :- It is a finite list of english like instruction in sequence to perform a specific task.

Q Write a pseudocode to make tea.
Solve- (i) Start
(ii) Add water in container
(iii) Put the container on top of lighten gas.
(iv) Add sugar and tea leave in container.
(v) Add milk to the container
(vi) Both for few time.
(vii) Stop

Rules of writing Pseudocode

(I) It must always 'START' and must always 'STOP'
(II) It must be finite and numbered.
(III) Each step must be self explainatory.
(IV) It must not contain ambiguous (Confuse) and long sentence.
(V) It may contain mathematical expression.

(i) Sequencly pseudocode

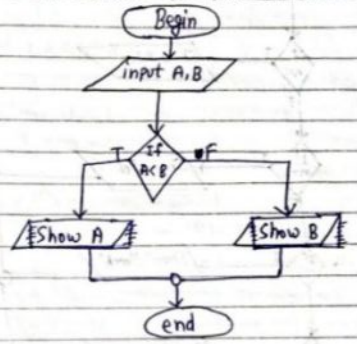
Q.1 Write an alogithum to add two number.
Solve. (i) Start (ii) Take two numbers (input A,B)
(iii) Add excepted numbers (C=A+B)
(iv) Show the result. (Show C)
(v) Stop.

Q. Write an algorithm to calculate area and circumference of a circle.
Solve. (i) Start
(ii) input R
(iii) ~~Declare~~ Set pie := 3.14
(iv) A = pie*R*R
(v) C = 2*pie*R
(vi) Show A and C
(vii) Stop

[Variable alfabate that I change that Variable of input]

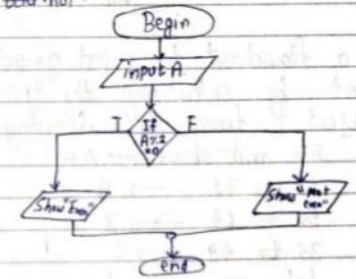
Q. Draw a Flowchart to find minimum b/w two numbers.

Ans-



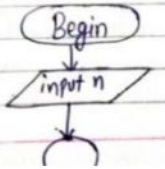
Q. Draw a Flowchart to check if a no. is even or ~~not~~. ~~odd~~ not.

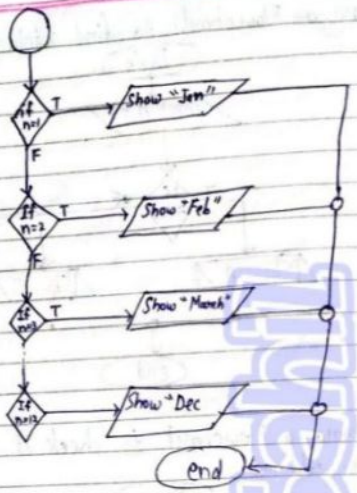
Ans-



Q. Draw a Flowchart to print Month of the year depending upon input

Ans-





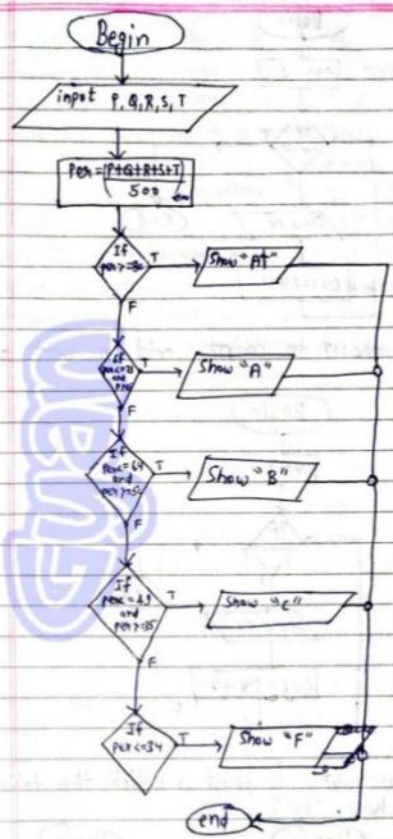
Q. Draw a flowchart to print grade of the student by calculating the percentage of 5 subject from the following Table.

80 and above → A+
65 to 79 → A
50 to 64 → B
35 to 49 → C
below 35 → F

Ans-

Begin

Begin



Q. ~~Write~~ Draw a flowchart to print number from 1 to 20.

Ans-

Syntax:

datatype variablename ;

OR

datatype variablename = value ;

eg- int num ;
num = 7;
float b = 7.68;
Char x, y, z;
y = 'M';

Operators & Expressions ⇒

Operators:- These are the symbols which are used to perform operation based on mathematics, comparison, evalution and deduction. operation can be unary (having one operand), binary (having two operand) and tersiary (having three operand). They are catagory into

(i) Arthmatic operators (ii) Relational operators
(iii) logical operators (iv) Bit wise operators
(v) Assignment operators (vi) Compound operators (Short hand)
(vii) Encriment operators
(viii) Decriment operators

1.) Assignment operators:- This operators represent = is used to asign value from RHS to LHS.

eg- num = 7 ;

int a, b, c;
a = 12;
b = 7;

2.) Arthmatic operators:- unary + → +a
unary - → -b

+ , - , / , * , %
a+b , a-b , a/b , a*b , b%a

3.) Relational operators:- These operators always evaluate to prove. true (1) & False (0)

unary !	c = !a	0
<	c = a < b	0
>	c = a > b	1
<=	c = a <= b	0
>=	c = a >= b	1
==	c = a == b	0
!=	c = a != b	1

4.) Logical operators:- These operators combine & evaluate relation expression.

(logical and) && (a>b) && (a>c) → T T T
(logical or) || (a>b) || (a>c) → F F F

5.) Compound operators:- These are short hand operator which represents entire expression.

a += b	a = a + b
z -= m	z = z - m
a *= b	a = a * b
p /= q	p = p / q
cy. = d	c = cy. d

6.) Encriment & decriment operators:- These operators are used to increase / decrease value of a variable by 1. They can be either prefix (applied before the operand) or postfix (applied after operand).

++a ;
--a ;

preFix incriment / decriment ⇒

eg- int a;
a = 7;
++a; // a = a+1
a = 8;

eg- a = 1;
--a; // a = a-1
a++;
a = 1

eg- int a, b;
a = 7;
b = ++a; → a = a+1; // a = 8
b = a; // b = 8

eg- a = -5
b = --a; → a = a-1 // a = -6
b = a // b = -6

Each specifier acts as a subsitution for exactly for a variable.

eg-
```
int a;
float b;
char c;
a=10;  b=7.5;  c='z';
printf("%d %f %c", a, b, c);
```

output
10 7.5 z

Escape sequences:- Use primary in printf are the character which have a special and are used to print characters that are not directly available.
They always begin with a back slash (\).

(i) \a - alert (ii) \t - tab
(iii) \n - newline (iv) \r - return to first colu
(v) \b - return to first column previous line
(vi) \v - vertical tab

eg- int a; , float b; , char c;

```
printf("%d \n\t%f", a, b, c);
```

output
10
7.5 z

Q. Write a program to perform math operation on a = 10 , b = 7

Ans-
```
# include <stdio.h>
# include <conio.h>
void main()
{
int a, b, ad, su, ml, di, mo;
ad = a+b;
su = a-b;
ml = a*b;
di = a/b;
mo = a%b;
printf("%d + %d = %d\n", a, b, ad);
printf("%d - %d = %d\n", a, b, su);
printf("%d * %d = %d\n", a, b, ml);
printf("%d / %d = %d\n", a, b, di);
printf("%d %% %d = %d\n", a, b, mo);
getch();
}
```

Output

10 + 7 = 17
10 - 7 = 3
10 * 7 = 70
10 / 7 = 1
10 % 7 = 3

Q. Write a program to calculate a simple interest.

Ans-
```
# include <stdio.h>
# include <conio.h>
void main()
{
int p, r, t, si;
p = 1.5;
r = 2.5;
t = 0.5;
si = (p*r*t)/100;
printf("(%f * %f * %f)/100 = %f", p, r, t, si);
getch();
}
```

Output

$\frac{1.5 * 2.5 * 0.5}{100} = 0.01875$

Q. Write a program to calculate area & parameter of rectangular.

Ans-
```
# include <stdio.h>
# include <conio.h>
void main()
{
int l, w, ar, pa;
l = 5;
w = 2;
ar = l*w;
pa = 2(l+w);
printf("%d \n %d \n %d \n %d", l, w, ar, pa);
getch();
}
```

Output

5
2
10
14

```
printf ("Press number to continue, 0 to exist");
scanf ("%d", &n);
  if (n!=0)
  go to stmt;
printf ("%d", s);
}
getch();
}
```

nested loop ⇒ When a loop ~~repeats~~ occurs inside another loop it is a nested loop

nested for :— When for repeats inside another for it is called a nested for. The loop which is outside is the outer loop & the loop which is inside the other loop is the inner loop. For each value of outer loop the inner loop complete itself for example if outer loop goes from 1 to 10 & inner loop goes from 1 to 7 then for each value from 1 to 10 the inner loop 1 to 7 will run combining the total run to 10×7 = 70 steps.

eg-
```
int r, c;
for (r=0; r<10; r++)
  {
  for (c=0; c<10; c++)
    {
    printf ("%d", c);
    }
  }
```
row → outer loop
colum → inner loop

Q. WAP to print the following pattern

```
#include <stdio.h>
#include <conio.h>
void main()
{
int r, c;
for (r=1; r<=4; r++)
  {
  printf ("\n");
  for (c=1; c<=4; c++)
    {
    printf ("*");
    }
  }
getch();
}
```

Q. WAP to print a following pattern

```
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
int r, c;
for (r=1; r<=4; r++)
  {
  printf ("\n");
  for (c=1; c<=5; c++)
    {
```

```
printf ("%d", r);
}
}
getch();
}
```

or

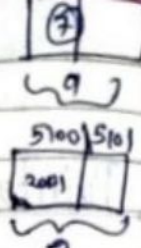
```
int r, c;
int m=1;
{
for (r=1; r<=3; r++)
  {
  printf ("\n");
  for (c=1; c<=4; c++)
    {
    printf ("%d", m);
    m++;
    }
  }
getch();
}
```

Q. WAP to print following pattern:—

```
1 2 3 4
5 6 7 8
9 10 11 12
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
int r, c;
int m=1;
for (r=1; r<=3; r++)
  {
  printf ("\n");
  for (c=1; c<=4; c++)
    {
    printf ("%d", m);
    m++;
    }
  }
getch();
}
```

```
7 →  printf ("%d", a);
2001 → printf ("%u", &a);
2001 → printf ("%u", p);
5100 → printf ("%u", &p);
7 →  printf ("%d", *p);
     getch();
}
```



Where & refer to address and the * refer to indirection operation which points to be the value inside the address.

Q. WAP to add using pointer.

Ay =
```
void main()
{
    int *p, *q;
    int a, b, c;
    a = 7;
    b = 5;
    p = &a;
    q = &b;
    c = *p + *q;
    printf ("%d", c);
    *p = 19;
    printf ("%d", a);
}
```

Q. WAP to multiple using pointer.

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

POPU

```
void main()
{
    int *p, *q;
    int a, b, c;
    a = 6;
    b = 5;
    p = &a;
    q = &b;
    c = *p * *q;
    printf ("%d", c);
```

Q. WAP to swap to no. using pointer & function.

Ay -
```
# include <stdio.h>
# include <conio.h>
void swap (int*, int*);
void main()                    // pass by address
{
    int a, b
    scanf ("%d%d", &a, &b);
    printf ("a=%d, b=%d", a, b);
    swap (&a, &b);
    printf ("a=%d, b=%d", a, b);
}
void swap (int*p, int*q)
{
    int c;
    c = *p;
    *p = *q;
    *q = c;
}
```

Ans = $(32)_x \rightarrow (?)_{10} = (33)_8 \rightarrow (?)_{10}$

$x^1 \times 3 + x^0 \times 2 = 5 \times 8^1 \times 3 \times 8^0$

$3x + 2 = 140 + 3$

$3x = 41$

$x = \frac{41}{3}$

Complement of a number ⇒ Primarily used in digital no. is the process to the obtain mathmethical substration using other math-methical operation.

For a certain base system with radix R complement can be either $(R)^n$ or $(R)^n - 1$ complement.

For example the binary no system their is 2s complement or 1s complement.

eg- for decimal their can be 10s complement or 9s complement.

eg- for hexadecimal their can be 16's complement or 15's complement

To calculate R complement we conclude $(R)^n - N$

To calculate R-1 complement we conclude $(R)^n - N - 1$

Here N is the number
n is no. of digits
R is the base or Radix

Q. find 9's and 10's complement of 437

Ans- 10's complement $(R)^n - N$

$(10)^3 - 437$

$1000 - 437$

$(563)_{10}$

9's complement $(R)^n - N - 1$

$(10)^3 - 437 - 1$

$999 - 437$

$(562)_{10}$

Q. Find R's & (R-1)'s complements of $(10111)_2$

Ans- R's complement $R^n - N$

$(2)_{10}^5 - (10111)_2$

$(32)_{10} - (10111)_2$

$(32)_{10} - (23)_{10}$

$(9)_{10}$

$(1001)_2$

(R-1)'s complement $R^n - N - 1$

$(2)_{10}^5 - (10111)_2 - 1$

$(32)_{10} - (23)_{10} - 1$

$(8)_{10}$

$(1000)_2$

Q. Find (R-1)'s complent of $(3A12)_{16}$

Ans (R-1)'s complement

$R^n - N - 1$

$(16)_{10}^4 - (3A12)_{16} - 1$

$(65536)_{10} - (14866)_{10} - 1$

$(50669)_{10}$

POPU Gray code

eg- $(1741)_{10} \rightarrow (?)_{\times 53}$

1	7	4	1
+3	+3	+3	+3
5	0	7	4

$(0101\,0000\,0111\,0100)_{\times 53}$

ASCII :— It is known as American standard code for information interchange. It is world wide popular code to represent char, symbol & number to binary. Each individual for ASCII is assigned a decimal value & a binary value. Initially a seven bit ASCII code containing character A to Z, Lower and upper case- special characters, special symbol), printable & non-printable. It is known a ASCII-7 which represent each char. using 7 Bits. further and additional & extentional version of ASCII is also present known as ASCII-8. Which uses 8 Bits to represent each character. It provides an additional 128 characters (printable & non printable)

000 0000 to 001 1111 → special non printable (0 to 31)

010 0000 to 010 1111 → special characters (32 to 47)

011 0000 to 100 0000 → number & math symbol (48 to 64)

100 0001 to 101 1010 → A-Z (65 to 90)

110 0001 to 111 1010 → a-z (97-122)

100 0001	— A	65
100 0010	— B	66
100 0011	— C	67
100 0100	— D	68
100 0101	— E	69
100 0110	— F	70
100 0111	— G	71
100 1000	— H	72
100 1001	— I	73

eg- I N D I A

73 78 68 73 65

100 1001 100 1110 100 0100 100 1001 100 0001

EBC DIC :— It is known as extended binary coded decimal for information code. It is an 8 bit character format to represent no., char. in system and was primary designed to be used with idol computer and can primary content 256 characters.

Purchase the Notes

100₹

Per semester
(All subjects)

Notes (Hand written) ✅
Most Questions ✅

All Branches

Min 100%
amount will go
into charity ⭐

For specific
Subject - 50₹

UPI ID -
sahilkagyan337@ybl

Er Sahil ka Gyan

🔥 Steps for getting NOTES and Most Questions -

👉Do payment using UPI ID -

sahilkagyan337@ybl

👉Take screenshot of transaction and send me on Email -

ersahildrive@gmail.com

Then finally access all Notes and most questions 🔥

Scan & Pay Using PhonePe App

SAHIL KHAN