Page No.
Date 01 12 23

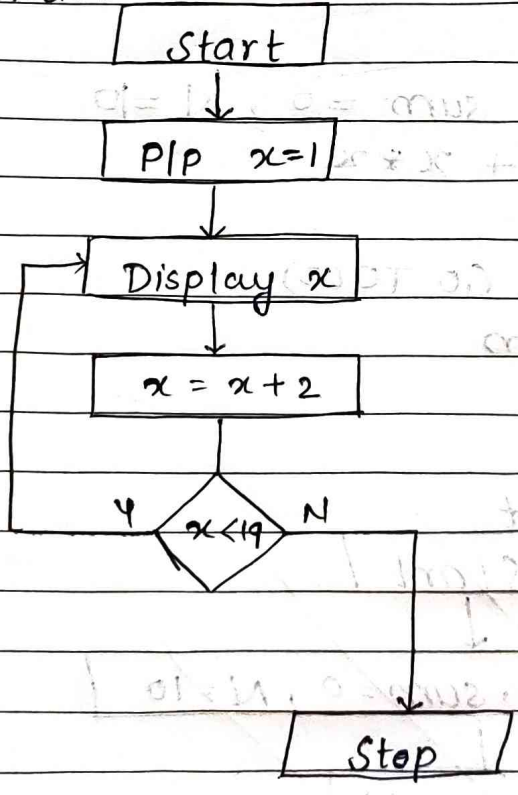
* Algorithm / flow chart

1) Write steps to generate following series
1, 3, 5, 7, ..., 19.

→ $S = \{x \mid x \in N \text{ s.t. } x < 20 , x \% 2 \neq 0\}$

i) Start
ii) Let $x = 1$
iii) Display x
iv) $x = x + 2$
v) If $x < 20$ GOTO (3)
vi) END

→ Flow chart



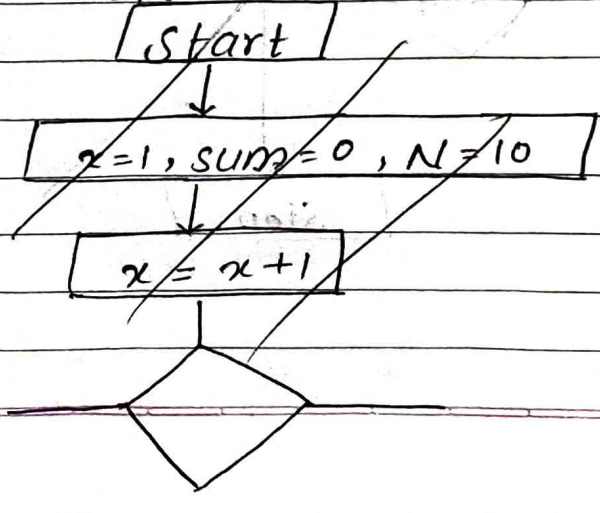
Q Write steps to find sum of first 100 odd numbers and also draw flow chart

i) start
ii) $x = 1$, $c = 0$, $sum = 0$
iii) $sum = sum + x$
iv) $c = c + 1$, $x = x + 2$
v) If $c < 100$ GO TO 3
vi) Display sum
vii) End.

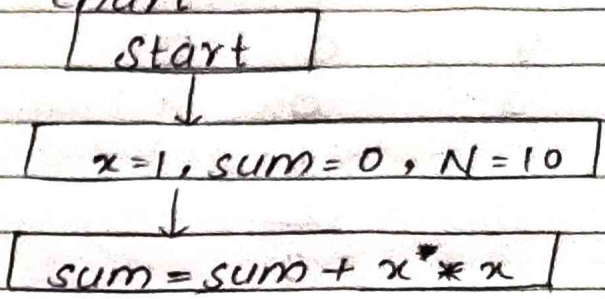
Q Draw a flow chart for following series.
$1^2 + 2^2 + \text{-----} + 10^2$

1) Start
2) Let $x = 0$, $sum = 0$, $N = 10$
3) $Sum = sum + x * x$
4) $x = x + 1$
5) If $x \leq N$, GO TO (3)
6) Display sum
7) END

→ Flow chart



→ Flow chart



Q Write algorithm for x/y, find Quotient and remainder if x > y

→ i) start
ii) Let x = 5 , y = 3 , Q = 0 , R = 0
3) x = x - y
4) If x ≥ 0, Q = Q + 1 , GO TO (3)
5) R = x + y
6) Display Q, R
7) END

→ Check

x	y	Q	R
~~7~~	3	~~0~~	0
~~4~~		1	
~~1~~		2	
-2		3	

$\frac{x}{y} \Rightarrow R = x - Q * y$

1) Start
2) $x = 5$, $y = 2$, $R = 0$, $Q = 0$
3) $R = x - Q * y$
4) $Q = Q + 1$
5) If $R > y$, GO TO (3)
6) $Q = Q - 1$
7) Display R, Q
8) END

x	y	R	Q
5	2	~~0~~	~~0~~
		~~5~~	~~1~~
		~~3~~	~~2~~
		1	~~3~~
		1	2

x	y	R	Q
3	4	~~0~~	~~0~~
		~~3~~	~~1~~
		~~3~~	0

Page No.
Date

* Elements of C

1- The character set:
a) All alphabets a - z
A - Z
0 - 9
All special characters.

2- Standard keywords:
There are 32 keywords in whole language.

3- Identifiers :-
1) can be arbitrarily long
2) Alphabets, underscore, number are allowed
3) Should begin with underscore or alphabet

4- They should not be key words

5- Data types :-

* Data types

Data type	Description	Memory	Range
int	Integer type	2 bytes (16 bits)	-32768 to 32767 -2^{15} to $(2^{15}-1)$
char	Single char	1 bytes	
Float	Floating no. (upto 6th decimal)	4 bytes	~~2^{-31} to~~
double	Double precision (14 digits)	8 bytes	

→ For increasing range we used unsigned short & long.

eg

int	16	-32768 to 32767
short int	16	//
long	32	-2147483648 ⇒ 2^{-31} to $(2^{31}-1)$
unsigned int	16	0 to 65535
unsigned short int		
unsigned long int		

* Escape Sequence (back slash)

\a	bell
\b	back space
\t	tab
\v	verticle tab
\n	new line

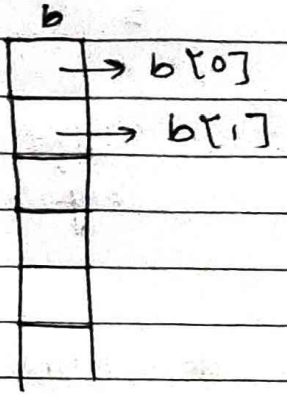
* Variable & Arrays

~~Arry~~ Arrays : Multiple storage

e.g.

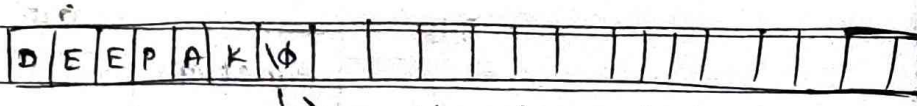
```
int a
int b[10]

char c
char d[20]
```

→ 16 bits



c = 'b'

d = "Deepak"

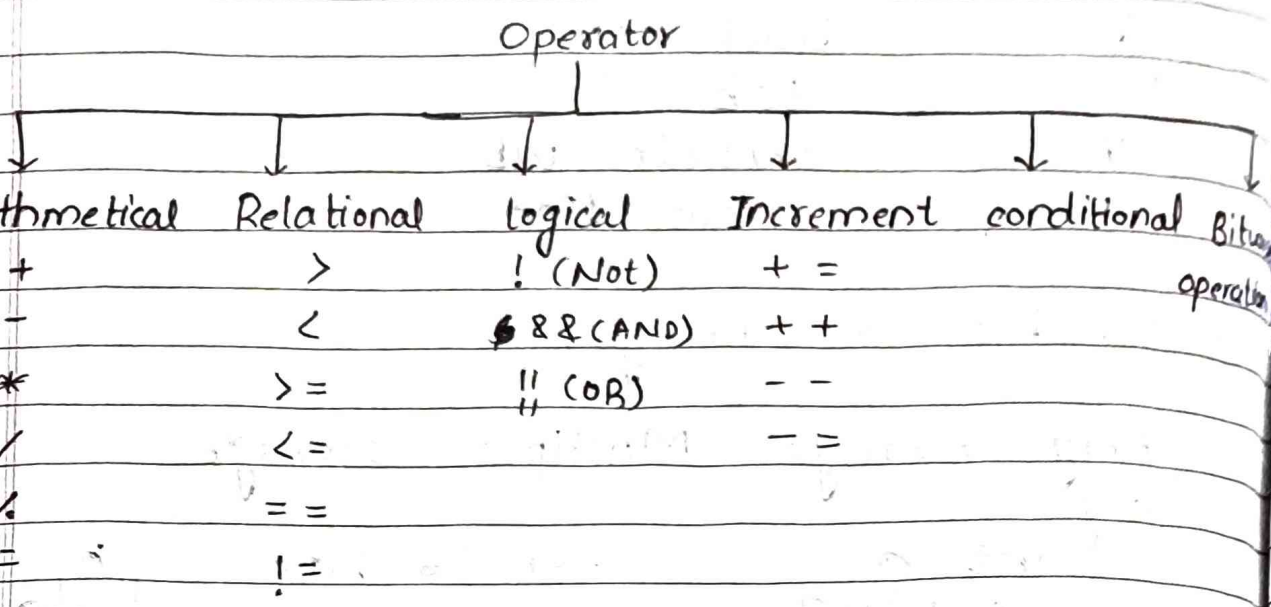


⇒ Operater

% Reminder Operator

To check whether given no. is prime or not

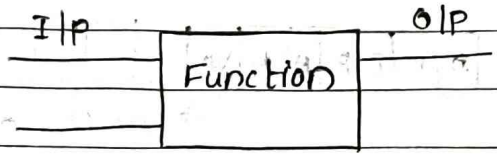
* Operator Expanssion and statements.



Left presidence
e.g : a = b

Right presidence
$\frac{d}{dx}(x^2+3+x^5)$

* Function.



(Argument) function-name (Arguments) {
↳ I/P
operation
}
Return O/P

e.g.

```
(int) addition (a, b) {
        c = a + b
        return c
}
```

Page No.
Date

* Input / output Statement (Function)

Library ( ) function-name ( ) {

return
}

Unlike other high level languages C does not have any bitten input and output state as a part of syntax.

The commenly used input and out put fun. are collectively called as

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

	I/P fun^	o/p fun.	data type
formated	scanf( )	printf( )	int, float, char
informated	getchar( )	putchar( )	
	getche( )	putch( )	char
	getch( )	putche( )	
	gets( )		

* Formated input function

~~St~~ Syntax

Variable = scanf("control string", variable-name
);

e.g. scanf("%3d, %6d", &n1, &n2);

Syntax

int n1, n2; float n3;

e.g. scanf("%3d, %6d, %2.3f", &n1, &n2, &n3);

Conversion character	Meaning
d	int
e	exponential |double
f	float
s	string
h	short int
c	
l	Long int

Syntax

printf("control string", variable names)

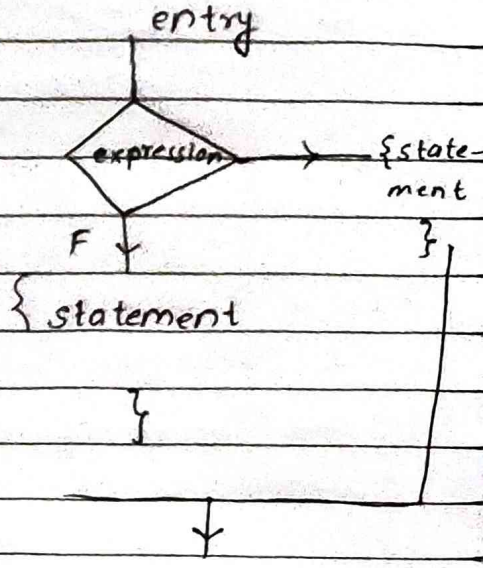
Page No.
Date

* Control Statement and Loops

If - else - continue - break , switch case
while
do - while
For

1) The 'if-else' statement

Flow chart



Syntax
if (expre

* Program (student passed or Failed)

```
# include <stdio.h>

main ( ) {

        int marks:
        print f ("\h enter marks : ");
        scan f ("% d", & marks);
        if (marks > = 35){
                print f ("\n passed");
                                }
        else { print f ("\n Failed");

                                }

                        }
```

2> Switch - case statement

Syntax :-

```
switch (expression){
                case value 1 : {
                                break;
                                }
                case value 2 : {
                                break;
                                }
                default : {
                                breaks;
                                }
                        }
```

We have to decide grades (value)

Credits	Grade
1	C
2	B
3	A
4	A+

```
# include <stdio.h>

main { int r; float m;
       print f("Enter roll no, marks");
       scan f("%d, %f, %d", &r, &m, &credit};
       switch (credit) {
        case 1: { print f("\n Grade = C"); break; }
       case 2: {print f("\n Grade = B"); break; }
       case 3 : { print f("\n Grade = A"); break ; }
       case 4 : { print f("\n Grade = A+ ); break ; }

                                }
                        }
```

* Loop in C

while loop

For loop

~~loop~~

→ while loop

Syntax

```
while (expression) {
        Block of statemer
        }
```

→ Example :-

1- Write a program to print square of first 'n' number.

```
# include < stdio.h>
  main ( ) { int n, s, i=1;
      print f ("Input n = " };
      scan f ("% d", & n );
      while (i <=n ) {
          print ("\n % d , % d", i , i * i );
          i = i+1
                }
        }
  Print ("\x END" );
```

i	n
1	5
2	
3	
4	
5	

Page No.
Date

Display screen

C:> Input x = 5

1	1
2	4
3	9
4	16
5	25

END

* The FOR statement

Syntax:

For (initialisation; test condition; increment/decrements) {

Block of statement

}

e.g. 2

Write a program using for loop to find $S = 1^2 + 2^2 + \text{----} + n^2$.

```
# include <stdio.h>
main(){ int n, s=0
print("\n input n= ");
scanf("%d", &n);
```

```
for(i=1; i<=n, i++){
        s = s + i*i        → s+ = i*i
print f("\n  s=%d" s);
        }
```

* Evaluate :-

n	s	i
4	0	1
	1	2
	5	3
	14	4
	30	5

s = 30

```
main(){ int  n , i=1 , s=0 ;
   scan f("%d", &n);

   while (expression) {

        s = s*i*i ,
        i = i+1
                    }

Print f("\n %d", s)
                }
```

Page No.
Date

* Functions

$$e^x = \sum \frac{x^n}{n!}$$

$$\sin\theta = \sum_{n} (-1)^n \frac{x^{(2n-1)}}{(2n-1)!}$$

$$\cos\theta = \sum_{n=1}^{\infty} \frac{x^{2n}}{n!}$$

* Expressions :-

Arithmetic operations :-

+ , - , * , / , % , ( )

while solving expression

1. Brackets
2. *, /, % Precedence - left
3. +, -

Q Write following expression in C format

1) $a^2 + b^2 - 2ab$

→ a*a + b*b - 2*a*b

2) $\left(\frac{x}{y} + \frac{P}{Q} - \frac{R}{S}\right)$

→ x/y + P/Q - R/S

3) $\frac{a+b}{c+d}$

→ (a+b) /(c+d)

4) $1 + \cfrac{a}{1+\cfrac{a}{1+a}}$

→ 1 + a | (1 + (a | 1+a))

Q Evaluate following expression.

1- 15 - 2*3 + 7 % 3 - 2

→ 8

2- 15 - (2*3 +7) % (3-2)

→ 15

Q Write a function (program) to ~~ev~~ integrate $f(x) = x$ over the interval (1, 2) using first principle.

→
```
# include <studio.h>
mean ( ) {
    int a, b, n, i; float dx, total = 0.0, y;
    float ~~fe~~ function (float)
    a = 1;      b = 2;      n = 10
    dx = (b-a) / n;
    for (i = a; i ≤ b; i = i + dx) {
        total = total + function (i); * dx;
    printf ("\n %f", total) }
}
```

```
float function (float k){
        return k.
                }
```

2. Write a program using user defined functions to find circumference and area of a circle (area.

```
# include <studio.h>
 main ( ){

int radius ;
printf (" Enter radius of circle");
scanf ("%d", & radius);
```

```
# include <studio.h>
# define pi 3.142
main ( ) {
 int n ;
float radius ; float Area (float), float
circum (float) ;
printf ("\n Enter radius = ");
scanf ("%f", & radius);
printf ("\n 1. Area \n 2. circumference");
printf ("\n select your choice = ");
scanf ("%d", & n );
switch (n){
case 1 ;
printf ("\n Area of circle = %f", Area (radiu
```

```
break;
Case 2:
printf("\n circumference of circle = %f", circum
        circum(radius),
breaking
default;
printf("\n wrong choice:");
break;
  }
    }
float Area(float r){

        return pi*r*r;
                }
float circum(float r){

        return 2*pi*r;
                }
```