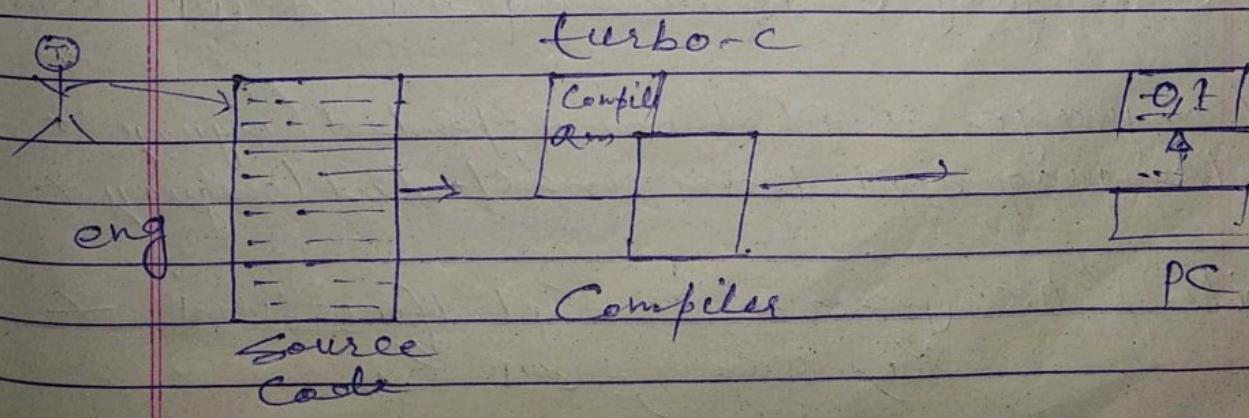
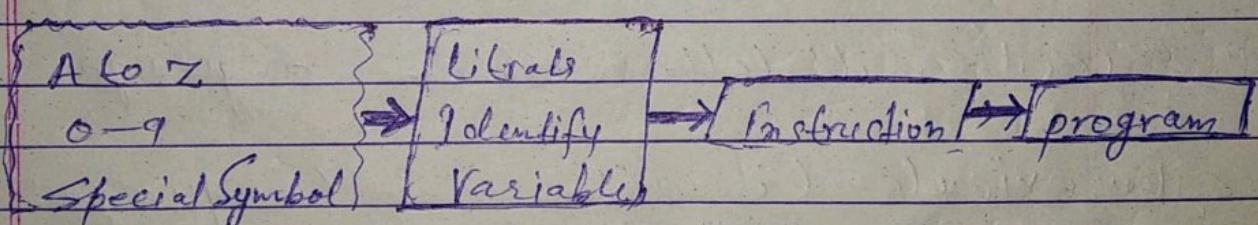


- * Collection of Instructions is called "program".
- * To perform some specific task in Computer is called programming.

	Year	Language	Scientist
(1)	1960	Algol	International Group
(2)	1967	BCPL	Martin Richard
(3)	1970	B	Ken Thompson
(4)	1972	Traditional-C	Dennis Ritchie
(5)	1978	K & R - C	Kernighan & Ritchie
(6)	1989	ANSI - C	ANSI Committee
(7)	1990	ANSI/ISO - C	ISO Committee
(8)	1999	C99	Standardization Committee

* Character Set



Wants to run your own programs (3) *

- Your own source code \rightarrow writing

~ Version \rightarrow Turbo-C *

Turbo-2.0 (i)

Turbo-3.0 (ii)

- Your own will & Compiler *

Compiler (i)

Run (ii)

Program (\rightarrow can check for syntax - Compile) *

if checker Syntax \rightarrow

Run the program in frame (Run) *

- Your self

choose small letter with C-language *

- You execute line by line C-language *

- Your C-Language is Accuracy with you *

C-Language \sim Platform dependent Language *

Java-Language \sim Platform Independent Language *

Compiler & OS you \rightarrow in platform - (i) *

- You Run your own key

According to Syllabus

* C - Character Set:-

(i) Alphabets :- A - Z

(ii) Digits :- 0 - 9

(iii) Special Symbols :- !, \$, #, %, ^, &, *, -, +, /, =, //, :, ;

- History of C - *

C ALERT is C - programming Language

C - language is written in 1972 at Bell Lab. USA

- by Dennis Ritchie

programming language

Reliable

Simple

- C is a general purpose language

(i)

(ii)

in C is program to C - Character Set *

(Basic Items) like (char)

- Given Character Set consists - c (lower case)

, digits , Alphabets are Character Set

- given lower case special symbols

Value (مقدار) NTT (متغير) - Constant (ثابت) *

Constant (ثابت) - Value (مقدار) NTT (متغير)

- (ثابت)

5

- exp

500

-15

Value (مقدار) NTT (متغير) - Variable (متغير) *

- Variable (متغير) -

- Marks (الدرجات) (exp)

Salary (الراتب)

C (C) - Language - Key Words *

- (كلمات مفتاحية) (Key Words)

Computer (جهاز الحاسوب) C (C) - Key Words (كلمات مفتاحية)

- (جهاز الحاسوب) (Key Words)

i) auto

ii) break

iii) case

iv) char

v) const

vi) continue

vii) default

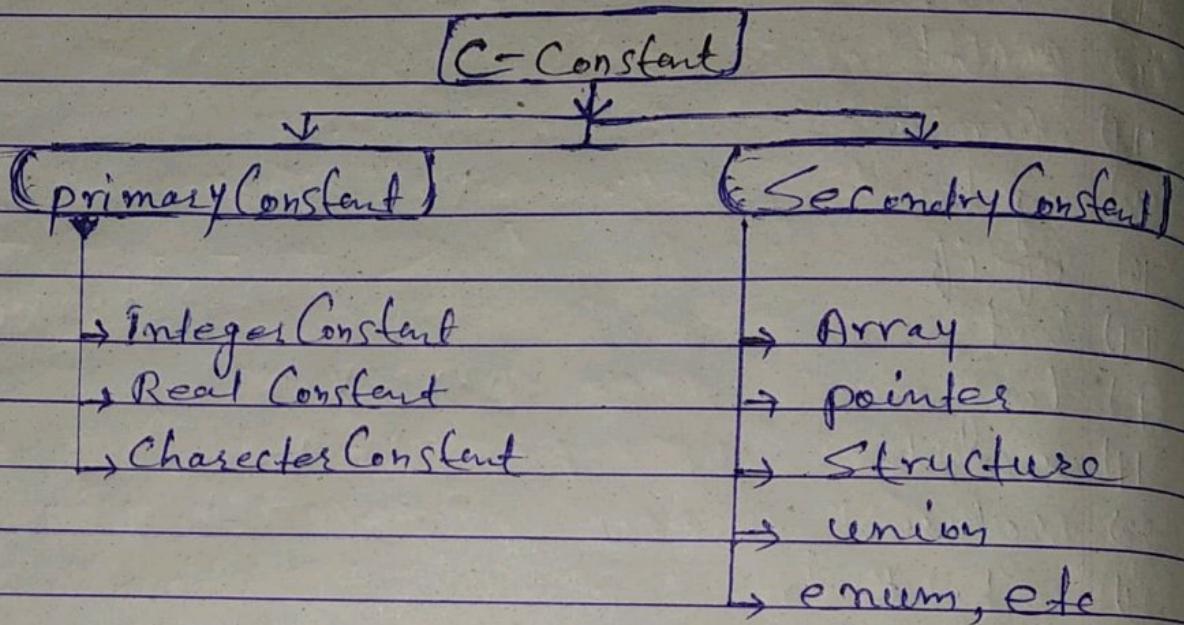
viii) do

ix) double

x) else

- 11) enum
- 12) extern
- 13) float
- 14) for
- 15) goto
- 16) if
- 17) int
- 18) long
- 19) register
- 20) return
- 21) short
- 22) signed
- 23) sizeof
- 24) static
- 25) struct
- 26) switch
- 27) typedef
- 28) union
- 29) unsigned
- 30) void
- 31) volatile
- 32) while

* Type of C-Content



* Integer Constant *

Just like integer Constant
like digit 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
(must have one digit)

ب) لیے، اور Decimal point (سیمیوین) (ii)
(No Decimal points)

iii) either negative or positive

U_n Integer Constant Sign (e.g. 15 or -15) (IV)
for job default (+ sign) is -1500000

(+ default) \rightarrow Integers

Common \rightarrow Integers or Integers Constant (v)

(Range) \rightarrow Integers Constant (vi)

$$-2147483648 \quad -2^{16}$$

$$2147483647 \quad 2^{15}$$

- Upto 2 bytes max Integers Constant *

examples

- (i) 426
- (ii) -8000
- (iii) +782 cfe

* Real Constant *

just like Real Constant *
floating point is also Real Constant *
there
are two types of Real Constants *
one is Fraction form 3.14 (i)
the other is Exponential form $3 \cdot e^{10}$ (ii)

Fraction form 3.14 (i)
Exponential form $3 \cdot e^{10}$ (ii)

must have one digits (i)
by two Decimal point (ii)
positive OR Negative (iii)
+ default sign (iv)
No Space and Commas (v)

~~examples~~

426.12
+ 782.38
- 8.5 etc

* Character Constant *

not ~~digit~~ ~~Character~~ Constant *
Character (i) p w f The Character Constant - An
any character is a Character - c s g i l e
- (i) other non digits (ii) Alphabets Symbols (iii)
- (iv) back ticks and Inverted Commas (v)

'A' (i)
'5' (ii)
'\$' (iii)

Variables

program execution (in C programming)

- (for int Variable for integer, float Values in memory (Computer stores Variable in sub part of memory (Computer (int - character is stored in file) execute Characters (Real, Integer) (for float & double Constant in Line) - for loop - for loop (float & double Variable sub part)

- Under Score (underscore) digits (numbers) Alphabet (letters) Variable (i)
Under Score (underscore) digits (numbers) Alphabet (letters) Variable (i)

Q. Alphabets (Character) i) s (no pt); ii) Variable
- (i) s, Under Score
emp-97 ex

Comments are called as local variable. (iii)
In local variable

5. Under Score (in 0^o) Special Symbols (IV)

editable screen = blue *

Console Screen = black

~~Example~~: valid}

emp - 10
my - int)
- Salary } - Variable

* // Simple programs in C-language

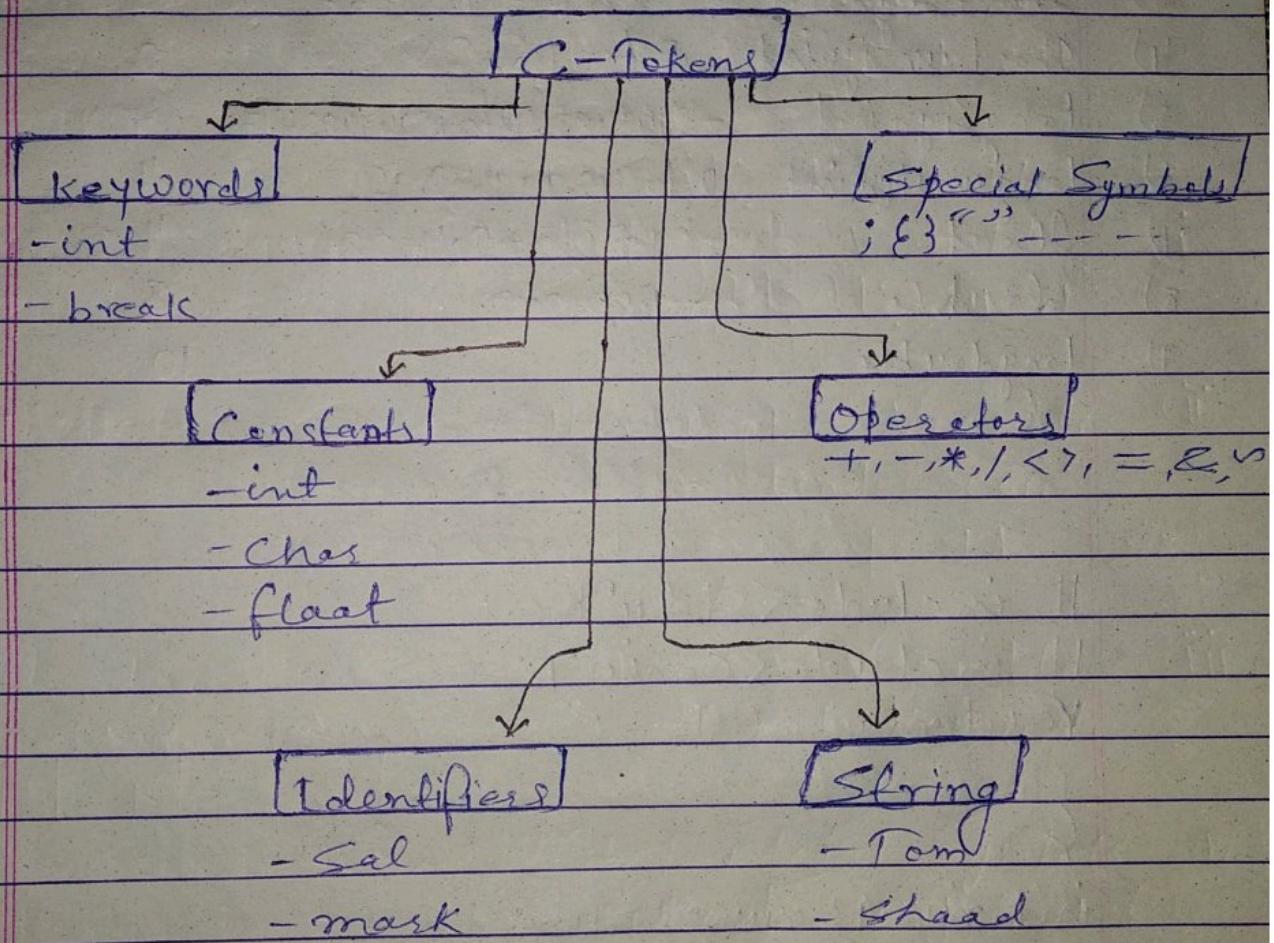
```
#include <stdio.h> // header file
#include <conio.h>
Void main()
{
    printf ("Welcome to manuu");
    // semicolon is use for line termination
    // printf is use to display smes
    getch();
}
```

* Subtraction of Two no in C-language

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int a=10, b=20, Result;
    clrscr();
    printf ("Subtraction of 2 No:");
    Result = a-b;
    printf (result);
    getch();
}
```

* C-Tokens:-

program ہے جس کا (Ur) C programming language ہے
- جس کے C-Tokens ہوں - اسے دوں ہیں اور
Basic building blocks کو جو part of C-Tokens
ہیں لیکن یہ 6 ہی C-Tokens - جس کے عوں
- (Ur) 25



* Data Types:-

- i) Int Data Type
- ii) Char Data Type
- iii) float Data Type

* Data type in programming C

*	Data type	Range	Byte	format
1)	SignChar	-128 to +127	1	%c
2)	UnsignChar	0 to 255	1	%c
3)	Short Sign int	-32768 to +32767	2	%d
4)	Short unsign int	0 to 65535	2	%u
5)	long Sign int	-2,147,483,648 to +2,147,483,647	4	%ld
6)	long unsign int	0 to 4,294,967,295	4	%lu
7)	float	-3.4e38 to +3.4e38	4	%f
8)	double	-1.7e308	8	
9)	long double		10	

* #include < stdio.h >

#include < conio.h >

Void main ()

num1	num2	Sum
2	4	6

{

int num1, num2, Sum;

printf ("Enter two No ");

Scanf ("%d %d", &num1, &num2);

Sum = num1 + num2;

printf ("Sum of two No are = %d", Sum);

getch ();

}

```

#include <stdio.h>
#include <conio.h>
Void main()
{
    int year, period;
    float amount, interest, value;
    printf ("Input amount, Interestrate, and period\n");
    scanf ("%f %f %d", &amount, &interest, &period);
    printf ("\n");
    year = 1;
    while (year <= period)
    {
        Value = amount + interest + amount;
        printf ("%d RS %.2f\n", year, Value);
        amount = Value;
        year = year + 1;
    }
}

```

* Operators *

- 1) Arithmetic operator +, -, *, /,
- 2) Logical operator &&, ||, !
- 3) Relational operator <<, ==, >>, !=, !=
- 4) Assignment operator =, +=, -=, *=, /=, %=
- 5) Increment operator
- 6) Decrement operators ++, --
- 7) Conditional operator ?:
- 8) Bitwise operators &, |, ^, <<, >>, ^
- 9) Special operators < * ,

* Arithmetic Operators *

operator	Description	Example
+	Add two No	$A+B = 30$
-	Subtract 2nd Operand from first	$A-B = -10$
*	Multiply both operand	$A*B = 200$
/	divide numerator by denominator	$B/A = 2$
%	Modulus operator & remainder after an integer division.	$B \% A = 0$

* C program to demonstrate working of Arithmetic operators.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a = 9, b = 4, c;
    c = a + b;
```

```
printf ("a+b = %d\n", c);
```

```
c = a+b;
```

```
printf ("a-b = %d\n", c);
```

```
c = a-b;
```

```
printf ("a*b = %d\n", c);
```

```
c = a/b;
```

```
printf ("a/b = %d\n", c);
```

```
return 0();
```

```
}
```

* Relational Operators *

Operator	Description	Example
$= =$	check if the value of two operands are equal or not if yes then the condition become true.	$(A == B)$ is false
$!=$	check if the value of two operands are not equal then the Condition become true.	$(A != B)$ is true

> check if the value of left operand is greater than the value of right operand if yes then Condition become true. $(A > B)$ is false. $10 > 20$

< check if the value of left operand is less than the value of right operand if yes then Condition become true. $(A < B)$ is true. $10 < 20$

\geq check if the value of left operand is greater than & equal to then the value of right operand. if yes then Condition become true. $(A \geq B)$ is false

\leq check the value of left operand is less than & equal to then the value of right operand. if yes then Condition become true. $(A \leq B)$ is false

* Logical Operators *

Operator

Description

Example

$\&$

Logical & if both the
operands are ~~not~~
non-zero then the
Condition become true.

$(A \& B)$ is
false.

\parallel

Logical OR if any the
two operands is
non-zero then the
Condition become
true.

$(A \parallel B)$ is
true.

!

Logical NOT Reverse
the logical bit

$(\bar{A} \& \bar{B})$ is
true

4 ~~OR, NOT~~ truth Table

A	B	$\& A \wedge B$	OR $A \vee B$	$!(A \vee B)$
0	0	0	0	1
0	1	0	1	0
1	0	0	1	0
1	1	1	1	0

* Assignment Operator *

operator	Description	Example
=	Simple assignment operator	$C = A + B$
+=	add & assignment operator	$C += A \Rightarrow C = C + A;$
-=	Sub & assignment operator	$C -= A \Rightarrow C = C - A;$
*=	mul & assignment operator	$C *= A \Rightarrow C = C * A;$
/=	div & assignment operator	$C /= A \Rightarrow C = C / A;$

* Conditional operator *

? :

operands 112 i.e. ? , a < b or a > b *
 operands 112 i.e. : if a > b execute
 - b or execute

$a = 12, b = 4$

if ($a > b$)? print (true) : print false

1274

1254

* Increment & Decrement operators *

* Increment operator ++,
+1

Type of Increment operator

- (I) pre increment ++a
- (II) post increment a++

* Decrement operator --,
-1

Type of Decrement operator

- (I) pre Decrement - -a X
- (II) post Decrement a - - X

* Bitwise operator *

& and operator

! OR operator

~ NOT operator

% XOR operator

<< Left Shift

>> Right Shift

ANKIT
Page No. 20
Date

* Special Operator *

L Address

* Value at Address

) Saperator operators

~~Date
29/01/19~~

* Conditional / Selective Statements

- (i) Simple if statements
- (ii) if with else statements
- (iii) if else if ladder (Nested if / multiple if)
- (iv) Switch statements

From -> Condition \Rightarrow If Condition *
- If Nested

- If Boolean operator is false or true *

if (Condition) then program goes (out) *
Else If Conditional Statement will execute
- Else If

۱) آئر صرف ایں میرٹ کو جائیں گے۔
کام سے مل کیا جائے۔

لیکچر اے بائی (معنی: blc) اے جو بائی اے
کسی if with else statements کے
لیکچر

iii) اگر کسی program میں if else if laddles ہے تو جانکر رہنا بہبہ کا کیا ہے۔

Program and menu driven program (iv)
Switch Statement (v) - for the list
-> $\{ \text{case} \}$ statement

Switch Statement (لیکن ۱ کو اور جو اسے کہا جاتا ہے) کو switch keyword کے ساتھ لکھا جاتا ہے اور اس کو کوئی کوڈ نہیں۔

Scitech (i)

Case (ii)

break (v.)

Continue (10)

default (v)

* Loops / Iterations *

i) for loop

ii) while loop

iii) do while loop / do loop

لما نريد ان نكرر امر معين في البرنامج
نكتب له حلقة
حيث ان كل حلقة لها 3 اجزاء
أولها هي التحقق من الشرط
ثانيها هي الامر الذي نكرره
ثالثها هي اعادة انجاز الامر

فيما يلي تفاصيل عن حلقة for

تحتاج لـ 3 اجزاء

Checking Condition (i)

Initialization (ii)

Updation of value (iii)

for loop

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

```
#include <conio.h>
```

```
Void main()
```

```
{
```

```
int i;
```

```
clrscr(); //if clear previous op from O.P Screen
```

```
for(i=0; i<=10; i++)
```

```
{}
```

```
printf("Value of i=%d\n", i);
```

```
getch();
```

```
}
```

* i = initialization

i <= Checking Condition

i++ updation

* for(i=0; i<=10; i++)

①

⑤

→(2) true

④

```
printf("Value of i=%d\n", i);
```

Let's say initialize i=0 in for loop *

- Key 1 Check if Condition is true (i)

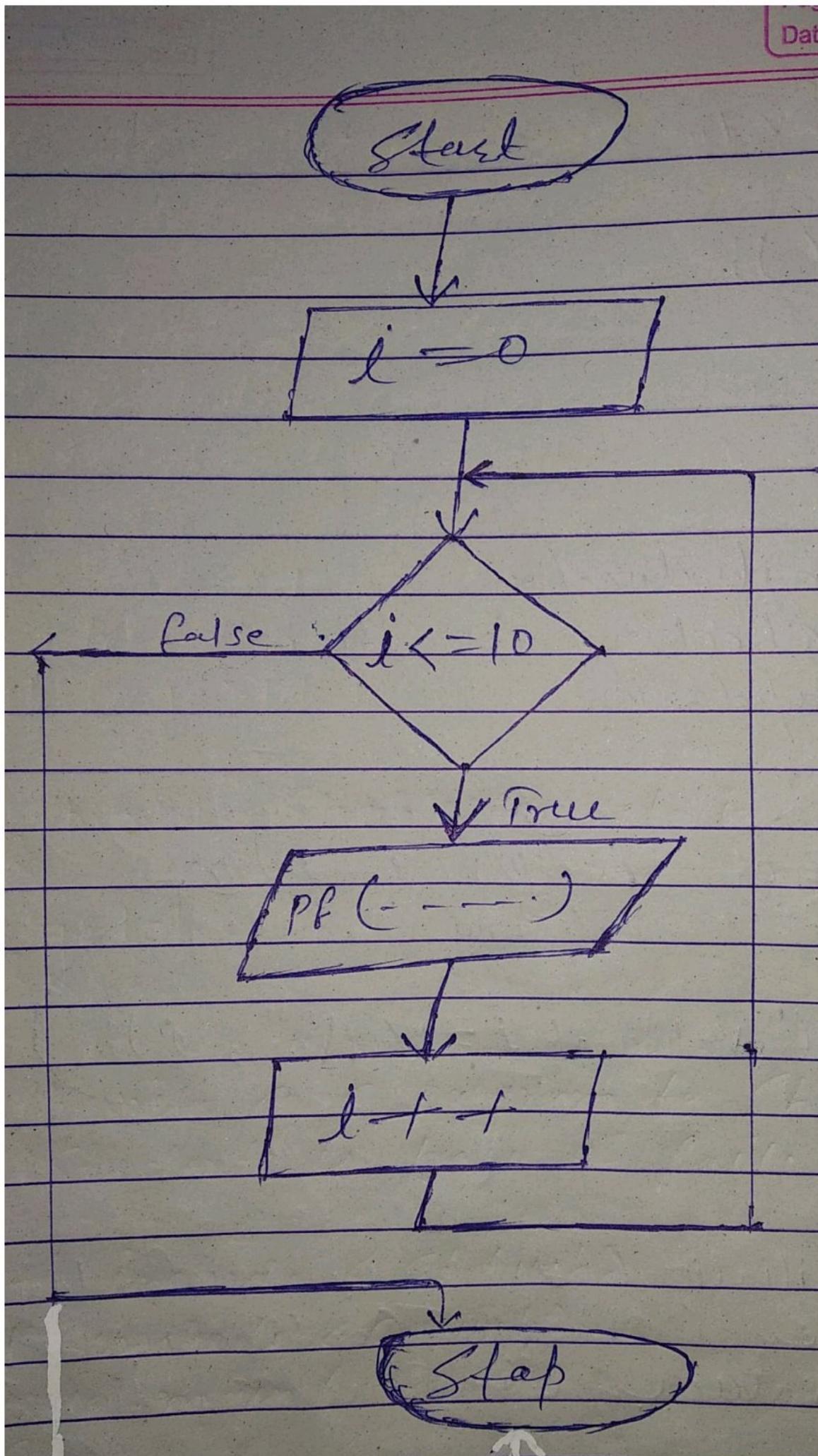
execute value in for True Condition (ii)

- Key 1

- Key 1 update i's value to 1, execute (iii)

- Key 1 Check if Condition is true (iv)

Again - Again



* Simple if
Syntax

if (condition)
{

Statement 1

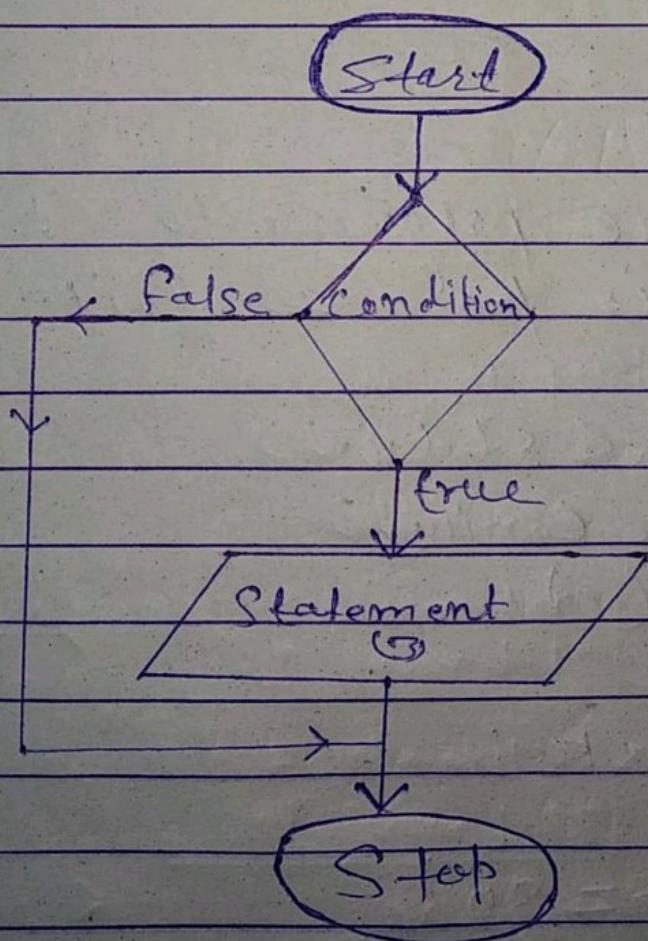
Statement 2,

Statement 3

}

Statement -X

* Flowchart *



*

1) write a c program to show whether candidate is eligible for voting.

```
#include < stdio.h >
#include < conio.h >
Void main()
{
```

```
    int age = 18;
    clrscr();
    If (age > 17)
    {
        printf("Eligible for vote in ");
        getch();
    }
```

x

2) write a c program to show whether student is pass or fail.

```
#include < stdio.h >
#include < conio.h >
Void main()
{
```

```
    int marks;
    clrscr();
    If (mark >= 30)
    {
```

```
3 printf ("Student is pass"); // true  
3 printf ("Student is fail"); // false  
getch();  
3
```

* //Taking input from user

```
#include <stdio.h>  
#include <conio.h>  
Void main()  
{  
    int marks;  
    clrscr();  
    printf ("Please Enter your marks = \n");  
    scanf ("%d", &marks);  
    if (marks >= 30)  
    {  
        printf ("Student is pass"); // true  
    }  
    printf ("Student is fail"); // false  
    getch();  
}
```

2: if else Statement

Syntax

if (Condition)
{

 Statement 1;

 Statement 2;

}

else

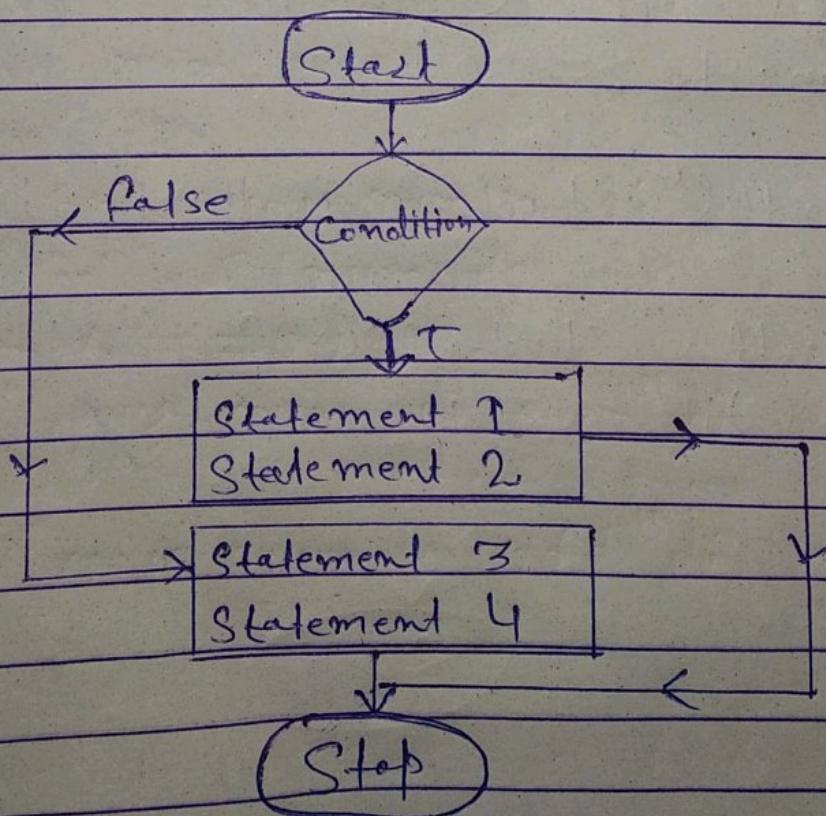
{

 Statement 3;

 Statement 4;

}

Flowchart



* // Write a C program to Show whether Candidate is Eligible for Voting.

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

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    int age;
```

```
    printf("Enter your age");
```

```
    scanf("%d", &age);
```

```
    if (age >= 18)
```

```
        printf("you are eligible for voting"); // true
```

```
    else
```

```
        printf("you are not eligible for voting");
```

```
    getch();
```

```
}
```

* // write a C ~~program~~ program to Show whether Candidate is pass.

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

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    int marks;
```

```
    clrscr();
```

```
    printf("Enter your marks");
```

```
Scarf("odd", &marks);
if (marks >= 40)
    printf(" Student is pass\n");
else
    printf(" Student is fail\n");
getch();
}
```

* // WACP to show wheather check the given number is even Number OR odd Number.

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int num;
    clrscr();
    printf("Enter your num\n");
    Scarf("odd", &num);
    if (num%2 == 0)
        printf(" your num is even");
    else
        printf(" your num is odd");
    getch();
}
```

* Write a c program to show whether check the given number is positive or negative.

~~#include <stdio.h>~~ -3, -2, -1.0, 1, 2, 3, 4, ...

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int num;
    clrscr();
    printf("Enter your num\n");
    scanf("%d", &num);
    if (num <= 0)
        printf("your num is Negative");
    else
        printf("your num is positive");
    getch();
}
```

3. if else if ladder / Nested if statement

1) write a c program to show whether given No is the largest number.

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int a, b, c ;
    clrscr();
    printf ("Enter the value of a,b,c ");
    scanf ("%d %d %d", &a, &b, &c);
    if ((a>b)&&(a>c))
    {
        printf ("a is largest num\n");
    }
    else if ((b>a)&&(b>c))
    {
        printf ("b is largest num\n");
    }
    else
        printf ("c is largest num");
    getch();
}
```

* // WACP to show calculate Student
marks & show result as per following
parameters.

- i) marks 70 to 100 → Distinction
- ii) marks 60 to 69 → 1st class
- iii) marks 50 to 59 → 2nd class
- iv) marks 40 to 49 → 3rd class
- v) marks 30 to 39 → pass
- vi) marks < 30 → fail

include < stdio.h >

include < conio.h >

Void main ()

{

```
int m1, m2, m3, m4, m5, total;  
float per;  
clrscr();  
printf ("Enter marks for 5 sub = \n");  
scanf ("%f %f %f %f %f", &m1, &m2, &m3, &m4, &m5);  
total = m1 + m2 + m3 + m4 + m5;  
per = total * 100 / 500;  
if ((per >= 70) && (per <= 100))  
{  
    printf ("you are pass with distinction\n");  
}  
else if ((per >= 60) && (per <= 69))
```

```

    {
        printf ("you are pass with 1st class\n");
    }
    else if ((per >= 50) && (per <= 59))
    {
        printf ("you are pass with 2nd class\n");
    }
    else if ((per >= 40) && (per <= 49))
    {
        printf ("you are pass with 3rd class\n");
    }
    else if ((per >= 30) && (per <= 29))
    {
        printf ("you are pass\n");
    }
    else if ((per < 30))
    {
        printf ("you are fail");
    }
}
getch();

```

जिस तरीके से हमें Armstrong No. →
 - फिर Armstrong No. का कैसे बनता है? जैसे 153

$$n = 153$$

$$1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153 \text{ Ans}$$

Same is w.r.t views or No limit
- for \leftarrow palindrom No f.w.i.e.

545 ex

Madam

Mom

دیگ دیگ داد

(4) Switch Statement Syntax

Switch (n)

{

Case Val 1:

 // Statement (s);

 break;

Case Val 2:

 // Statement (s);

 break;

Case Val 3:

 // Statement (s);

 break;

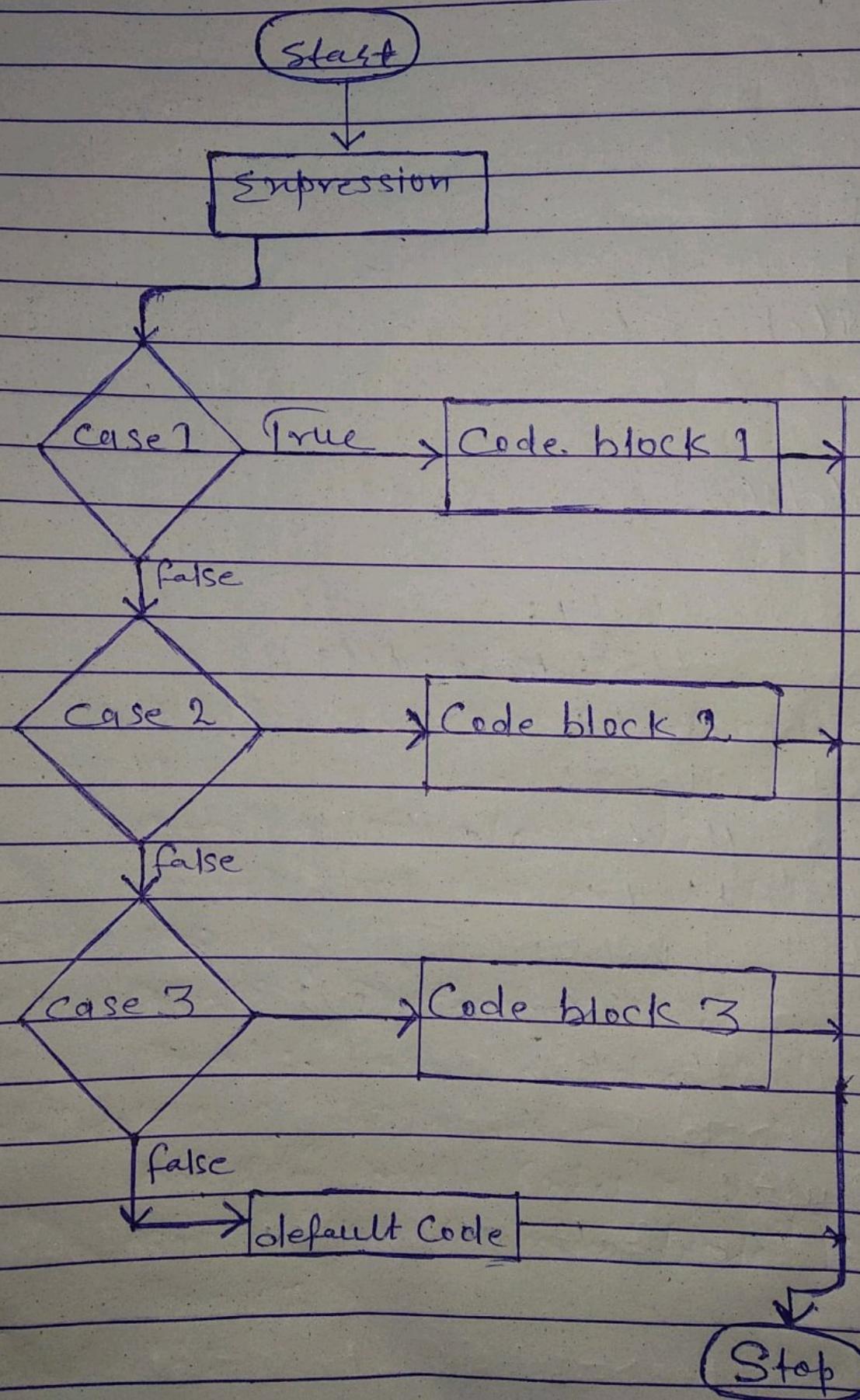
default :

 // default statement;

}

*

Flowchart



* // WACP to show wheather write give month name.

#include <stdio.h>

#include <conio.h>

void main()

{

int rafid;
clrscr();

printf("Enter month number:");

scanf("%d", &rafid);

switch (rafid)

{

Case 1:

printf("January\n");
break;

case 2:

printf("February\n");
break;

case 3:

printf("March\n");
break;

case 4:

printf("April\n");
break;

case 5:

printf("May\n");

```
        break;  
Case 6:  
    printf(" June\n");  
    break;
```

```
Case 7:  
    printf(" July\n");  
    break;
```

```
Case 8:  
    printf(" August\n");  
    break;
```

```
Case 9:  
    printf(" September\n");  
    break;
```

```
Case 10:  
    printf(" October\n");  
    break;
```

```
Case 11:  
    printf(" November\n");  
    break;
```

```
Case 12:  
    printf(" December\n");  
    break;
```

```
default:  
    printf(" Invalid Entry\n");  
    printf(" plz try next time");  
    break;
```

}

3 getch();

Output

Enter your month number : 6
june

Enter month numbers : 14

invalid entry

plz try next time

* // WACP to show calculator

#include < stdio.h >

#include < conio.h >

void main()

{

int k, a, b, c;

clrscr();

printf("Enter your choice");

printf("1. addition\n");

printf("2. sub\n");

printf("3. mul\n");

printf("4. div\n");

scanf("%d", & k);

Switch (k)

{

case 1 :

```
printf("Enter the Value of addition\n");
scanf("%f %f", &a, &b);
```

$c = a + b;$

```
printf("addition two num is %f", c);
break;
```

Case 2 :

```
printf("Enter the Value of Sub\n");
```

```
scanf("%f %f", &a, &b);
```

$c = a - b;$

```
printf("Sub two num is %f", c);
break;
```

Case 3 :

```
printf("Enter the Value of mul\n");
```

```
scanf("%f %f", &a, &b);
```

$c = a * b;$

```
printf("mul two num is %f", c);
break;
```

Case 4 :

```
printf("Enter the Value of div\n");
```

```
scanf("%f %f", &a, &b);
```

$c = b / a;$

```
printf("div two num is %f", c);
break;
```

default :

```
    printf (" Invalid Entry\n");
    printf (" plz try next time\n");
    printf (" thanks");
    break;
}
getch();
```

Loops

// multiplication table using for loop.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, i;
    printf("Enter an integer:");
    scanf("%d", &n);
    for (i=1; i<=10; ++i)
    {
        printf("%d * %d = %d\n", n, i, n*i);
    }
    getch();
}
```

Date 15-02-19

Array

* Array is Linear Collection of Similar Elements

Array is Collection linear is $\{ \text{var}_1, \text{var}_2, \dots, \text{var}_n \}$ (say, i)
in Sub-Script variable for Array - var_i in
Array - $\{ \text{var}_1, \text{var}_2, \dots, \text{var}_n \}$ Variable (Array) - var_i
- i (say, i starting from 0) Indexing (i)

~~execute~~ // WACP to Show average of ten number

```
#include <stdlib.h>
#include <conio.h>
void main()
{
    int a[10], i, sum = 0;
    float avg;
    printf("Enter ten numbers");
    scanf("%d", &a[i]);
    for (i = 0; i <= 9; i++)
        sum = sum + i;
    avg = sum / 10.0;
    printf("Avg is %.f", avg);
    getch();
}
```

* Rule of Array

(ii) int a[];

Size کی توجہ کر کر Deallocate کرے اور error نہیں ہے۔
Size کی توجہ کر کر Initialize کرے اور Compiler نے
کوئی Error نہیں دیا۔

(iii) int a[5];

Deallocate کرے اور Initialize کر کر
Garbage value کو Memory block میں Store کرے۔

(iv) int a[] = {3, 4, 5, 6,};

Initialize کو Value کے Deallocate کرنے کی
لئے Size کی توجہ کرے اور error نہیں ہے۔
Learner note: یہ اسی Reason سے ہے کہ اس کا size 5 ہے اور اس کا limit 100 ہے اور اس کا limit 100 ہے۔

(v) int a[5] = {4, 5, 6, 7, 8, 9, 2};

Value کے Deallocate کرے اور Initialize کو
limit کے لئے error ہے۔ اس کے لئے Initialize کو
کوئی error نہیں ہے۔ اس کے لئے Initialize کو
Two many Initialization ہے۔

(ii) $\text{int } a[5] = \{ 4, 2, 3, \dots, \dots \};$

عند كتابة دالة `main()` في المثلث تكتب دالة `int a[5];` ثم تكتب المحتوى `{ 4, 2, 3, ..., ... };` حيث يتم تعيين القيم المحددة في المثلث، والقيم غير المحددة يتم تعيينها بـ 0.

* Write a C program to generate prime numbers.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int l, u, x, i;
    printf("Enter two numbers");
    scanf("%d %d", &l, &u);
    for(x = l+1; x < u-1; x++)
        for(i = 2; i < n; i++)
            if(n % i == 0)
                break;
            if(i == x)
                printf("%d", x);
getch();
}
```

* //wecp to Show MAANUU on the output Screen
5 times in while Loop.

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int i = 1;
    while (i <= 5)
    {
        printf("MAANUU");
        i++;
    }
    getch();
}
```

* //wecp to Show MAANUU on the output
screen from helps of do while:

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int i = 1;
    do
    {
        printf(" MAANUU");
    }
}
```

i++;

3

while (i<5);

getch();

3

* Looping *

- c. l. b. Group K Ush. Looping
 execute kli (Repeatedly) or fin No of times
 - l. b. Continue (in Repetition) ~ c. l. b.
 false Condition ~ true Condition
 - c. l. b. k terminates Loop i. c. l. b. ev

- P. c. l. b. k. m. o. r. p. l. b. Loop Statement
 for loop (i)
 while loop (ii)
 do while loop (iii)

- P. c. l. b. k. m. o. r. p. l. b. Do while loop i. while loop
 enter do loop w/ Condition or while loop (i)
 do while wh. - c. l. b. w/ Check it. w. i. w. i. w. i.
 check w/ s. c. j. i. execute & loop w/ Condition or
 - c. l. b. w.

- c. l. b. k. m. o. r. p. l. b. pre-tested loop w/ while loop (ii)
 - c. l. b. k. m. o. r. p. l. b. post-tested loop w/ do while loop

Loop w/ l. b. w/ false Condition i. w/ while loop (iii)
 - l. b. w/ execute (or) i. w/ statement
 i. w/ (at least) w/ c. l. b. Loop w/ do while wh.
 - l. b. w/ execute

→ Syntax of while loop

```
int i=0;
while (condition)
{
    Statement;
    i++/i--;
}
```

→ Syntax of do while loop

```
i=0;
do
{
    Statement;
    i++/i--;
}
while (condition);
```

→ Syntax of for loop

```
for (Initialization; Condition; i++/i--)
{ }
```

```
// Statement
}
```

* ARRAY *

Collection of Elements of Similar Type in Array

- c, Structure, Object, Static, int, char, ... - c, k, v, Store of Value & List is called Array (جیزیٹ کرنے والے لیسٹ)

- جیزیٹ کرنے والے لیسٹ

Elements - c, k, v, Equiv to zero (0), Index of Array

- c, k, v, Size (-1)

datatype.

arrayname

[size];

int marks [5];

marks	[0]
marks	[1]
marks	[2]
marks	[3]
marks	[4]

فہریت اے Array

→

کا دلائل اے جیزیٹ کا Group اے Variable of Array

- جیزیٹ

c, k, v, memory of the member is an array

- c, k, v fast access

- جیزیٹ کا Independently Excess of Element

↳ char is a data type. (i) Element size is Array
ii) Size of Array is

data size (j) - char size . Size of Array
- (j) less memory

* integer 2 byte

* float 4 byte

* char. 1 byte.

* User define data type → Structure

* primitive data type →

* Non-primitive data type →

* Struct data today, cl ;
 ↑ ↑ ↑
struct structure variable

- (i) declare (S. →) in body & void main
- (j) Local declaration for col

- (i) declare (S. →) in body & void main
- (j) Global declaration for col

Structure

Variable فی C++ میں - C, Plus وہ طریقہ ہے کہ Structure (i)

- جس کے Grouping ہے

Element، dissimilar ہوں۔ (ii) Similar Structure (iii)

- c کا ہے Collection

- c کا ہے، Uses define datatype (iv) & (v)

Keyword Struct کے لئے create کو Structure (vi)

- جس کے لئے define ہے

New datatype کا ہے کہ define کو Structure (v),

- c کا ہے کہ define کو

Structure Variable کے لئے create کو Structure (vi)

- جس کے لئے create کو

Element کو Structure میں شامل کرنے کے Operators (vii)

excess سے بچنے کے لئے اسے شامل کرنے کا Structure Variable

- جس کے

→ Syntax of Structure:-

Struct Structure name.
{
 datatype element 1 }
 datatype element 2 } member of Structure
 datatype element 3
};

→ example:-

Struct date
{
 int d, m, y;
};

→ example:-

Struct Student
{
 int Roll;
 char name [15];
 float marks;
 int age;
};

~~encircle~~

ANKIT

Page No.

88

Date

* //wapp to show & through structure.

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

```
#include <conio.h>
```

```
struct date
```

```
{
```

```
    int d, m, y;
```

```
};
```

```
Void main ()
```

```
{
```

```
    struct date today, d2;
```

```
    today.d = 29;
```

```
    today.m = 02;
```

```
    today.y = 2019;
```

```
    d2 = today;
```

```
    clrscr();
```

```
    printf(" Enter today's date ");
```

```
    scanf("%d/%d/%d", &today.d, &today.m, &today.y);
```

```
    printf(" Date: %d/%d/%d ", today.d, today.m, today.y);
```

```
    getch();
```

* data type array name [Size];

int a[5] = {2, 4, 5, 9, 10};

float b[5] = {4, 5, 9};

char c[] = {'A', 'B', 'C'};

int	2	4	5	9	7
	a[0]	a[1]	a[2]	a[3]	a[4]

char	A	B	C
	c[0]	c[1]	c[2]

* #include <stdio.h>

#include <conio.h>

Void main()

{

int a[5] = {2, 4, 5, 9, 10};

int i;

for(i=0; i<=5; i++)

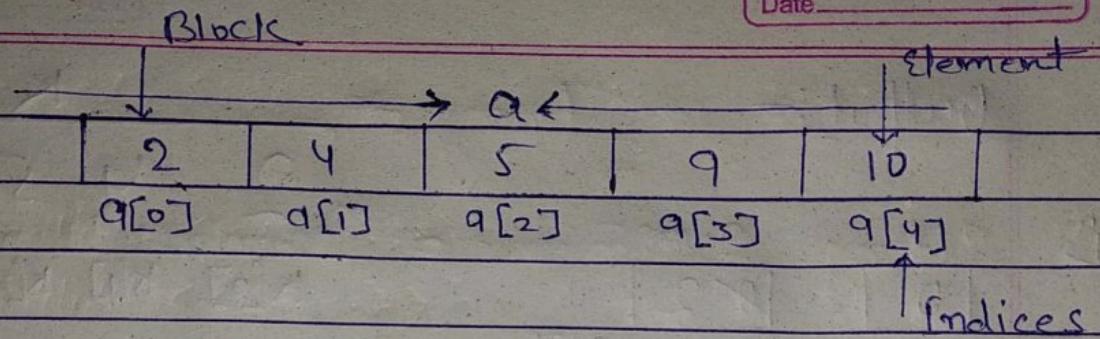
{

printf("%d", a[i]);

}

getch();

}



* // write a c. program to show whether
Sum & average array.

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

```
#include <conio.h>
```

```
Void main ( )
```

```
{
```

```
int k[] = { 2, 4, 5, 9, 10 };
```

```
int i, sum=0, avg;
```

```
for (i=0; i<5; i++)
```

```
{
```

```
sum = sum + k[i];
```

```
avg = sum / i;
```

```
}
```

```
printf (" Sum of array is %d ", sum );
```

```
printf (" avg of array is %d ", avg );
```

```
getch();
```

```
}
```

Output:-

i	j	Sum
0	2	2

K	2	4	5	9	10
K[0]	2	4	5	9	10
K[1]					
K[2]					
K[3]					
K[4]					

$$\begin{aligned} \text{Sum} &= 0 + K[0]; \\ &= 0 + 2; \\ &= 2; \end{aligned}$$

i	j	Sum
1	6	6

$$\begin{aligned} \text{Sum} &= 2 + K[1]; \\ &= 2 + 4; \\ &= 6; \end{aligned}$$

i	j	Sum
4	20	20

$$\begin{aligned} \text{Sum} &= 20 + K[4]; \\ &= 20 + 10; \\ &= 30; \end{aligned}$$

i	j	Sum
2	11	11

$$\begin{aligned} \text{Sum} &= 6 + K[2]; \\ &= 6 + 5; \\ &= 11; \end{aligned}$$

i	j	Sum
3	20	20

$$\begin{aligned} \text{Sum} &= 11 + K[3]; \\ &= 11 + 9; \\ &= 20; \end{aligned}$$

* // WAP to largest no in Array.

#include <stdio.h>

#include <conio.h>

void main()

{ // Largest element of array

int k[5] = {2, 4, 5, 9, 10};

int i, large = 0;

for (i=0; i<5; i++)

{

if (large < k[i])

large = k[i];

}

printf("Largest element of array is %d",
large)

getch();

}

Output

2	4	5	9	10
k[0]	k[1]	k[2]	k[3]	k[4]

i	large
0	0

large < k[0];

0 < k[0];

0 < 2;

= 2 ans.

* String *

String Sname = "Kember Han";

k e m b e r ' H a n \0

Null char ←
End string ←

* String function:

What is definition of String Function?

i) strlen() Find the length of string?

ii) strcmp() Compare two string?

iii) strcpy() Copy one string into another string?

iv) strcat() Join two string?

* Collection of characters is known as String

* Dynamic memory location

String Length

```

* #include <stdio.h>
# include <conio.h>
# include <string.h>
void main ()
{
    s1[20] = "Manuu DB6";
    printf ("Length of string s1: %d", strlen(s1));
    getch();
}

```

```

* #include <stdio.h>
# include <conio.h>
# include <string.h>
void main ()
{
    char s1[20] = "Manuu";
    char s2[20] = "DB6";
    char s3[20] = "Bihar";
    printf ("Length of string s1: %d", strlen (s1));
    printf ("Length of string s2: %d", strlen (s2));
    printf ("Length of string s3: %d", strlen (s3));
    getch();
}

```

Output:- S1=5

S2=3

S3=5

* String Compare *

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
Void main()
{
    char S1[20] = "Beginnce books";
    char S2[20] = "Beginnce book.com";
    if (strcmp(S1,S2) == 0)
    {
        printf("String1 & String2. are equal");
    }
    else
    {
        printf("String1 & String2. are different");
    }
    getch();
}
```

* String Copy *

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char s1[20] = "Hello world";
    char s2[20] = "Beginners book.Com");
    strcpy(s1, s2);
    printf("String s1 is: %s", s1);
    getch();
}
```

- Collection of Same Variable is called Array.
- Collection of different Variable is called Structure.

Function:-

نے کرنے والا program ہے جو اسے
Function ہے کہ کرنے والے کو دھیوں میں اسے
جس سے جس سے

- C, C++, Java, Python, etc. میں اسے جس سے جس سے

Syntax

Return type.	Function name	Input parameters (Argument)
--------------	---------------	--------------------------------

example

i) int add (int a, int b)

ii) float Salary (int days)

iii) char msg();

- (i), (ii), (iii) are all Function →

لیے ہوئے ہوں گے اسے Function, user define - : Ans

predefine Function (i)

user - define Function (ii)

*

```
#include <stdio.h>
#include <conio.h>
main()
{
    msg(); // Function declaration
    printf("CSE Department");
}

msg() // Function definition
{
    printf("Welcome to MANUU");
}
```

*

```
#include <stdio.h>
#include <conio.h>
void main()
{
    printf("In I am in main");
    italy();
    brizil();
    china();
}

italy()
{
    printf("In I am in Italy");
}
```

brazil ()

{ printf("In I am in brizil");

china ()

{ printf("In I am in china");

getch();

Output

I am in main

I am in Italy

I am in brazil

I am in China

* // Sending & Receiving Value between Function.

```
#include <stdio.h>
#include <conio.h>
main()
{
    int a, b, c, sum;
    printf("In Enter any 3 number");
    scanf("%d,%d,%d", &a, &b, &c);
    sum = calsum(a,b,c);           // Actual parameters
    printf("In Sum = %d", sum);
}

calsum (n,y,z)          // Formal parameters
int n, y, z;
{
    int d;
    d = n+y+z;
    return(d);
}
```

Ques 21

Ans 21

Output

Fun()

{

```
char ch;
printf("n Enter any alphabet");
scanf("%c", &ch);
if (ch>=65 && ch<=90)
    return (ch);
else
    return (ch+32);
```

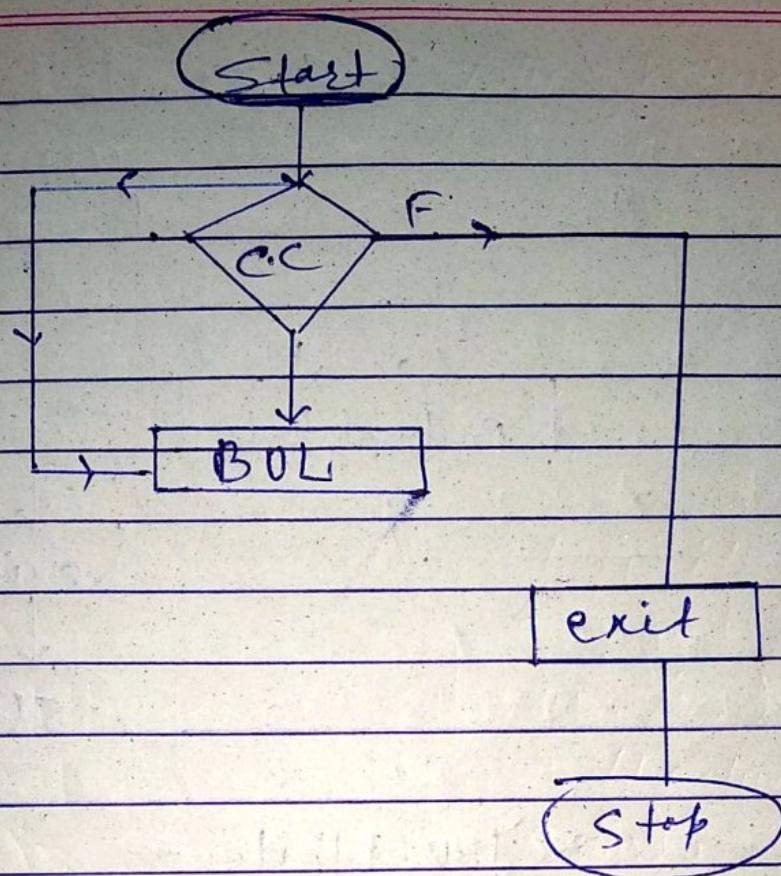
}

Difference between loops

* For loop *

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i=0;
    clrscr();
    for(i=0; i<=10; i++)
    {
        printf("value of i = %d", i);
    }
    getch();
}
```

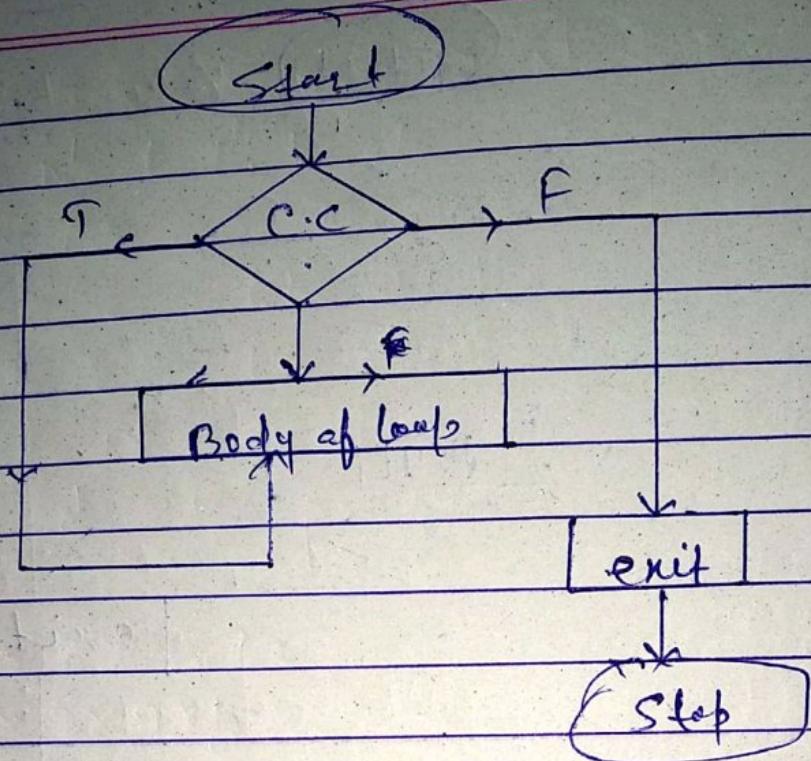
2



* while loop *

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int i = 0;
    clrscr();
    while (i<=10)
    {
        printf("Value of i = %d", i);
        i++;
        getch();
    }
}
  
```

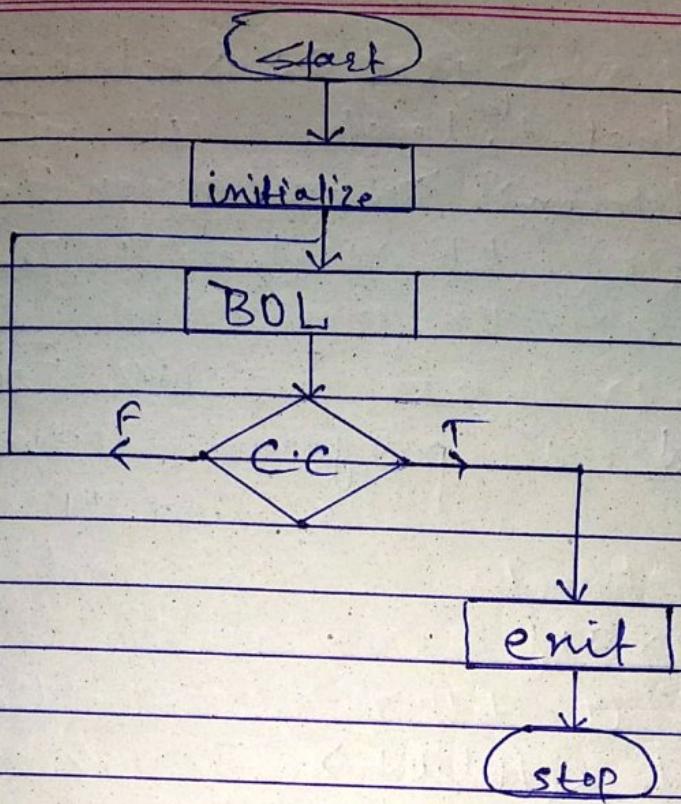


* Do While *

```

#include <stdio.h>
#include <conio.h>
Void main( )
{
    int i=0;
    clrscr();
    do
    {
        printf("Value of i = %d", i);
    }
    while (i<=10)
    i++;
    getch();
}

```



while loop \Rightarrow Normal program execution
 - c. k. w. j. l. m. n. b.

w. j. l. m. n. b. keyword in the while loop
 - c. k. l. s.

do while loop \Rightarrow Menu driven program
 - c. k. w. j. l. m. n. b.

w. use b. keyword \Rightarrow w. do while loop
 - c. k. l. s.

- C W Nested Loop *

- C W Nested loop will c loop
program ans. Recursion is Factorial

- C W New Text

- C W Factorial *

Factorial is program is even no

- C W use %

ex: $5 \times 4 \times 3 \times 2 \times 1$

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, i, f=1;
    clrscr();
    printf("Enter a no");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        f = f * i;
    }
    printf("Factorial = %d\n", f);
    getch();
}
```

- Data type \leftarrow جنس اطلاعات +
Variable \leftarrow اطلاعات علی

Data type (جهاز بواي) - Data type (جهاز بواي) Ans
جهاز بواي - جهاز بواي جمع كرنباي - كرنباي جمع كرنباي - كرنباي Operations (Operations) Operations Operations

- i) primary (Fundamental)
- ii) derived data type
- iii) User define data type

Aptenodytes

الآن، في الدرس السادس من C-language

Character

Real (ii)

Integers (iii)

Dell (f, s) (y)

وون کو عالمی طور پر نظریہ Character type *

کرنے کے لئے ایک جگہ - اسکے مقابلے

We will see how progress is checked in a program

→ ASCII میں کم تر ۱۶ بیٹے کی تعداد ہے۔

نظائر مسلمة هناء - سلمان "R" - سلمان هناء

لهم إني أنت عدو الكفاف، فاجعلني ممدوحاً

(5) Byte "1" in Σ -class beers.

وَهُوَ مُعْلَمٌ بِالْأَفْسَادِ فِي الْمَعْلَمَاتِ - Character Datatype (ii)

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - Unsigned Char (ii)

- جُنْدِيَّةٍ عَلَيْهِ تَكْرِيرٌ

لُوْجِيَّةٍ - جُنْدِيَّةٍ Unsigned Char لُوْجِيَّةٍ

جُنْدِيَّةٍ دَرِيَّةٍ كَوْنِيَّةٍ دَرِيَّةٍ

- جُنْدِيَّةٍ كَوْنِيَّةٍ

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - Signed char (ii)

جُنْدِيَّةٍ دَرِيَّةٍ كَوْنِيَّةٍ

لُوْجِيَّةٍ دَرِيَّةٍ كَوْنِيَّةٍ

- جُنْدِيَّةٍ كَوْنِيَّةٍ

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - String (iii)

جُنْدِيَّةٍ String

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - C Variable Character Data type

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - Variable Data type

خَاصَّةً طَرِيقَةً مُعْلَمٌ بِالْأَرْبَاعِ

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - C Characters Variable Name

جُنْدِيَّةٍ كَوْنِيَّةٍ

- Integer (iv)

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - Integer Datatype

لُوْجِيَّةٍ -32768 وَدَرِيَّةٍ 32767

لُوْجِيَّةٍ دَرِيَّةٍ كَوْنِيَّةٍ 32767

وَهُوَ مُعْلَمٌ بِالْأَرْبَاعِ - memory integer constant

جُنْدِيَّةٍ دَرِيَّةٍ 16 two byte

-32768 سے بڑے اور +32767 fr 4 Bytes (اور ان سے جوچھے تک رسیں معنیت کر سکتے ہیں مثلاً کے طور پر دو افسوس

3 Integer میں کر لانا میں کر لانا

Default type 3 Integer ~ :- Signed Integer (i)
-32768 0 32767 جس میں سے تو
لیکن +32767 کے درمیان کوئی بھی غیر دیا جائے
Bytes 3 میں دو بائیٹ یعنی کام کر سکتے ہیں جیسا کہ بعضاً میں نمبر معنیت سے کام کر سکتے ہیں

ARRAY

- c. Key structure \rightarrow $\{u_1, u_2, \dots, u_n\}$ called Array
- $\forall i \in \{1, 2, \dots, n\} \rightarrow u_i \in \mathbb{R}$ \rightarrow $u_i \in \mathbb{R}, u_1, u_2, \dots, u_n$
- $\forall i \in \{1, 2, \dots, n\} \rightarrow$ u_i has size b \rightarrow memory \rightarrow u_i
- $\forall i \in \{1, 2, \dots, n\} \rightarrow$ u_i has size b \rightarrow u_i is array
- Virtual Homogeneous Data \rightarrow u_1, u_2, \dots, u_n

$\text{float Balance} = 5000;$

$\text{float Balance} = 1000;$

$\text{float Balance} = 500;$

- $\forall i \in \{1, 2, \dots, n\} \rightarrow$ Variable u_i has size b
- $\forall i \in \{1, 2, \dots, n\} \rightarrow$ $u_i \in \mathbb{R}$
- $\forall i \in \{1, 2, \dots, n\} \rightarrow$ u_i has size b
- $\forall i \in \{1, 2, \dots, n\} \rightarrow$ u_i is array
- $\forall i \in \{1, 2, \dots, n\} \rightarrow$ u_i has size b

$\text{float balance}[3] = \{5000; 1000; 500\};$

Balance		
5000	1000	500
Balance[0]	B[1]	B[2]

Fig: Continuous memory location

سے - جسے Continuous Memory Block کہا جاتا ہے اور
جسکی 1st element کو Address والے سے اور
Last Element کو Address والے سے آخر تک
اور یہ سے آخر تک متعارف کرنے کا
معنی ہے۔

کا متعارف کرنا *

کو متعارف کرنے کے Array of C - Language
Value کا جمع ہے اور element کو programmer کے
لئے size کے (جسے) کا ذریعہ ثابت کرنے کا
اس طرح کہ جسے اس طرح - c کے

Data type Arrayname [size];

eg:

int - salary [];

float marks [];

char s.name [];

کو متعارف کرنے کے لئے یہ جائز
ہے - اسکا جامی ہے۔

Data type

کو C - Language
معنی ہے اس کا جامی ہے۔

int (i)
float (f)
char (c)

Array Name.

programmes - c, kew, z, b 15 pl. & Variable n
- c, kew es pl. (3) w (3). m (3)

Size of Element

2. Kew - kew, pl. es 0. Array (0)

Square sum - c, kew, pl. es 0. Array size.
- c, kew, list 0, [] Bracket

How to initialize array

int salary[5] = {5000, 4000, 3000, 2000, 1000};

or

int salary[] = {4, 5, 7};

Salary				
5000	4000	3000	2000	1000
sal[0]	sal[1]	sal[2]	sal[3]	sal[4]

→ Index / Base Index.

write \rightarrow read کے مطابق \rightarrow ای پر جسے array
Index \rightarrow یہ ایک لئے جاتے ہیں اسے index کہا جاتا ہے
Base Index اسی کے برابر ہے جو اسے 0 view
کرنے کے لئے کام کرتا ہے

Accessing Array Element

float Salary [10];

- یہ اسی میں شہیڈ variable کے array
کا حاصل کرنا یہ location کا پر وہ ہے
کہ درجہ کے loop frame کے key
کام کرنے کے لئے یہ lines - یہ

#include <stdio.h>

#include <conio.h>

void main()

{

int salary [5];

int i, j;

clrscr();

for (i=0; i<5; i++)

{ salary[i] = i+100;

}

for (j=0; j<5; j++)

{

```

printf("Element [ %d ] = %d \n", i, salary [i]);
}
getch();
}

```

* Write a C program to Show Two Dimensional Array?

```

#include <stdio.h>
#include <conio.h>
Void Transpose (int, int, int [][ ]);
Void main ()
{
    int i, j;
    int rows, cols;
    int mat[5];
    clrscr();
    printf ("Enter no of row");
    scanf ("%d", &rows);
    printf ("Enter no of cols");
    scanf ("%d", &cols);
    for (i=0; i<rows; i++)
    {
        for (j=0; j<cols; j++)
        {
            scanf ("%d", &mat[i][j]);
        }
    }
}

```

Transpose (rows, cols, mat);

}

void Transpose (int rows, int cols, int mat[][])

{

int i, j;

printf ("The transpose of the given matrix
is:");

for (j=0; j<cols; j++)

{

for (i=0; i<rows; i++)

{

printf ("%d", mat[i][j]);

}

putchar ('\n');

}

→ If we declare array char like String *
→ Then null char '\0'

\0

→ Compiles it -> key is also to Null char
→ It will be string till \0 like,

Char s.name[6] = {'m', 'a', 'n', 'u', 'v', '\0'};

printf ("Welcome to: %s\n", s.name);

(ii) strcpy (s₁, s₂)

سے کیا جس کو s₁ کی s₂ string میں

(iii) strcat (s₁, s₂)

s₁ کے جوڑنے کے آخر سے s₂ کی s₁ string

(iv) strcmp (s₁, s₂)

s₁ اسی سے s₂ کو s₁ کی - عرض 0

(v) strlen (s₁)

کے لئے 0 کی تکمیل کی s₁

~~Date
29/03/19~~

Function

proper activity to do something.

(عملیات کا کامنہ کرنے کا عمل - کامنہ کرنے کی Function)

نام

Syntax / prototype

output type / Return type / datatype Function

(List of argument /
parameter /
Inputs) ;

Sub 2 unit کی مراد سے Function اور Computer *

Program کو کہتے ہیں - کماؤنٹ کے مکمل کوئی ہے - Command / program
کوئی اپنے دل کا اور اپنے حاصل کرنے ہیں -

- کیا حاضر ہے - Function *

(logically) سے یہ Program کو بھی طور پر (:Ans)
محلی و جگہ - جو کہ حاسوب میں بانٹ دیا جاتا -
جسکی وجہ سے Program کو لکھنا اور روشن کرنا اور
- کیا ہے اسے جانتے ہیں

Function prototype

proto = First

type = Model

prototype level for function کو Function () کی

of Function prototype نہیں کہا جاتا

- کیا کرنے کے Compiler میں سمجھتے ہیں

- یہیں دوسرے میں سن بنے جاتے ہیں

output type / Return type ()

Syntax :

Return type Fn (List of parameters);

e.g. int add (int a, int b);

Return type

وہ وہ کوئی بھائی ہے - جو فنکشن کو آؤت ہے
 کوئی طور پر کھا دیا ہے - مثال کے طور پر ایک
 ہے (وہ وہ وہی کو جھے رکھے ہے) - Function
 تو یہ کسی اسکا آؤٹ پر مانی دیکھیں -
 اگر کوئی مانی دیکھیں۔ اگر کوئی مانی دیکھیں
 return void کو جانے کو Compiler نہ دیجائے تو
 کرنے پڑے۔

Function Name.

ایک پروگرام میں کوئی Function کی سلسلہ نہیں -
 اسکے کام کے مطابق ہے
 مثال کے طور پر جمع کرنے کے pre-defined Function پر کوئی add
 کی طرح نہ ہے

Parameters List

کوئی وہ نہیں ہے - جو کوئی طرف پر یا سے
 کرنے ہے - اب جتنے ایسی Functions کی طرف
 ہے کہ دنالے - اسے کوئی parameters کے ساتھ
 کہا جائے گا - اسے کوئی parameters
 parameters کے ساتھ کہا جائے گا - Compiler نے

- a. 5 निम्नलिखित data type का

* Type of parameters *

- i) Input parameter
- ii) output parameter
- iii) Formal parameters

```
#include <stdio.h>
#include <conio.h>
int sum (int, int); // input parameters.
Void main ()
{
    int r;
    clrscr();
    r = sum (5, 10); // output parameters
    // calling function
    printf ("%d", r);
    getch();
}

int sum (int a, int b) // Formal parameters
{
    return (a+b);
}
```

Called Function

✓ Return Statement ✓

Transfer کو والجمن و ڈی Statement
- کو call of Function کو پڑھتے ہیں
اب جس جو وہ دی جائے ساخت
کو C++ اس پرورشام کیں
15 O/P ہے جو اس سے solve
کو return

a ڈی - کے 5,10 parameters کیا ہے اس پرورشام کی
- کیا Formal parameter کو b ہے
- کیا Actual parameters کے 5,10 ہے

* Local & External Variable

- کیا یہی یہی میں مشارفہ میں body کی گئی Variable ہے
- یہی Local Variable میں
جس کو ہے

```
Fun(i,j)
int i,j;
{
    int a,b;
}
```

Answer - کیا Local Variable b ہے اس ڈی
- کیا یہی body کی Function ہے

Function کا صرف Local variable ہے
 جو اسی میں استعمال کر سکتے ہیں اور جو
 اس فنکشن کے باہر معرف کی جائے تو اس
 - جس کو Global variable
 میں کے طور پر

Local Variable &

void main()

{

int area;

area = a * b;

printf("Area = %d", area)

}

getch();

Global Variable

int a, b = 10;
 void Fun();

int main()

{

a = 8;

clrscr();

Fun();

getch();

مثال کسی بھی program کی Global variable
 - جس کو

* Difference between Local & Global variable

اے. یاد کرو کہ Global Variable (i)

whj - جس کو External variable

کہا جاتا ہے اور اس کو Local variable

- جس کو Automatic Variable کہا جاتا ہے

(ii) بور (Escape) نہیں Global Variable
 نہیں - اسے program
 (iii) Function میں کوئی Local Variable
 - اسے (پر) کہتے

(iv) اس main () کی Global Variable
 نہیں - اسے (کل) کہتے
 (v) اس main () کی Local Variable
 - اسے (پر) کہتے

Local Variable

void main ()

{

int area;

area = a * b;

printf("area=%d", area);

}

Global Variable

int a, b = 10;

void main ()

{

a = 8;

clrscr();

fun();

Automatic Variable

ایسا اسکریپٹ کو Function کہا جائے - کوئی

اسکریپٹ کو Function کہا جائے - Function کے Variable

اسکریپٹ کا Storage Class ہے اس کو just

in stack - اسکے Automatic Variable

یعنی Local program کے Variable

~~Secure auto var~~: Automatic Variable - اس کو ایسا دوسرے بروگرام کے لئے (secure)

Access (اصل) ہے اسی کی Variable ہیں اس طریقے سے جسکے لئے کوئی دوسرے بروگرام کو اس کا ملکہ نہیں کر سکتا۔

auto

datatype

Variable list;

Static Variable

- اس کو static declare کرو کر جسکے لئے بھی کوئی نہیں کر سکتے اس کو Garbage ہے اور اس

- اس کو Initialize کرو Null Value ہے اس کو دوبارا

as Static Value

execution کے لئے اس کو Initialize کرو کر جسکے لئے بھی کوئی نہیں کر سکتے اس کو

Constant کہا جائے اس کو دوبارا

- اس طریقے سے جسکے لئے کوئی دوسرے بروگرام کو اس طریقے سے

Static

datatype

Variable list;

Recursion *

این کو Function کہا جاتا ہے اس کے لئے ایک call کو کہا جاتا ہے

- آپ کو call کرنے کے لئے

call کیوں کے fun() کے لئے یہ سچا ہے

کرتا ہے اس کے لئے Recursion کے مکمل بروگرام کے لئے

- key call کیوں کے لئے دوبارہ

Recursion Function کے Direct Function کے

- پر میں قبیلہ

Factorial Function

```
int Factorial (int n)
{
    if (n==0)
        return 1;
    else
        return n * Factorial (n-1);
}
```

~~Date
02-04-19~~

Structure & Union

Value (س، ع، ا) میں وہی Variable (س، ع، ا) کو اپنے int Array (س، ع، ا) کو سمجھ کر سکتا ہے جو اسے دیکھ لے۔ اسے دیکھ لے تو اسے datatype (س، ع، ا) کا یعنی اسے دیکھ لے جو اسے دیکھ کر سمجھ کر سکتا ہے۔ اسے دیکھ کر سمجھ کر سکتا ہے اسے دیکھ کر سمجھ کر سکتا ہے۔

Roll No. ۱۹۱ اسماء نام، عمر ۱۹۱ و عورتی marks address

جیسا کہ ۱۹۱ کا میں اسے دیکھ فرمائیں میں اسے دیکھ فرمائیں

Array (س، ع، ا) میں جو اسے دیکھ سکتا ہے اسے دیکھ سکتا ہے۔

میں صرف ۱۹۱ کو دیکھ سکتا ہے جو اسے دیکھ فرمائیں کہ اسے دیکھ فرمائیں کہ اسے دیکھ فرمائیں

ذیکر کو جمع کرتا ہے تو اسے دیکھ فرمائیں

- جمع کرنا یہ کام عمل ہے جو میں کو age
 - کرنا یہ کام عمل ہے structure ہے
 - کہ میں کو Hetrogenous structure کیا
 - ~~کہ~~

Structure

- کوئی نوع یہ کام کے قابل ہے
 - کام کو Variable ہے اس کے Structure
 - جس کو اس کے members کیا جائے - جس کو structure کہا جائے -
 - اس کے members Field کو element

Syntax

Struct < tagname >

{

 datatype member 1 ;

 datatype member 2 ;

 :

}

Structure members - variable ;

Eg:

```
Struct      Stu_info  
{  
    int Roll No;  
    char Sname [30];  
    float marks;  
}  
St;
```

Stu_info (i) - keyword (ii) structure

- (iii) tag name - اپنے کو structure میں دکھانے کے لئے اپنے structure کو دکھانے والا ملکی نام ہے

marks, Sname, Roll No - اسے دکھانے کے لئے اپنے کو خاص باتوں کا حیال

- جسکے عبارت ہے اسے دکھانے کے لئے اپنے کو خاص باتوں کا حیال

roll, name, marks کو دکھانے کے لئے structure (ii)

float, int, char (iii)

(i) Function pointers, Array (ii) Structure (iii)

- اسے دکھانے کے لئے اپنے کو خاص باتوں کا حیال

Memory (i) کو دکھانے کے لئے structure, Compiles (iii)

- اسے دکھانے کے لئے اپنے کو خاص باتوں کا حیال

- اسے دکھانے کے لئے اپنے کو خاص باتوں کا حیال

Structure Variable

Struct date.

{

int day;

int month;

int year;

}

birthday, marriage-day;

* * How to initialize Structure Variable *

Struct stu

{

int height; } Member
float weight; }

}

S1 = { 18DPC5003DB};

Accessing Structure members

There are 13 operators to access structure members
• operator . (dot operator) and -> (arrow operator)

~~جاءات پلور ۱۶۰ جو سکرچ ۲۰۰ کیسے
وں جسے ۲۰۰ نیکھل کر جو کسے - جو کسے~~

Structure - Variable :: date members (S1);

e.g:

S1.height = 5;

S1.weight = 55.7;

S2.height = 6;

S2.weight = 69.2;

if (today.day == 25 & & today.month == 12)

{
printf(" today is strcuture");

}

Structure Assignment

• OR

Array Structure.

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

```
#include < conio.h >
```

```
main()
```

```
{
```

```
Struct andaleeb
{
    char name [20];
    char pin [18];
    char add [15];
}
```

st;

Struct - andaleeb s1 = { "RAFIQ ANDALEEB",
 "18DPCS063DB",
 "DBS Bihar" };

clrscr();

printf("stu name : %s\n", s1.name);

printf("stu pin : %s\n", s1.pin);

printf("address : %s\n", s1.address);

getch();

}

Nested Structure

Syntax:

Struct date

```
{
    int day;
    int month;
    int year;
}
```

Struct stu

{

```
int pin;  
struct date dob;  
};
```

program

e.g.

```
#include <stdio.h>  
#include <conio.h>  
Void main()  
{  
    Struct student st;  
    St.pin = 10;  
    St.dob.day = 1;  
    printf("%d\n", st.pin);  
    printf("%d\n", st.dob.day);  
}  
getch();  
}
```

Structure & Function

(i) Array or Variable (or) Function \approx b (or)

(ii) Structure \approx b (or) - (or) New ex. Argument

- (or) New ex. b is Argument Function

- (or) New ex. b is Function

(i) Passing Structure element & their address
to function.

(ii) passing Entire Structure.

Call by Reference (or) Function Call. \approx (i)
- v. by

Call by Value (or) Function Call \approx (ii)
- v. by

Self Referencing Structure

Struct node

[

int date;

struct node * nextptr;

]

Structure & Array

Difference between Structure & Array.

(i) - عکس نهاد کا ہے اسے heterogeneous in structure ہے (1)

- کوئی ایسا نہاد کا ہے جو اسے homogeneous in structure کہا جائے سکے

(ii) - جس کا جو عکس کا ہے اسے homogeneous in Array کہا جائے سکے

- کوئی بھی بالفہرستی کا ہے جو اسے homogeneous in array کہا جائے سکے

- (iii) members of structure کو define کریں - (iii)

- (iv) elements of Array کو define کریں - (iv)

- کوئی جگہ b (k.w) - struct کو structure کریں - (v)

پنجمین جگہ کو k.w (k.w) کے لئے اسے array کہا جائے سکے

array کا نام نہیں کیا جائے بلکہ square bracket کا نام دیا جائے سکے

Structure کو define کرنے والے members کو structure کے members کہا جائے سکے

array کو define کرنے والے members کو array کے members کہا جائے سکے

مبحث ۱۰ مفہومیت اور ساختار کی Variable کی structure (v)

- جسے بعذیزی میں
جیسا

مبحث ۱۱ اس معاصر فریڈ ویلے اینی تریب میں جمع صورتی
کے میں اس معاصر فریڈ ویلے اینی تریب میں جمع صورتی

- جسے جنگی کے جمع کرنا ہے heterogeneous structure (vi)

جیسا

- جسے جنگی کے جمع کرنا ہے homogeneous view Array

مبحث ۱۲ جا بنا کی structure میں (vii) میں جیسا

- جسے جا بنا کی view Array میں (viii) میں جیسا

مبحث ۱۳ طرفی pointer variable کے place structure (viii)

- جسے کر سکے

جیسا

- جسے کر سکے place کے array

Argument کے بطور function view structure (ix)

- جسے کے call by value میں جا بنا کی
جیسا

Argument کے بطور function view array

- جسے کے call by reference میں (xi)

dot operator (.) کے Variable صبر کی structure (x)

Period operator (.) - جسے کر Access کرنے کی

- جسے کے

Elements in array Index عناصر کو جو اے جو Array
کی طرف میں Square Bracket [] کے ساتھ - جو ہے

* Period operators *

Period operators کو Dot operator
- جو اسی میں ہے

* Pointers *

ذیلی میں ممکن ہے کوئی خاص جگہ میں Variable کا نام لینا

Address کی لیے پہلے میں 1000 - کریں گے

پہلے کو کوئی دیگر دلیل اور ذیلی میں ہے
pointer کو Address کے item کے میں = ہے

ویسے کو Address - یہ کوئی Variable کو اور
- جسیکہ pointer Variable کو کہا جائے

- کوئی Variable کو یہ کوئی Pointer کہا جائے

کوئی memory address کو

"*" کی طرف بالفہری pointers کے لئے دکھانے کے طور پر یہ کوئی Variable
کوئی کوئی دلیل نہیں

* Uses of Pointers *

- Return کے عوامیوں کے زیراً میں اس فنکشن سے زیراً میں اس فنکشن سے کرنے کے لئے ۱) i

فونکشن کے لئے String کو بطور Array دوسرا فونکشن کے مختلف زیراً میں ii

linked list اور جیسا کہ Complete D.S
وغیرہ Binary tree iii

جائز کارڈ دینے کے لئے جیسا کہ جائز کارڈ دینے کے لئے iv
کرنے کے لئے memory v

فرائیٹ کرنے کے لئے ورنے کے Dynamic memory Allocat vi

Address & ~~c~~ derefrancing

جس - یہ کوئی ضاریUnary operator ہے C-language address کیتی (&). - جسکی وجہ سے اس کو کوئی مذکور نہیں کر سکا۔ اور ($*$) کو کوئی مذکور نہیں کر سکا۔ وہ بیو کو فراہم کرتا ہے۔
 جس کے لیے اس کی وجہ سے اس کو کوئی مذکور نہیں کر سکا۔

51292 [10]

$(\&i)$ i address = 51292

$*(\&i)$ value at address = 10

main()

{

int n=100;

printf("address of n: %d", &n);

printf("Value of n: %d", *x);

(OR)

3 printf("Value of x: %d", *(&n));

* Declaration of pointer *

جس طرح دوسرے کے اسی مثال کرنے کے لئے معرف کرنا ہے
- کو declare کرتا ہے اسی طرح variable
- کو یہ طریقہ کہا جاتا ہے

type-name * pointer-name;

int * i;

int * p1, p2;

* pointer type Declaration *

فیلم کو دوسرے مثال کرنے کے لئے معرف کرنے کے لئے اسی طرح اسی طرح
کو declare کرنے کے لئے اسی طرح اسی طرح اسی طرح اسی طرح
کو declare کرنے کے لئے اسی طرح اسی طرح اسی طرح اسی طرح

Data-type * pointer name;

pointer variable کو اسی طرح بنا کر اسی Compiler ' * ' سے
بُن کر کرنے کے لئے address کو pointer variable کو
whole number کی address کو کہا جاتا ہے - اسی طرح
کو جو اسی طرح اسی طرح اسی طرح اسی طرح

e.g. int *ptr;

pointer variable کو ptr کے لئے اسی Compiler ' * ' کے
لئے اسی address کو اسی طرح integers کے جو اسی طرح
کو جو اسی طرح اسی طرح اسی طرح اسی طرح

میں دوسرے ورنہ - پہلا (x) سے c-variable کیا جائے۔ میرے بھائی کو variable کا معرفت کا مدرسہ دیا جائے۔

eg.

float *f;
char *ch;
int *ptr;

address of 5 characters will be \sim like 6 *ch
1 character value will be stored in memory - 1 byte
- 1 byte store

int *p1, p2; 0.5,

points variable \sim p1, why - c-variable ple \sim p2
- c like d' address \Rightarrow - c

* points Assignment

- j is a pointer & value is pointer \sim
- j has ans. of D + 3 bytes not

int j = 1, *ip;	j = *ip;	j = *(ei)
ip = & j;	j = *ip;	
j = *ip;		
* ip = 0;		

Array & pointers

کا نعلقی باللہ کا نعلقی کا pointes یہی Array ہے C-language
کے جس طرز اور بنائی جائیں گے اس Compiler کے ساتھ مطابق کر دیا گے اسی وجہ سے
خالی (0) address
address کے Variable کو کہا جائے گا اسی وجہ سے f کے form کے pointes
جس طرز پر کام کیا گا پس

variable کے pointes (پیٹر) یہی array کی بھروسہ ہے جس کے
لئے

ptr = n;

ptr = &x[0];

کہ جس طرز f کے form کا جائیں گے اسی وجہ سے C Array کی
- جس طرز

x[4] or *(ptr+4) or *(x+4)

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int i, j, n, a[27], *ptr;
    printf("enter the no of element");
    Scanf("%d", &n);
    printf("Enter the Array element");
    for (i=0; i<n; i++)
        a[i] = i;
    ptr = a;
    for (i=0; i<n; i++)
        printf("%d\n", *(ptr+i));
}
```

Scansf(" %d ", &a[i]);

*ptr = a[0];

for (i=0; i<n; i++)

if (a[i] > *ptr)

*ptr = a[i];

printf(" the biggest element in array %d ", *ptr);

getch();

*pointer & string *

- passing values of function

اے اے parameters / Argument of Function

- اے اے اے اے

call by value (I)

call by reference (II)

call by value

جس کو copy کیا جائے تو اس کا value اس کا variable کہا جائے اس کا argument کہا جائے اس کا address کہا جائے اس کا pointer کہا جائے اس کا memory location کہا جائے اس کا local variable کہا جائے اس کا formal parameter کہا جائے اس کا actual parameter کہا جائے اس کا formal argument کہا جائے اس کا actual argument کہا جائے اس کا formal value کہا جائے اس کا actual value کہا جائے

Argument کو address کہا جائے اس کو C.B.R.M Function کو بھیجا جائے اس کو Function کو call کر دیجئے اس کا value اس کا address اس کا call کر دیجئے

لیے ایک Actual Value جو کسی Variable کا جس عکس میں جسے C.B.R کے ساتھ لے لیا جائے۔

```

#include <stdio.h>
#include <conio.h>
Void main()
{
    int n, y, change(int, int);
    clrscr();
    pf("enter the value of n & y");
    sf(".\d.\d", &n, &y);
    change(n, y);
    pf("In main() x = .\d y = .\d", x, y);
}

change(int a, int b)
{
    int k;
    k = a;
    a = b;
    b = k;
    pf("      ");
}

```

Call by Reference

Address value of function is passed to pointer اسی میں کریں وہی میں اسی میں کو pointer

- جیسا کہ reference کو دوسرا میں value ہے اسی میں کو اسی میں کے بعد value of call by reference

- اور اس طریقے میں اس طریقے میں address reference

of Formal Argument to Actual Argument اس طریقے میں اس طریقے میں

(b) - میں میں -

Argument Value میں صرف یہ ہے reference میں C.B.R
operators (l) میں جس بھائی جاتا ہے Address میں کو اسی میں C.B.R
- میں ظاہر ہے اسی میں درج ذیل سچے مثال ہے۔

eg:

```
#include <stdio.h>
#include <conio.h>
main()
{
    int n, y, chage(int*, int*)
    clrscr();
    pf(" enter the value of n & y: ");
    sf("%d %d", &n, &y);
    chage(&n, &y);
    pf("\n in main() x=%d y=%d", n, y);
    chage(int*, int**)
    {
        int *k;
        *k = *a;
    }
}
```

$* a = * b;$
 $* b = * c;$

$\{ p("In charge() n=1d y=1d", *a, *b);$

Car \rightarrow ایسے دیکھو کہ a Original Value ہے C.B.R
کہ b Original value Argument ہے

Dynamic memory management Function

ایسا جیسے فرمت malloc ہے C-LANG C-language
کہ malloc memory ایسے کہا جائے storage
کہ malloc ایسے کہا جائے ہے میں میں claim کر لے گوں
جس کے malloc D.M.N ہے C-language ہے۔

functions کے malloc & C-language ہے ایسے جیسے
فرمات کرتا ہے

`<stdlib.h>`

The `malloc()` Function (1)

`<alloc.h>`

The `calloc()` Function (2)

The `realloc()` Function (3)

The `free()` Function (4)

ایسا \sim sizeof of size of operators for like \sim new
size of data \sim new - operator UNARY operator

Library \sim new operator \sim new \sim operator \sim new
 \sim new keyword \sim new \sim operator \sim new

Size of (char) is (1)
size of (int) is (2)
size of (float) is (3)

- * اسے کہا جاتا ہے "Dynamical memory" اسے Run time طور پر بھی کہا جاتا ہے "Dynamic memory" اسے اور زبان کی صورت میں "Run time memory" کہا جاتا ہے۔
- (Malloc) کو بروگرام کو دے سکتا ہے اسے خارج کرنے کے لئے اسے memory کو اپنے مسائل کے لئے دے دیا جاتا ہے۔

malloc() function

Continuous بس بروگرام فر کر لائیں اور Functions کو میسا کر سکتے ہیں memory

e.g.

pointer Variable = (cost type) malloc (size);

malloc Function

- ہے ملک Variable کو اس کے pointer Variable کے لئے ←

میں کوئی فونکشن کے طور پر کہا جاتا ہے کہ اس کو malloc Function کو cost type (int, char float) - کے لئے دیا جائے لے کر اس کا value (3)

کو اس کے باوجود اس کے size ←

- 6 by w' output

Starting (لے سوئی) Success Allocat (پر نہیں (سوسنیں) نہیں
and success (لے سوئی) - لے سوئی پر 10b c output address
- 6.0b null is 6 by

$x = (\text{int}^*) \text{malloc}(20)$

(س) 10b (لے سوئی) 20 byte of pointer variable (لے سوئی) int
O/P 6 byte (لے سوئی) starting address کیا ہے اور چھ ہے اور
- 10b لے سوئی

$x = (\text{int}^*) \text{malloc}(10 * \text{size of int})$

$$10 \times 2 = 20$$

↑ int

e.g.

```
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
void main()
{
    int k, *p, j=0;
    clrscr();
    puts("In How many numbers:");
    sf ("%d", &k);
    p = (int *) malloc (k * size of (int));
    while (f = 1)
    {
```

```

Pf (" number .%d ", J+1);
Sp (" .%d ", P+j);
J+1;
}
J = 0;
Pf (" In the number are: ");
while (j != k)
Pf (" .%d ", *(P+j++));
}

```

calloc Function

میں پروگرام کو Run time function میباشد
 اسکے لئے جائزیہ ایجاد کیا جائے۔ اسکے لئے دوبار برابر برابر
 میسر ہے۔ (continious) میں اس کے لئے

pointer variable = (cast type *) calloc(n, size)

~~x = (int *) calloc (5, 10);~~

Array کو خاص کر کے Function & calloc کو جائزیہ کیا جائے۔ اسکے لئے structure یا



Realloc Function

callloc() اور malloc() میں دم فنکشن کو function جو فنکشن سے دی یعنی مسحوری کر کر بڑھ کرنے کے لئے اس عمل کیا جاتا ہے۔ جس کے نتیجے کا طریقہ کہ اس طرح ہے۔

new_ptr = realloc (old_ptr, new_size);

Variable کے C-Language میں New pointer کیا، malloc() کے جس کے pointer or old_ptr اور new size کیا۔ اس فنکشن سے callloc() کے برابر ہے اور کوئی ترتیب نہیں۔

y = (int*) malloc (50);

x = realloc (y, 30);