Spectrum
DATE / /
PAGE NO.

Unit - I

Qution 1. What is a ~~Computer~~ Computer ?

Answer - Computer is an electronic device which takes data as Input proccessor that and give desire Results output.

A Computer can be define as a an electronic device which accept data perform the ~~request~~ require Mathmatical high speed and Output the Results

⇒ data and information -

- Computer accept data, proccess on it and produce information.
- Data Refers to some row pacts or figures and information is the proccessed data.

Exampal

date

If data 20/05/1906 is the ~~data~~ of birth the student than it is data.

When be proccess this data and say the age of students is 20 years then become as a information.

Characters stick of Computer.

1 Speed - C.P.U (Imega h_2), (giga h_2)(Units)
(MIPS) (MBPS) Units

Spectrum
DATE / /
PAGE NO.

2 Versitile → बहुत सारे कार्य एक साथ करना।

3 Accuracy → सटिकता

4. Automation ⇒ After build the programm Work as the task automatick ~~With~~ Which Repeted

5. Reliability → विश्वनियता

6. Availability → उपलब्धता

7. Stronge Capility ⇒ डेटा को अधिक समय तक सम्भाल कर रखना

8. Power of Remmembaring - ~~यदि~~ याद करने की क्षमता ~~दिमाग नही होता।~~

9 No I. Q. → सेम दिमाग नही होना।

10 ~~Ridigence~~ No ~~Feelings~~ → कोई भावनाऐं नही होती।

11 Dildgence →

Note - Calculator is deffrant from Computer become its ~~cant~~ Cant store the data

✱ Components	Processing
Input Device keyborde, mouse, Scanner, camera, microphone	

Components -

Inputs Device	Proccessing	Output devices	storage
Keybord, mouse Scanner, Camera microphone	CPU - Centad central Proccessing unit ALU - Arith metic logic unit	Speaker, Moniter printer, project- or, plotter	Pendrive, DVD = Digital Vesit DisK HDD - Hard DisK Drive SSD - Solide State drive

-> Explain the data processing Cycle of Computer.

- Computer proccessing is a proccess Which for Summerise data, analyes and Convert into useable information

- Becuse data are most usefull When it is Well Perseneted & informative.

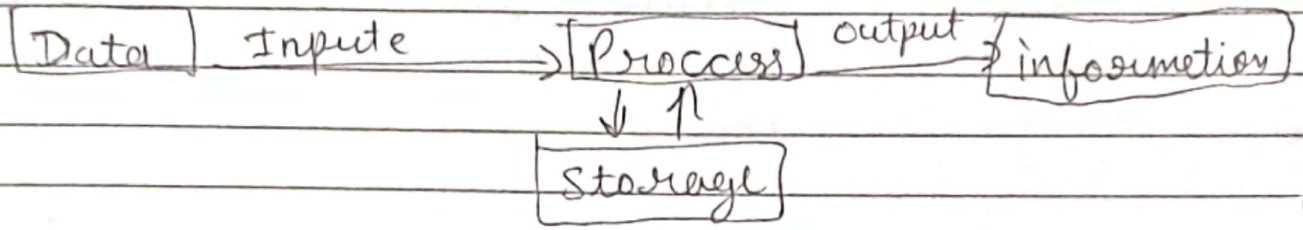
-> The data proccessing Cycle:-

- Data proccessing described all activites Which Common to all processing system from manual to electronic System.

- main aim (2184) is Convert the data meaning-ful information.

Spectrum
DATE / /
PAGE NO.

- Data proccessing system (DPS) are often refered to as information system
- Information system Typically take ~~raw~~ raw data as Input to produced information as Outputs.



- Data cycle Contaiint main four fuctions

* Data Input.

- The term Input refers to the aditivities required to record data.
- It's a proccess to ~~et~~ entered data into computer system

* Data Proccessing-

- The term proccessing include the activities like dossing storing Calculating, Comparing or summar is try the data.
- The Prokkess mean to use techquests to Convert data ~~to~~ into meaningful information

Spectrum
DATE / /
PAGE NO.

Data Storage -

inxolves the storing the data and ~~structure~~ information structior use

classification of the Computer by data proccessed

Computer are devide mainy three tipes

(i) Analog Computer (ii) digital Computer
(iii) Hybrid Computer

PAGE NO.:
DATE: / /

12/09/2025

C Programming language

~~Why We learn Cprogeramming~~

~~Peffinetion of C~~

Unit 3 c programming

Defination -> c progeraming language is the (gensol purpos, middle letter, procedral ~~tangeage~~) - devloped by 1972 Dennis Ritchie at ST Bill laboretory.

~~Histow~~ History of c progeramming language -> The root of All

~~A~~ ~~B Model~~

Mordern langaag (ALGOL (algorithmic lang.) desloped in early 1960 ALGOL is a first Computer lang. Whos a Block Structur ALGOL gave the Cansept of Structured progeramming to Computer ~~Science~~ Scincl

In 1967, In 1967 CBL

I 1967 CBL It was oder come the limletion of ALGO It is a multi-purpos programming langeug. It has Rechaer seht of data typs and Fnpt Outpat fasilits and supost of Syst am progeraming But it is

Complexed and Inefficient to Intimet of on the limited hardware,

In 1967 Martin Richareds Developed the language (BCPL) Basic combined programing lang. for Writing system Software It is a simplify Version of CPL it is highly portabl language.

In 1969-7 Ken Thompson created the language using many feturs of BCPL and called a B language We was early Version of Unix os operiling Systan at Bell Laboratrys

PAGE NO.:
DATE: / /

In 1972 C was ~~Involved~~ evolved for after ALGOL BCPL and be ~~Way~~ by Dahis Ritchie Bell leboratory C uses many Consepts from this & C languages and added of the Concepts of data type and other powerfull fiturs.
It is strongly acosiated with the Unix OS Unix is a one of the most popular network opreting System this bacic C languag is now as tradutional C'

4 the languag Becom more popular after publication of the go The C programming language by Brain Kernighan and denis dennis Ritchi in 1978 the book was so popular then the languag can to be non ~~to~~ has K & R C.

5. In 1989 ammerican netional Stendard Institut (ANSI) approved a Vesion of C Knon as (ANSI C)

6 It Was than approved by the International Standard Organisation (ISO) In 1990 and is Knon as C ISO/ANSI C or C90.

PAGE NO :
DATE : / /

Fith fetures of c language

1. C is a structured programing language With flow charts

2 C is highly portabl programing
langruag the program Written
on ofor one computer can
3 be russ for another With or Without
any modification

3. the program Return Written c
ar efficient and fast

4. C has sarsral predfined functions
C has only 32 key Words

5. C supports all bala conversion and
mix more opretors

6. Dynamic memory alocation is
posibal With C lang. ~~easyly memuples~~

7. C easyly menypet ~~by~~ byts/bits
addeses

8. C compiler Combines of the capabilitys
of asemdly level language With the
fiturs of high level languay
so it is called thas middle
level lang.

PAGE NO.:
DATE: / /

9. C has Ritch Set of opretors

Importent Poiats

1. C is casesansitive language c is ~~stene~~ stelments are entered in lower case letters.

2 every c Stetment must be terminated With ~~se~~ Semicolon (;)

3. C has program has many functions thair is one futuction ~~it~~ name main() is nacacry in all programs Without main We can't exicut c program.

4. A function name always followed by with a pair of Perenthesis ()

Bacic Structur of c progrems

1. Structure of C program -

(Documentetion Section)
link Section
Global Declaration Section
main ()
{
Local Declaration Section
Exculable section
}
User Defined function section

15/09/2025

1. documentation section → It Consist the set of Comment line. the document Section Consist the set of comment lines giving the name of the program. another name and other details

/* ——— */ used for mutiple Coment
used for single line Comments

Comments are non executable statements which are skiped by the Compiler

2 link Section → It provides Instructions to the compilor to link the ~~functions~~ functions from the System library

Preprocessor (#) → the Preprocessor is a Programm that proccess the source Code beforre it passes through the Compiler

~~# Include.~~

Include → It is a preprocessor file Inclusion directive and is used to Include header files

Syntext - # include <file name>

ex - # include <Stdio.H> } stendard
include <conio.H> } Input/output Header f

Spectrum
DATE / /
PAGE NO.

Stdio.H → Standerd Input Output header file
Conio.H → Console Input Output

Global Declaration Section → The Veriabls data that are used in morethen one function are known as global Veriabl and this Verabls are decaired delaired in the global deleareties sections

Main () Section → every c programm must have one main () function its this Saidvey its fast local decleration section and excutabl parts. the local declerections Section declaired all the Veriebls used in stetments. the stetment part Consist a Sequnce of exiculebal stetmes ts This two parts must appears betwee the opning and closing curly braces ({ }) - the programm exicution start as opning bress({) and end at the closing bress (}).

Sub programming Section → this Section Consis- of of the all the the users define function.

Spectrum
DATE / /
PAGE NO.

'a Simpal c programm.

```
# include < stdio.h>
# include < conio.h>
    int
Void main ()
          {
          Printf ("Welcome to cprogramm-
                  ing ");
```

C tokens →

1. The Smallest indivisual Units or elimehts in a program are toled token. there are five types of tokens.

1) Identifiee
2) Keywords
3) Constants
4) Opretors
5) Special characters

1 Identifires - Identifires can be define as a name of the Veriables, arrays, functions and same other Program elements using the combinations of the following characters.

alphabets → A to z
a to z

Spectrum
DATE / /
PAGE NO.

Digits → 0 to 9

Underscore → _ Cused as link between two words in long Identifire)

Note → the first Character and identifir Will be an atphaben or Underscore

2) the default Identyfire length is thirty two 32 Characters

(2) ~~Keyw~~ Keywords → Keywords ~~are~~ are the Words ~~the~~ Whos meaning as already ~~e~~ exptain to the compiler at the the time of designing a language some Words a Reserved to do specifir tasks ~~so~~ Such Words are called Keyword or reserved Words C has only 32 Keywords They are.

Key Words (C tokens)

1 Identifers → Auto default
break extern
case ~~float~~ float
Char for
Const for
Continue goto

Spectrum
DATE / /
PAGE NO.

if	Union
Int	Unsigned
long	Void
register	While
return	do
short	double
Signed	else
Sizeof	enum
static	
Struct	
Switch	
typedef	

3 constants → Constants Refers to fix Value that do not changed during the execution of a program. C Supports several typs of Constant

- Constant
  - numberic Constants
    - Integer Constants (2, 4, 7)
    - Real Constants (10.2, 4.6)
  - Characters Constants ('A', 'a', --)
    - Singel character Constants ('a', 'b', 'c')
    - string Constants ("Student name")

Spectrum
DATE / /
PAGE NO.

4. Operators → it is a Symbol which performe a particular operation, see has a Rich set of operators the following operators are available in See.

1. Arithmatic operator
2. Logical operator
3. Relational opreator
4. Bitwise opreator

5 Spacial Berectors → all cherectors other Then alphabet and digits are treated as spacial Symbols

ex , " , ' , () , { } , ; , < >

6 Variables → a Variable is a data name that may be used to store a data Variable Value. it is a named peace of memory which is used to Store data and access at Whenever require.

Unlike Constents a Varible may take differnte Values at different time during excution

Rules for naming a Variable In C →

- Variable name Conciet may Consist of letter

Spectrum
DATE / /
PAGE NO.

~~a~~ digital an underscore

- must begain with letter or a Underscor.
- ~~Its~~ Length ~~should~~ should not be ~~accent~~ exceed from 8 characters for good practice.
- Shaled not be a Key words
- White spaces are not allowed.
- The ~~name~~ name must be unique and meaningfull
- lowercase and uppercase are diffrent
- Valid Variable names are number, x, sum, first-no
  invalid - 123, group one, price $

Datatype → the type of data of data that ~~are~~ Variable may hold in a proggramming languag is ~~bold~~ datatype. C language is Rich in ats datatype

C datatyps can be classify into three types.

1. Primary / Primitive datatype
2. Derived data type
3. user defined data type.

Spectrum
DATE / /
PAGE NO.

Datatypes

- Primary datatypes
  - int
  - float
  - double
  - char
  - Void
- Derives dt
  - array
  - pointer
  - functions
- User defined dT
  - union
  - Structure
  - enum

① int datatype ⇒ It is used to store intiser numbers (positive, nagative, 0 Without decimnal part) We used int keyword to declair the intiser Veriabls it usess to 2 Byles (16 bit) Memory in storage

Ex ⇒ ---- -3, -2, -1, 0, 1, 2, 3 ----

② float datatype ⇒ floting point (Real) numbers are stored in 32 bit (4 Bytes) Withe 6 Six digits of presision thais are define by float keyword.

format Spacifier = %f

Ex 10.25, 20.60, 0.2445 ----

Spectrum
Date / /
Page No.

3. double datatype → a double datatype uses 64 (8 bytes), giving a preasison of 14 bits this are define by double data-type.

Format Specifier = %lf %.1f

exampal → 20564.289

4 char datatype → Singal caractar can be define as charactar type data. It uses ~~1 bit (8~~ 1 byte (8 bit) in storage. This are define by char Key-Word

exammpal = 'a', 'H', '*', '8'

5 Void datatype → Void are ~~no~~ known as empty datatype of c thay are use withe the functions which do not returnot any value or do not have any arguments

* Modifiers - this are used way with bacic datatypes to Modifi thair storing capacity

Syntext

Modifier	datatype
Short	int
long	int

Spectrum
DATE / /
PAGE NO.

1. Signed Modifire → It is a defaut setting means every Veriabl decaired is Outometicly ~~sine signed~~ . Signed type.

2. long Modifire → It alloceled some extra bits to normal Veriabls in orbar to increas its storage capacity.

or

long int.

3 Short → It is used to allocat half numbar of byits than the achual size to a Variable. Some of C Compilors alocat 4 bits to a normal Veriabl and 2 bytes for Short int Variable.

2 derived datatype → this ~~data~~ type of datatypes are derived from the existing datatype to make the programing more easye and decrease the comp- -lexcity and the laingth of the programm.

(3) User defined datatype → C awlsay provid programars to create to their on datatyps

Spectrum
DATE / /
PAGE NO.

according to the requirment of the logic at the time of coding the program.

	Data type	Size	Range
①	char	1byte	-128 to 127
②	unsigned char	1byte	0 to 255
③	short int / int	2byte	-32768 to -32767
④	long int	4byte	-2.147,483,648 to 2.147,483,647
⑤	unsigned int	4 byte	0 to 4.294,967,295
⑥	float	4 byte	~ 6 digit precion
⑦	double	8byte	~ 15 digit
⑧	long double	16byte	~ 18+ digit

declaration of Variables ⇒ In c proggramming a Varible declaration telles the Compilor what type of data a Variable Will Store and alocat allocats memory for it.

Synter →

Datatype - Var 1, Var 2, Var 3

Exammpal.

```
int    price ;
char   name ;
float  rate ;
double a, b ;
```

Spectrum
DATE / /
PAGE NO.

* multipal Veriabls of the Same time can be declaired togather

* We can gesign Values to a Veriabl after declaretion by using assigment opretor (=)

* ex → price = 200 Syntex
rate = 20.5 Variable name = constent;
name = 'd'

* We can oulsaw asign Value to Variabl at decearetion time.

ex int balance = 2000;
char grade = 'A';

basic inpute Outpute functions →

1. Printf() → most commanlly use Output () function of C Which can display a massge or anytipe Values Spacified by the formet Spacifier

Syntex - Printf (" Message");
Printf ("format Specifire", Variable Name);

ex : printf ("Welcome");
Printf ("%d", Price);

Spectrum
DATE / /
PAGE NO.

2 format specifier / control string ⇒ They are used with the Standerd output ~~funcf~~ function (Printf) and Standarded input function (Scanf). its Spacify the type of Values use in the input output opretion

3 datatype	formet spacifier
(i) integer	%d
(ii) char	%c
(iii) unsigned integer	%u
(iv) float	%f
(v) doubEl (long float)	%lf
(vi) string	%s
(vii) long intiger	%ld

~~Exampale~~ Exampale -

```
# include <stdio.h>
int main () {
        int a = 2;
        float int b = 4.;
        char int c = a + b;
        printf("%d
        double d = 54
```

Spectrum
DATE / /
PAGE NO.

Exampal

```
#include <stdio.h>
int main ()
        {
            int a = 2;
            float b = 2.5;
            char c = '*';
            double d = 54.6780;
            printf("%d", a);
            printf("%f", b);
            printf("%c", c);
            printf("%lf", d);
            return 0;
        }
```

\n = new line
(like (Raj) log)

Out put

2 2.5 * 54.6780

3. Escape Sequence character ⇒ the data to be displayed on Output Screen each is formated by using spacial source codes known as escape Sequance Charectars C supports sum spacial (esc) that are used in output func. function.

Ex =

\n - new line
\t - horizontal tab

Spectrum
DATE / /
PAGE NO.

\a = beep Sound
\f = form feed
\0 = null character
\' = print ' sign
\" = print " sign
\\ = print \ sign
\b = back Space

↳ Proggram - Write a proggram to print Hello in 5th line

```
# Include <stdio.h>
int main ()
        { Printf ("\n\n\n\n\n Hello ");
          Return 0;
        }
```

↳ Standerd input function & Scanf() = Another way of giving Values to a Veriable is to input data throw the Keyboard using the ~~Sc~~ Scanf() Which can accept any type of Values like intiger, float, characte ext.

Syntax →

Scanf ("format specifiers", & Variable, & Variable, ——);

eg

Spectrum
DATE / /
PAGE NO.

eg -

```
Scanf (%d, &a);
Scanf ("%d %d %d", &a, &b, &c);
```

Programming

```
# include <stdio.h>
int main () {
        int a;
        char b;
        float c;
        double d;
        Printf ("enter an intiger no");
        Scanf ("%d", &a);
        Printf ("enter a character");
        Scanf ("%c", &b);
        printf ("enter a decimal no");
        Scanf ("%f", &c);
        printf ("enter a double no");
        Scanf ("%if", &d);
        Printf ("given integer = %d", a);
        printf ("given character = %c", b);
        printf ("given decimal of %f", c);
        printf ("given double of %if", b);
        return 0;
}
```

Spectrum
DATE / /
PAGE NO.

Output

enter an integer no : 2
enter a character : Z
enter a decimal no : 10.2
enter a double no : 100.0005
enter given integer = 2
given character is Z
given a decimal no = 10.200000
given a double no = 100.000500

Q1 Write a program to add two integer numbers.

Ans 1
```
#include <stdio.h>
int main() {
    int a = 5;
    int b = 5;
    int c = a + b;
    printf("%d",
```

Ans 1
```
#include <stdio.h>
int main() {
    int a, b, c;
    printf("enter a no");
    scanf("%d", a);
    printf("enter a no");
    scanf("%d", b);
    c = a + b;
    printf("The sum of two
        no is = %d, c);
    return 0;
}
```

Spectrum
DATE / /
PAGE NO.

Output

enter a no: a
enter a no: b
The sum no of two no is = c

Opretors → Opretors are some methmetical and logical opretions on the ~~then~~ given values (operent) c Supports a Rich set of built in opretors. thay are classifid into three catagrues

(1) Unary → Which have one operand and known as unary Oprators

ex
Opretor) ++a
++ (inceriment ~~Opretors~~
-- (decriment Oprators) --a

(2) Binary → Which have two oprands by a binary ~~p~~ opretors.

ex
+, -, *, /, %,

(3) Trinary → Which have three Oprand as Trinery oprators

~~Got~~
~~Conditional~~ Opretor (? :)

Spectrum
DATE / /
PAGE NO.

Conditional operator
(a > b) ? true : false

- There are eight typs of oprators in C

1. Arithmatic oprators
2. Relational Operators
3. Lagical Operators
4. Assigment Operators
5. Increment Operators
6. Conditional ~~Opr~~ Operator
7. Bitwise ~~oprat~~ Operator)
8. Special operator

(1) Arithmatic Operator → C provide all the basic arthmatic operators four adition, supretition, Multification, ~~Revision~~, and ~~modulors~~ division modullors

They are - +, -, *, /, %.

~~Opr~~ Operators		
+	Additioned	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	division	a / b
%	Modulodivision	a % b (Remainder of the division Operators)

Spectrum
Date / /
Page No.

```
#include <stdio.h>
int main() {
    int a, b, c;
    Printf("enter two integer no");
    scanf("%d %d", &a &b);
    Printf("The sum of a+b = %d", (a+b));
    ;
    Printf("The subbration of a-b = %d", (a-b));
    Printf("The multiplication of a*b = %d",
    (a*b));
    Printf("the division a/b = %d", (a/b));
    Return 0;
```

② Relational operators → to compare to quantityses we use Relational opretors that are six ~~Relational~~ Relational opretors the value of Relational expression is either one (true or zero (false)

Operator	meaning	
<	lessthan	a < b
<=	lessthan equalte	a <= b
>	greater than	a > b
>=	greater than	a >= b
==	equalth to	a == b
!=	not equal to	a != b

Spectrum
Date / /
Page No.

Example
```
#include <stdio.h>
int main ()
{
    int a, b;
    printf ("enter two integar no");
    scanf ("%d %d", &a &b);
    printf ("a == b is %d", a == b);
    printf ("a != b is %d", a != b);
    printf ("a > b is %d", a > b);
    printf ("a >= b is %d", a >= b);
    printf ("a < b is %d", a < b);
    printf ("a <= b is %d", a <= b);
    return 0;
}
```

Logical Operators → This Operators are used for Combined the Result of two or more expressions.

Their are three logical Operators

&& Logical AND
|| Logical or
! Logical not

exp 1	exp2	exp1 && exp2
T	T	T
T	F	F
F	T	F
F	F	F

Spectrum
DATE / /
PAGE NO.

exp1 || exp2	!exp1	!exp2
T	F	F
T	F	T
T	T	F
F	T	T

```
#include <stdio.h>
int main() {
    printf(
        int a,b;
    printf("enter two integer numbers");
    Scanf("%d %d",&a,&b);
    printf("(a>5) && (b>5) is %d", (a>5)&&(b>5));
    printf("(a>5) || (b>5) is %d", (a>5)||(b>5));
    printf("!(a>b) is %d", !(a>b));
    return 0;
}
```

output
(a>5)&&(b>5) is 0
(a>5)||(b>5) is 1
!(a>b) is 1

oprelor->

4. Assignment ↑ this are used to assign a value of an expression to a variable. C has a set of shorthand Assignment Operators this are

+=, -=, *=, /=, %=

Ex		
	a = a + 10.	a += 10
	a = a * b	a *= b
	a = a % 5	a %= 5

Spectrum
DATE / /
PAGE NO.

(5.) Increment Operator/decrement Operator =

(1) Increment Operator (++) = the Operator (++) is used for Incrementing by 1(one) it means it adds one to the ~~opre~~ Operat.

ex → ++a = a+1

(1.) Pre Increment → ++a ⇒ a = a+1
(2.) Post Increment → a++ ⇒ a = a+1

(2.) decrement Operator (--) ⇒ the Simpal (—) minus ~~mark~~ is used for decrementing by 1 it means it Subtract 1 from Oprend. (ex --a = a-1

of

(1.) Pre Decrement → --a
(2.) Post Decrementop = ~~a~~ a--

(1) examples →

```
#include<stdio.h>/* Pre increment prograssm */
int main() {
    int x = 5;
    Printf ("befor increment x =
        %d", x);
    ++x;
    Printf ("after increment x=%d
        ,x);
    oreturn 0;
}
```

outpeit
befor increment = 5
afterincrement = 6

Spectrum
Date / /
Page No

```
/* Post increment program */
#include <stdio.h>
int main() {
    int x = 5;
    printf("before increment x
        = %d", x);
    x++;
    Printf("After increment x=%d
        , x);
    return 0;
}
```

Output

before Increment = 5
after Increment = 6

```
/* Pre decrement program */
#include <stdio.h>
int main() {
    int main x = 5;
    printf("befor decrement = %d", x);
    --x
    printf(" befor decrement = %d", x);
    return 0;
}
```

Output

Befor Decrement = 5
after Decrement = 4

```
/* Post decrement */
#include <stdio.h>
int main() {
    int x = 5;
    Printf(" befor decrement = %d", x);
    x--;
    print f(" befor decrement = %d", x);
    return 0;
}
```

output

Befordecrement = 5
Afterdecrement 4

```
#include <stdio.h>
int main()
{ int a = 5;
  Printf ("initial Value = %d", a);
  printf ("a++ = %d", a++); // Post increment
  printf ("Value of a after a++ = %d", a);
  printf ("++a = %d", ++a); // Pre increment
  printf ("Value after ++a = %d", a);
  printf ("--a = %d", --a); // Predecement
  printf ("Value after --a = %d", a);
  printf ("a-- = %d", a--); // post decrement
  printf ("Value after a-- = %d", a);
  Return 0;
}
```

Output

a++ initial Value = 5

a++ = 5

~~after of a a++~~ Value of a after a++ = 6

++a = 7

Value of a after ++a = 7

a-- = 7

Value of a after a-- = 6

--a = 5

Value of a after --a = 5

Spectrum
Date / /
Page No.

6. Bitwise Operators → bitwise operator are used to perform bit level operations on the operate operand. the operator are first converted to bit level and than the calculation this performed on the oprets.

thair are Six types of Bitwise Operators

1. Bitwise AND (&) => performs bit by bit and operation.
   - Sets each bit to 1, if both bit are,

a	b	a & b
1	1	1
1	0	0
0	1	0
0	0	0

Spectrum
DATE / /
PAGE No.

2. Bitwise OR (|) → Perform Bit by bit or operation

a	b	a | b
1	1	1
0	1	1
1	0	1
0	0	0

3 Bitwise XOR (^) → Sets each bit to one if bits are different

a	b	a ^ b
1	1	0
1	0	1
0	1	1
0	0	0

4 Bitwise Not (~) =>

flips all bits
(0 -> 1) (1 -> 0)

a	~a
1	0
0	1

5. Bitwis left Shift (<<) → Shift is bits to the left, filling with 0's

a = 1010 a<<1

cx →

a = | 0 | 1 | 0 | 1 |

a<<1 = | 1 | 0 | 1 | 0 |

Bitwise Right Shift (>>) → Shifts bits to the right

eg - a | 1 | 0 | 1 | 0 |
a>>1 => | 0 | 1 | 0 | 1 |

① #include <stdio.h>
int main() {
int a = 5, b = 3;
printf("a&b = %d\n", (a&b)); // AND
printf("a|b = %d\n", (a|b)); // or
printf("a^b = %d\n", (a^b)); // x-or
printf("~a = %d\n", (~a)); // not
printf("a<<1 = %d\n", (a<<1)); // left Shift
printf("a>>1 = %d\n", (a>>1)); // Right Shift
return 0;
}

Output

a&b = 1
a|b = 7
a^b = 6
~a = -6
a<<1 = 10
a>>1 = 2

⑦ Conditional Operators (ternary operators) →

Syntax

Condition ? expression1 : expression2

This operator is used for small

Spectrum
DATE / /
PAGE No.

~~disison~~ decision making

if first expresion Returns true than it evaluat (execute) second expression and if first expression returns false then it ~~evaluse~~ evaluate third expression

Condition is a ~~error~~ boolean extention which ~~is a~~ gives either true or false.

a>b, b<c, a==b, a<=b

```
# include <Stdio.h>
int main ()
    { int a=5, b=3
printf ("a>b ? a-b : b-a = %d ", (a>b ? (a-b) : (b-a));
    return 0;
    }
```

Output

a>b ? a-b : b-a = 2

```
# include <Stdio.h>
int main () {

    int marks = 45;
    marks >= 50 ? "pass printf ("pass") :
        printf ("fail\n");
        return 0;
}
```

Output → fail

Spectrum
DATE / /
PAGE NO.

Spacial Operators →

1. Sizeof operators → it Returns the number of bit Occupied by a Variable or datatypes

Exampal →

```
# include <stdio.h>
int main () {
        int a ;
        doubl b;
        char c;
        printf ("Size of int = %d", sizeof (a));
        printf ("Size of double = %d", sizeof (b));
        print f ("Size of Char = %d", sizeof (c));
        Relurn 0;
}
```

Output →

Size of int = 0
Size of double = 0
Size of char = 0

Spectrum
DATE / /
PAGE NO.

2. Comma operator (,) ⇒ used to seaprat multypul expretions all expretions are evaluated left to Right

```
#include <stdio.h>
int main(){
    int x;
    x = (1,2,3,4)
    printf("Value of x = %d", x);
    Return 0;
}
```

Output

Value of x = 4

3. address of Operator Operator (&) ⇒ gives the mamory address of a Variabal.

Value

4. Value at address Operator (*) ⇒ Returns the Value of the address

```
#include <stdio.h>
int main(){
    int x = 10;
    int *p = &x;
    printf("address of x = %p\n", &x);
    Printf("Value of address &x = %d\n", *p);
    Return 0;
}
```

address of x = 000000 0000 62FE 14

Value of address = &x = -10

Spectrum
Date / /
Page No.

Operator - Operator precedence & associativity

Operator precedence & associativity →

① Operator Precedence → defines which operator is evaluated first in an expression that has more than one operator

Exampal → a + b * c

here multiplication higar Residenc then addition

associativity → defines the order in ~~preadg~~ which operators of the same Residence are ~~evaluted~~ evaluated it can be left to Right (L→R) Right to ~~to~~ left (R→L)

Exampal →

```
int x = 5;
    y = 10;
    z = 2;
int Result = x + z + y;
```

Printf ("here ~~here~~ here + has left to Right associativity so it evaluvet has"

, (x+y) + z);

output

· here + has left to Right assosiativity so it evaluvet has = 17

Spectrum
DATE / /
PAGE NO.

	Operators	Precedence	Associativity
(1.)	() []	1	L -> R
(2.)	++, --, !, ~, *(value at address) &, Size of	2	R -> L
(3.)	* (Multiplication), /, %	3	L -> R
(4.)	+, -	4	L -> R
(5.)	<<, >>	5	L -> R
(6.)	<, <=, >, >=	6	L -> R
(7.)	==, !=	7	L -> R
(8.)	& (Bitwise AND)	8	L -> R
(9.)	^ (Bitwise XOR)	9	L -> R
(10.)	| (Bitwise OR)	10	L -> R
(11.)	&& (Logical ~~AND~~ AND)	11	L -> R
(12.)	|| (Logical OR)	12	L -> R
(13.)	? :	13	R -> L
(14.)	=, *=, /=, %=, -=	14	R - L
(15.)	, (comma operator)	15	~~R -> L~~ L -> R

(1) $10 + 5 + 2$ — Pre $(10+2)+2) = (15+2) = 17$ (L-R)

(2) result = 100 / 10 * 5;

(3) int a = b = c = 10;

(2) associativity (R -> L)

~~((100/10)*5)~~ = (100/(10*5))

(100/50)

= 2

(3.) associativity

a = 10 b = 10 c = 10

Spectrum
Date / /
Page No.

Decision making in C = In C proggrams can choos which part of the code to execute execute based on some condition. this ability is called disegon making and Statments we fore it are called conditional Statments. This Statmenty are evaluated one or more condition and make diisyn Whether to execude a block of code or not.
fore exampal thair is a show that stats only wen shurten numdar of pepole are praseint in the audience So you can ~~how~~ Write a programm like

Programm →

```
# include <stdio.h>
int main () {
        int num = 100;
        // Conditional Code inside decision
          making Statments
        if (num = 50);
          { printf ("Start the show");
          }
          return 0;
        }
```

Output → Start the Show

Spectrum
DATE / /
PAGE NO.

In the above the program, the show only starts when the number of the repeat is of ">" or "=" so it is spacified in the of if Statment.

Types of Conditional Statments in C⇒

1) Simpal if ~~Statment~~ Statement.
2) Nested if Statement
3) if / else Statement
4) Nested if - else Statements
5) if - eles if ladder
6) Switch Statement
7) Jump Statement
   - break
   → Continue
   → goto

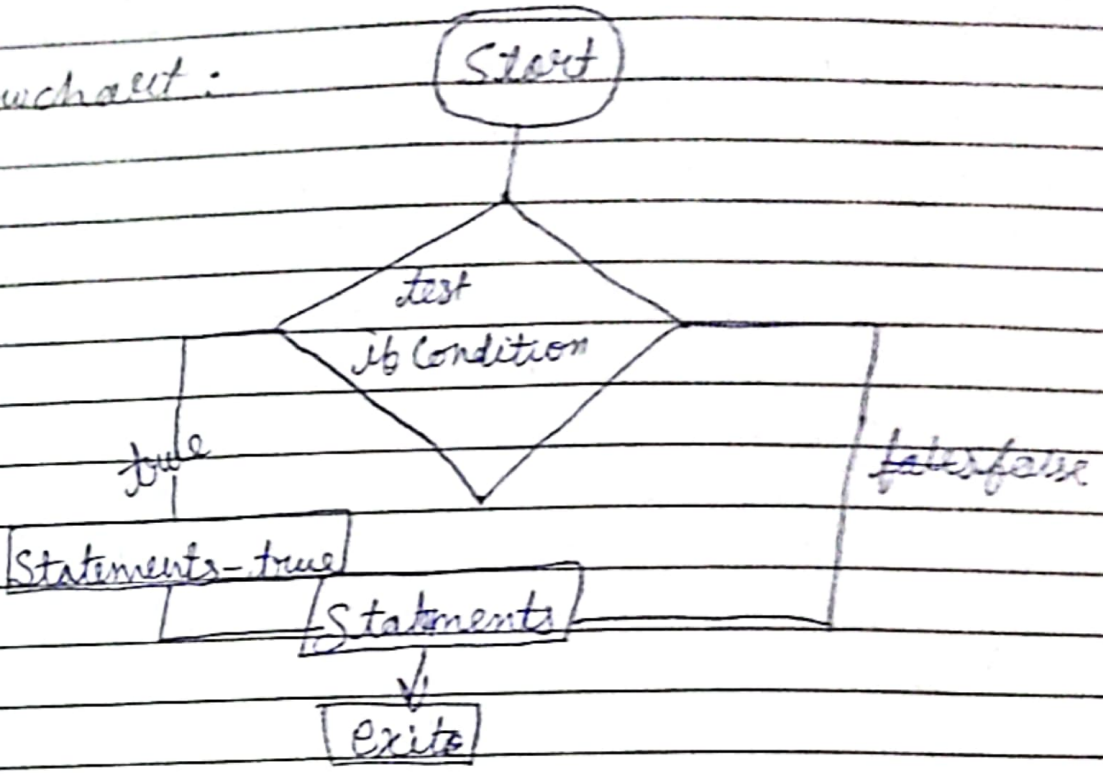
1) Simpal if Statement ⇒ It is used to decide whether a Surtent Statments and blocks of Statments will be executed or not if means if a Saten condition is true than a block of Statement is executed Otharwise ~~Knon~~ not.

Syntax ⇒

```
if (condition)
{
    Statements - true; // execut when -
    - if condition is true
}  Statment x(
```

→ hear Statements ~~-xxxx~~ Will be executed either condition ~~is~~ is true and false

Flowchart:



Programm →

```
# include <Stdio.h>
int main (){
        int i=10;
        if (i > 15){
        printf ("i is greater than 15");
        }
        printf ("I am not in if");
        return 0;
}
```

output

I am not in if

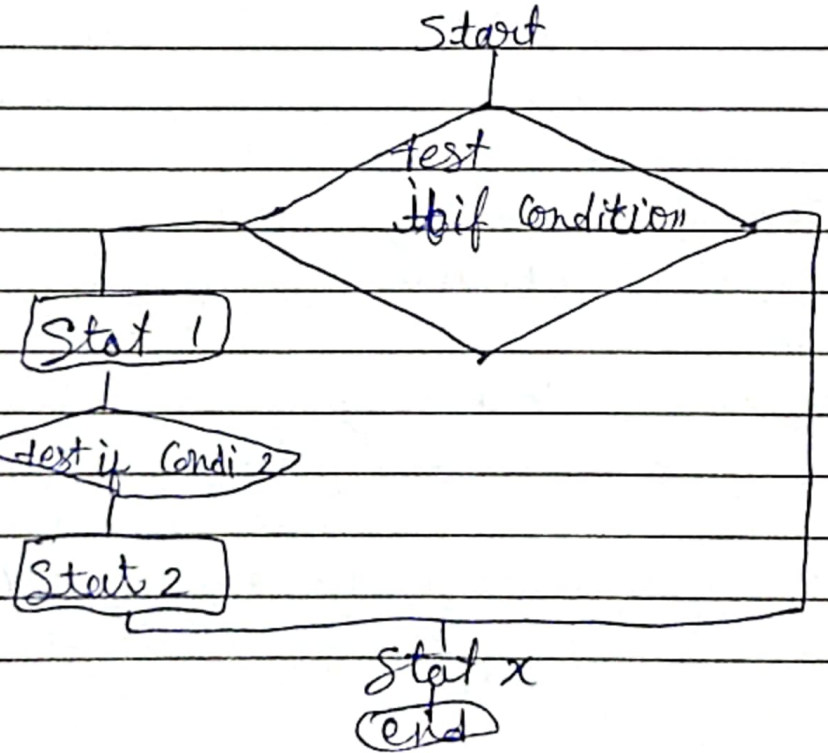
Spectrum
DATE / /
PAGE NO.

2 # Nested if Statement -> Nested if Statement means an if Statement insigd On1 another if Statement. We can placed and if Statement inside another if statement

Syntax

```
-> If (condition)
   { Statement -1; // executed when if -
                   -Condition) true
     if (condition 2)
      {
        Statement -2 // execute when if :-
                     - Conditions is true
      }
   }
   Statement -x;
}
```

Flowchart ⇒



Spectrum
DATE / /
PAGE NO.

Program =

```
# include <stdio.h>
int main ()
{
    int i = 10;
    if (i == 10)
    {
        if (i < 15)
        {
            Printf ("i is small than 15"),
        }
    }
    return 0;
}
```

Output = i is small than 15

3 If else = We can use the else statement with if statement to excute a block of statement when the condition is false

~~flowchart~~

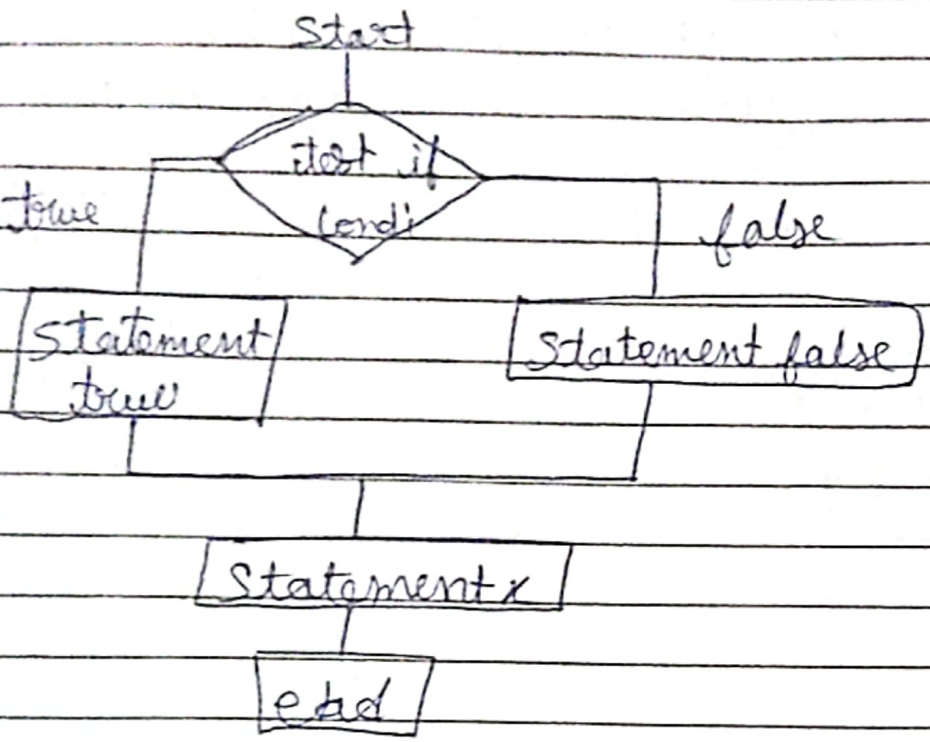
~~Stytme~~

Syntax

```
if (Condition)
{
    statement - true
}
else
{ statement - ~~true~~ false
}
statement n
}
```

Spectrum
DATE / /
PAGE NO.

↳ flowchart



Program

```
#include <stdio.h>
int main()
{ int a = 10;
  int b = 15;
  if (a>b)
    { printf ("a is a greater than b");
    }
  else
  { printf ("b is a greater than a");
  }
    return 0;
}
```

Output - b is a ~~greater~~ greater Than = a

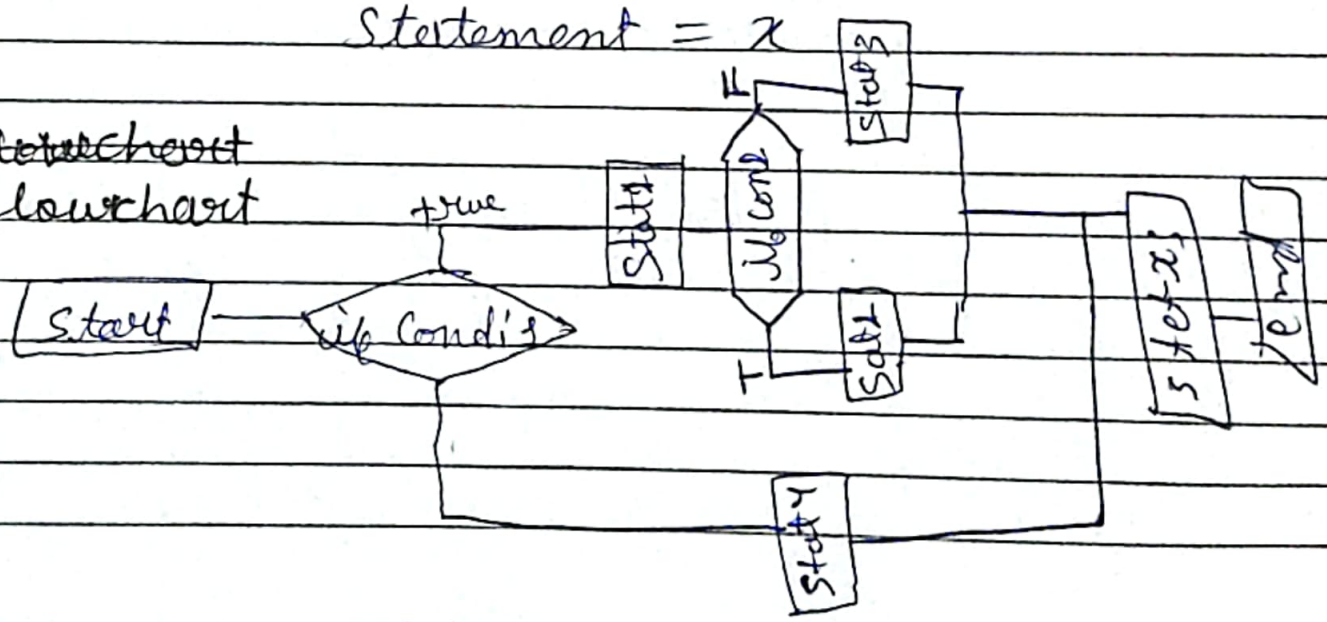
Spectrum
DATE / /
PAGE NO.

c. nested if else ⇒ We use nested if else in the following form.

Syntax ⇒

```
if (condition)
  { stat-1;
    if (cond 2)
      {
        Stat-2;
      }
    else
      {
        Stat-3;
      }
  }
else
  {
    Statement=4;
  }
  Statement = x
```

~~flowchart~~
flowchart



Spectrum
DATE / /
PAGE NO.

Programm:

```
#include <stdio.h>
int main ()
{ int a = 10;
  int b = 15;
  int c = 20;
if (a>b)
   { if (a>c)
      { printf ("a is greater ");
      }
      else
      {
         printf ("c is greater ");
      }
   }
else
   { if (b>c)
      {
         printf ("b is grater ");
      }
      else
      {
         printf ("b is grater ");
      }
   }
```

Output

c is greater

Spectrum
DATE / /
PAGE NO.

if - else

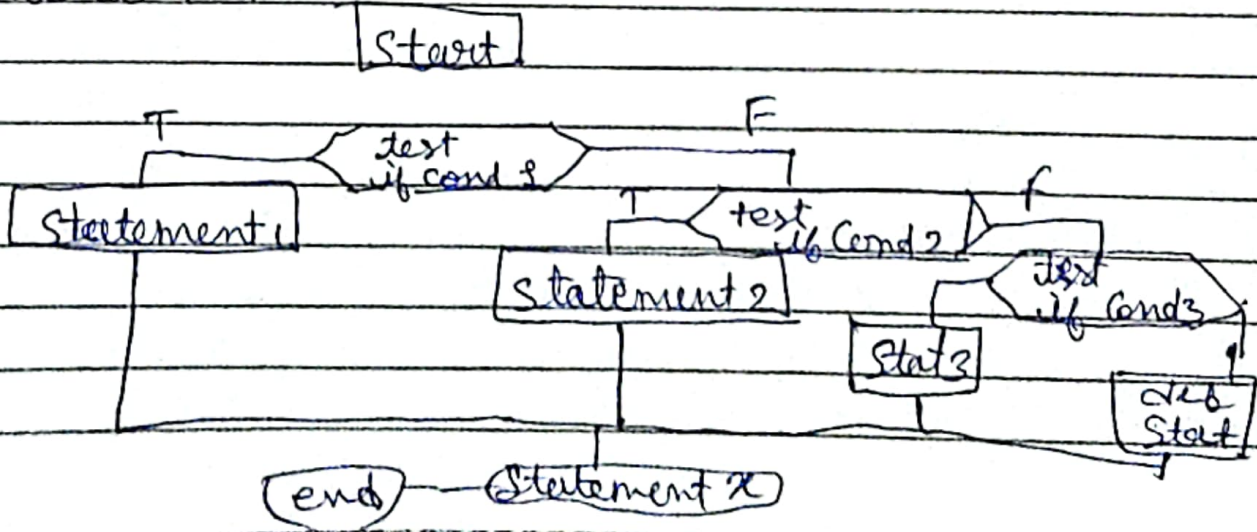
5 else ends if ladder =

Syntax =

```
if (condition)
  {
    Statment - 1
  }
elsif
elseif (condition-2)
  {
    Statement - 2
  }
elseif (Condition - 3)
  {
    Statement - 3
  }

else
  {
    default - Statement
  }
    Statement - x
```

flowchart ->



Spectrum
DATE / /
PAGE NO.

Programm -

Marks	grading
80 to 100	~~hounor~~ honour
60 to 79	Ist division
40 to 59	IInd division
0 to 39	fail

```
# ~~progro~~
# include <Stdio.h>
  int main () {
                int marks = 92;
if (marks<=100 && >=~~80~~ (if (~~marks~~ 100 > marks > 80))
                {
                  printf ("Honour");
                }
else if
elseif (>=60 && <=79) elseif (79 > marks > 60)
                {
                  Printf ("Ist division");
                }
else if (>=50 && <=59) elseif ( 59 > marks > 40)
                {
                  Printf ("IInd division");
                }
else if (>=40 && mars<=49) return 0;
                }
                  printf ("IIIrd division)
else if (>=0 && marks<=39)
    printf ("fail");
else
printf ("invallid number. enter number between 0 to 100
    ); return 0; }
```

Spectrum
Date / /
Page No.

6 Switch Statement ⇒ The ~~Some~~ Switch case Statement is used to execute the Conditional code used based of the value of the variable specified in the switch statement the ~~switch~~ Switch Statement tests the value of the given variable and expression again the case of values and when a match is found a block of statements associated with that case is ~~executed~~ executed

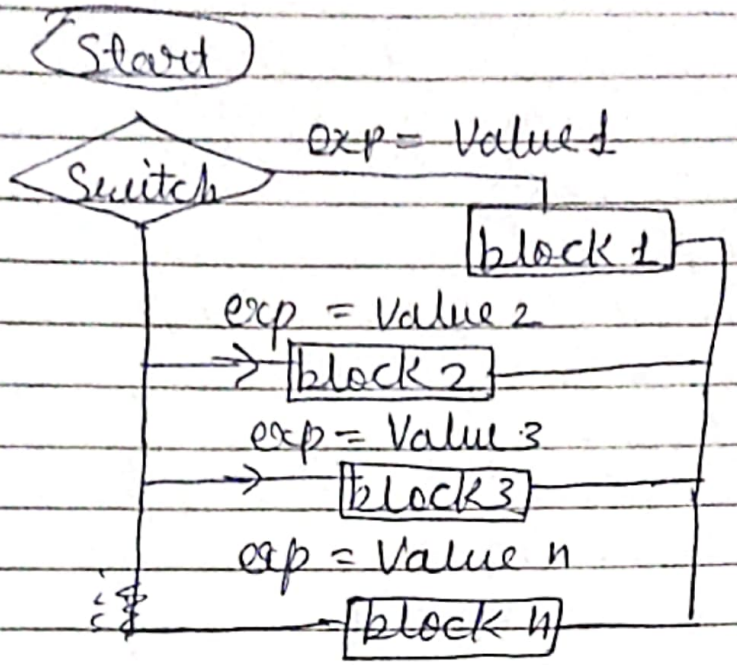
Syntax ⇒

```
Switch (expression)
{
    case Value-1:
            block 1;
            break;
    case Value-2;
            block-2;
            break;
    case Value-3;
            block-3;
            break;
    case Value-n:
            block-n;
            break;
    default :
            default-statement;
}
Statement x
```

Spectrum
Date / /
Page No.

~~Statement~~

flowchart →



Program →

```
# include <stdio.h>
int main() {
        Switch
    int a = 18;
    int vare;
    printf ("enter number 1 to 7");
    Scanf ("%d", & vare);
    Switch (vare)
    {
      case 1:
        printf ("monday");
        break;
      Case 2:
        printf ("Tusday");
        break;
```

```
Case 3:
        printf ("Wednesday");
        break;
case 4:                Thursday
        printf ("friday");
        break;
case 5:
        printf ("friday");
        break;
Case 6:
        printf ("Saterday");
        break;
Case 7:
        printf ("Sunday");
        break;

default:
        printf ("Default Value then 1 to 7");
        break; }
    return 0;
}
```

output

enter number of 1 to 7 = 5 friday

15/10/25

7 Jump Statment Statement = This statment are used in C for unconditional flow of control flow of the throughout the fuctions

Spectrum
DATE / /
PAGE NO.

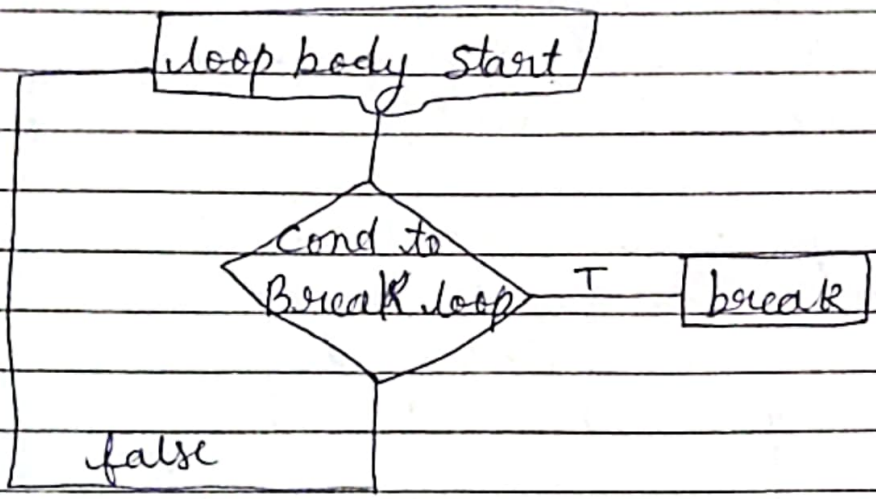
- of the program c supports two types of jump statement

loop

1. break ⇒ This more control statement is used to ~~determine~~ terminet the loop are soon as the break statement is encountes the loop iterations stops thair control Returns from the loop immediately to the first statement after the loop

Syntax ⇒ break;

flowchart =



Exampal-

```
# include <Stdio.h>
int main () {
        int i;
for (i=1; i<10; i++)
    {
    Printf ("%d", i);
     if (i==5)
```

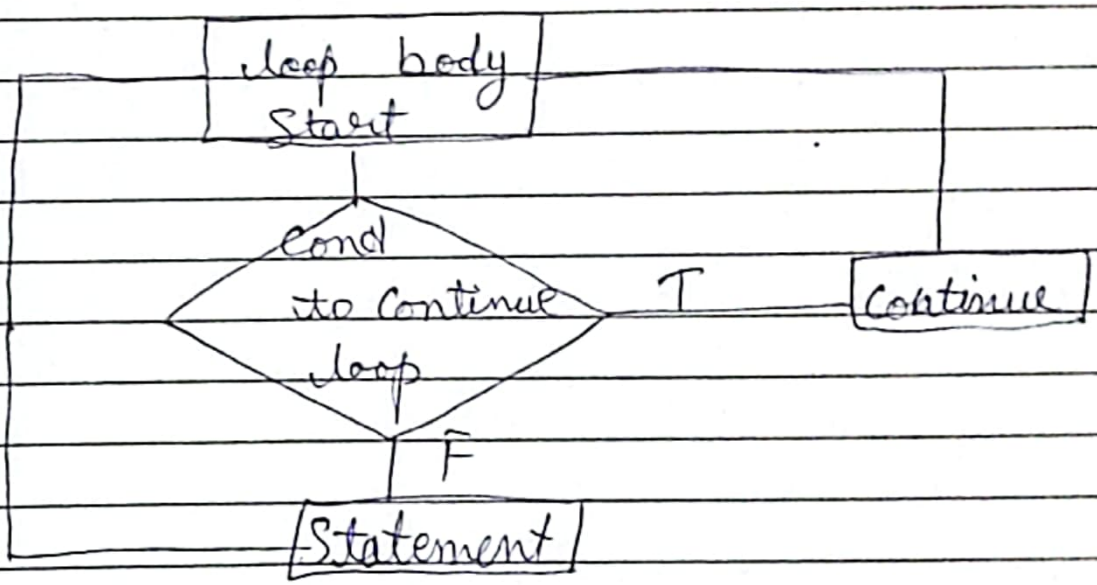
```
    break;
}
```

Output 1 2 3, 4, 5

2. Continue ⇒ The countnue is opposite to break Statement it forces to ecicute ~~t~~ next ~~iteration~~ iteration of the loop When the countinue statement is exicute in the loop, the code inside the loop following the Continue Statement will be skiped and next iteration of the loop will beging begin

Syntax = Continue;

~~Exampal~~ = flowchart =



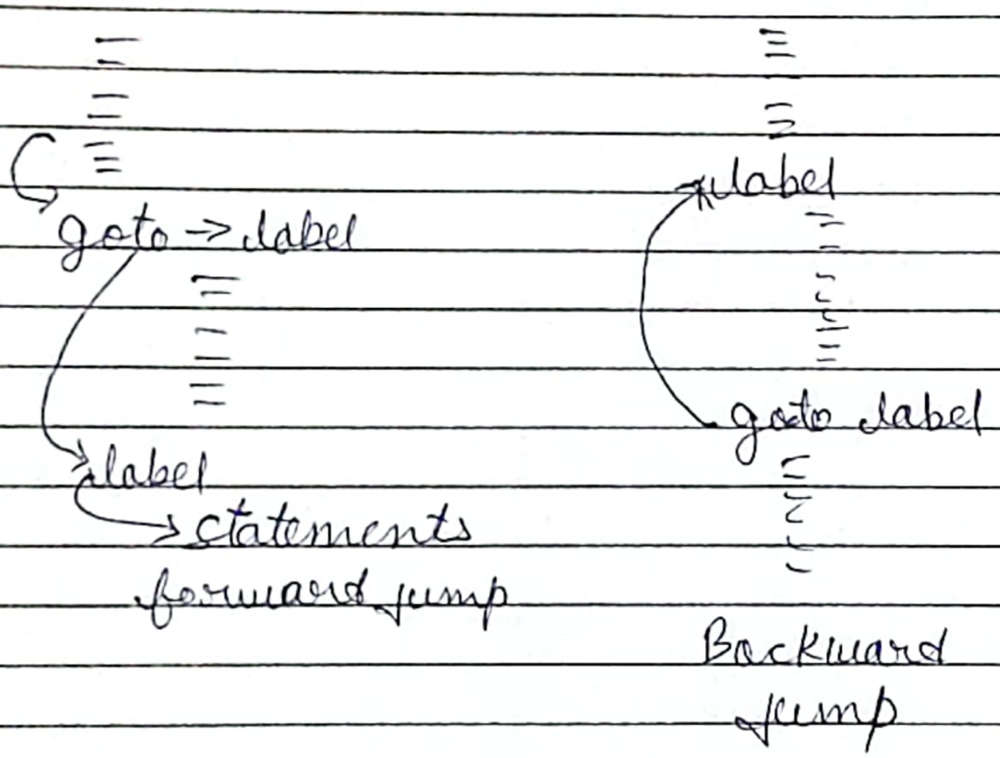
Programm=

```
# include <stdio.h>
int main () {
        int i;
        for (i=1 ; i<10 ; i++)
        {
            if (i==5)
             continue;
            printf ("%d", i);
        }
B}
```

Output = 1 2 3 4 6 7 8 9

3 go to statement => This ~~ac~~ anconditional jump can be used to jump from one point to another using label

Syntex =



Spectrum
DATE / /
PAGE NO.

Programm =

```
#include<stdio.h>
int main(){
        int i=1;
        start:
    if(i<=10)
    {
        printf("%d",i);
        i++;
        goto start;
    }
```

Output

1 2 3 4 5 6 7 8 9 10