



1. Introduction Of C in Hindi	4
2. Characteristics Of C	4-5
3. Algorithm In C	5-6
4. What Is Flowchart	6-9
5. Basic Structure Of C	9-11
6. <b>Pre-Procisiv or Directiv</b>	<b>11-12</b>
7. <b>Character set</b>	<b>12-13</b>
1. Latter	12
2. Digits	12
3. Special Characters	13
4. White Space	13
8. <i>Identifiers</i>	<i>13-14</i>
9. Key-Word	14-15
10. Variable	15
11. Data Type	16
12. Operator	16-17
1. Arithmetic Operator	17
2. Relational Operator	17
3. Logical Operator	18
4. Assignment Operator	18

5. Conditional Operator	18
13. Printf() Function	19
14. scanf() Function	19-20
15. Decision Control Statement	20-29
• If Statement	21-22
• If else Statement	22-23
• If else if Statement	23-26
• Switch case	26-29
16. Looping Statement	30-35
• For Loop	31-32
• While Loop	32-33
• do – while Loop	34-35
17. Break Statement	35-36
18. Continue Statement	36-37
19. Array In C	37-41
• One Dimensional Array	39-40
• Two Dimensional Array	40-41
• Multi Dimensional Array	41
21. String In C	42-49
• gets()	43
• puts()	43-44
• Strcat()	44-45
• Strcpy()	45-46
• Strlen()	46
• Strcmp()	47
• Strlwr()	48-49
22. What Is Function	49-54
• Definition Of Function	49
• function Declaration	50
• Function Call	50
• Function Prototyping	51
• Actual Parameter	52
• Formal Parameter	52
• Call By Value	54-55
• Call By Reference	55-56
• Local Variable	52
• Global Variable	53-54
23. Pointer	56-60
• malloc()	59
• calloc ()	60
• realloc()	60

• free()	60
23. Structure In C	60-61
24. Union In C	61
25. File Handling In C	62-68
• What Is File	62
• fopen()	63
• File Opening Modes	63-64
• fprintf()	65
• fscanf();	66
• fputc()	67
• fputs()	68

## **Introduction Of C**

c एक Programming Language है 1972 के आरम्भ में डेनिश रिची ने AT & T Bell Laboratories में C Language Developे की थी C के वास्तविक रूप को प्रारंभ में k & RC कहा जाता था इस नाम में k को Brain Kerningham तथा R डेनिश रिची के लिए प्रयुक्त किया जाता है C Language पास्कल की तरह एक Genral Purpose Language है

1960 के आरंभिक दिनों में अधिकांशतः प्रचलित Computer Language किसी क्षेत्र विशेष के कार्यों को सम्पादित करने के लिए लिखी गई थी जैसे Fortran वैज्ञानिक प्रोग्राम के लिए तथा COBOL व्यापारिक प्रोग्राम के लिए इस लिए किसी एसी Computer Language की आवश्यकता महसूस की जा रही थी जिसमें विभिन्न प्रकार के प्रोग्राम सामान दक्षता से लिखे जा सकें अतः कुछ इसी Language का Development Start हुआ जिन्हें आज Genral Purpose Language कहते हैं जैसे C Paskal Basic आदि इन सभी Language में वैज्ञानिक प्रोग्राम भी उसी दक्षता के साथ लिखे जाते हैं जिस सरलता से व्यापारिक प्रोग्राम तथा Computer Game लिखे जाते हैं इसके इतने विशिष्ट गुणों के कारण c Language बहुत जल्द विश्वभर में प्रसिद्ध हो गई C Programming Language में सबसे ज्यादा प्रसिद्ध Language है

## **Characteristics Of C**

c Language की प्रमुख विशेषताए निम्नलिखित हैं-

- यहाँ क्षमतावान तथा विश्वसनीय है
- यह Middel Laval Programming करने में सक्षम है

- यह काफी सक्षम Language है जो basic Language से कई गुनी तेज गति से कार्य करती है
- यह Programmer को main Memory में अकेले Bit's पर कार्य करने के लिए भी स्वतंत्रता व क्षमता प्रदान करती है
- C लैंग्वेज में मात्र 32 Key-Word होते हैं परन्तु इसमें सेकड़ों अन्य प्रोग्राम भी हैं जिनके सहयोग से जटिलतम प्रोग्राम भी सरलता पूर्वक निष्पादन किये जा सकते हैं
- इसमें प्रोग्राम के अग्रिम विकास की भी अनेक सम्भावना निहित हैं

### *Algorithm In C*

एक कंप्यूटर प्रोग्राम किसी भी एक Programming भाषा में लिखा जाता है जिससे कंप्यूटर सम्बन्धित कार्य कर सके परन्तु इसके लिए उस प्रोग्रामिंग भाषा के व्याकरण व शब्दावली की पूर्ण जानकारी होनी चाहिए इसके विपरीत एल्गोरिथम किसी समस्या के समाधान की लिए लिखे गई क्रमवार पदों की श्रंखला होती है जो किसी भी प्रोग्रामिंग भाषा पर निर्भर नहीं है अतः एक एल्गोरिथम को किसी भी प्रोग्रामिंग भाषा में प्रोग्राम बनाने के लिए use कर सकत्व है जो प्रोग्राम से पहले का step होती है

कंप्यूटर का अपना दिमाग नहीं होता है इसे जो निर्देश दिए जाते हैं यह उन्हीं के अनुसार कार्य करता है अतः यह प्रोग्रामर का दाइत्व होता है की वह सही निर्देश दे इसी के लिए एल्गोरिथम लिखा जाता है ताकि प्रोग्रामर भाषा की जटिलताओं में ना उलझते हुए प्रोग्राम के Logic पर ध्यान केन्द्रित कर सके एल्गोरिथम के पदों का विवरण , लॉजिकल क्रम के चरणबध होना चाहिए प्रोग्रामर द्वारा एल्गोरिथम में प्रोग्राम की Step लिखी जाती है जिसकी सहायता से प्रोग्रामर प्रोग्राम के लॉजिक का सही से प्रयोग करता है और प्रोग्राम में गलती होने की संभवना कम रही है

एल्गोरिथम की प्रमुख विशेषता निम्नलिखित है

1. एक एल्गोरिथम में पदों जो निश्चित संख्या होनी चाहिए तथा पद सरल , स्पष्ट व् संक्षिप्त होने चाहियें
2. एल्गोरिथम के कथन किसी भी प्रोग्रामिंग भाषा से अप्रभावित होते हैं
3. कथन स्पष्ट संक्षिप्त का समझने में आसान होने चाहिए
4. अंत में सही आउटपुट मिलना चाहिए
5. एक निश्चित समय में आउटपुट से कर एल्गोरिथम समाप्त होना चाहिए

Ex :

1 से 50 तक की सम संख्या को प्रिंट करने के लिए  
एल्गोरिथम

step 1: INITIALIZATION :N<-0

step 2:n<-n+2

step 3:PRINT N

step 4:IF N<50 THEN GOTO STEP 2

step 5:STOP

## **What Is Flowchart**

कंप्यूटर द्वारा किसी समस्या के हल के लिए लिखे जाने वाले प्रोग्राम के लॉजिक का design करने के लिय दो महत्वपूर्ण tool होते हैं- एल्गोरिथम तथा फ्लोचार्ट कंप्यूटर द्वारा किसी समस्या का हल विभिन्न चरणों में किया जाता है प्रथम चरण में फ्लोचार्ट बनाया जाता है किसी समस्या को हल करने के लिए अपनाये जाने वाली प्रक्रिया का रेखाचित्रीय रूप फ्लोचार्ट कहलाता है फ्लोचार्ट प्रोग्राम के लॉजिक को represent करता है फ्लोचार्ट में किसी समस्या के समाधान में प्रयुक्त प्रक्रियाओं, गणनाओ, निर्णय के बिंदु

आदि का प्रवाह प्रदर्शित किया जाता है किसी समस्या के हल का फ्लोचार्ट उस समस्या के लिए प्रोग्राम लिखने के लिए बहुत सहायक होता है अर्थात् एल्गोरिथम को चित्र के रूप में प्रदर्शित करना फ्लोचार्ट कहलाता है





फ्लोचार्ट की विशेषता

फ्लोचार्ट की विशेषता निम्नलिखित है



- फ्लोचार्ट कार्य करने के विभिन्न चरण का क्रम बनाता है
- फ्लोचार्ट प्रोग्राम तर्क का चित्रित वर्णन है
- यह प्रोग्राम के विभिन्न हिस्सों के बीच सम्बंध व लॉजिक प्रवाह प्रदर्शित करता है
- प्रोग्राम के विभिन्न चरणों को चित्रित रूप से रेखांकित करता है अतः समझने में आसान होता है
- फ्लोचार्ट प्रोग्राम की गलतियों को सुधारने व बदलने में सहायता करता है
- यह किसी भाषा विशेष के आधार पर निर्भर नहीं होगा
- यह प्रोग्राम प्रलेखन का हिस्सा है

फ्लोचार्ट के संकेत-

एक सामान्य फ्लोचार्ट बनाने के लिए विभिन्न संकेत का प्रयोग किया जाता है हर संकेत का अलग कार्य होता है तथा वे अन्य संदर्भ में प्रयोग नहीं किये जाते हैं कुछ महत्वपूर्ण संकेत निम्नलिखित हैं

Operation	Description	Symbol
<b>Start End</b>	इस संकेत को टर्मिनल कहते हैं यह प्रोग्राम के लॉजिक का प्रारम्भ व अंत को दर्शाता है	
<b>Processing Box</b>	यह एक आयातकार बॉक्स होता है इसके द्वारा प्रोग्राम के लॉजिक में विभिन्न अंकगणितीय आउट लॉजिकल एक्सप्रेशन क्रियाओं को व्यक्त किया जाता है	
<b>Input Output Box</b>	इस संकेत द्वारा इनपुट \ आउटपुट क्रियाओं को व्यक्त किया जाता है	
<b>Decision Box</b>	यदि किसी क्रिया में इसे प्रश्नों के उत्तर देने हो जो "हां" या "नहीं" से सम्बन्धित हो तो ऐसे निर्णय Decision Box द्वारा प्रदर्शित किये जाते हैं	



Subroutine	निम्नलिखित संकेत सबरूटीन को दिखने के लिए प्रयुक्त किये जाते हैं कई बार प्रोग्राम के किसी भाग का कार्य प्रोग्राम में कई बार दोहराया जाता है अतः इसे भाग को अलग मोड्युल में लिखा लेते हैं जिसे सबरूटीन कहा जाता है सबरूटीन को प्रोग्राम में कई बार प्रयुक्त कर सकते हैं	
Flow Line	इसका प्रयोग प्रोग्राम में लॉजिक का प्रवाह दिखने के लिए किया जाता है ये तीर के रूप में सरल रेखा होती है	

## Basic Structure Of c Program

c प्रोग्राम का structure निम्नलिखित है

एक c प्रोग्राम कितनी भी संख्या में Directive , Declarations ,definition ,statement, का संग्रह होता है

एक डायरेक्टिव c प्रिप्रोसिसर को कम्पाइल होने के पहले प्रोग्राम के टेक्स्ट पर विशेष कार्य करने का निर्देश देता है

एक डिक्लेरेशन variable या function के एट्रिब्यूट और नाम के बिच संगुणन स्थपित करता है c में किसी भी variable का use करने के पहले डिक्लेअर करना जरूरी होता है

**c प्रोग्राम का basic structure** निम्नलिखित है

```
Documentation Block
Line Block
Definition Block
Global declaration Block
Main () Function Block
{
    Declaration
    Executable Statement
}
User Define function Block
Function 1
Function 2
Function 3
.....
.....
.....
Function n
```

Documentation ब्लॉक में यूजर प्रोग्राम का विवरण उसका उद्देश्य , प्रोग्राम का नाम , लेखक का नाम आदि लिख सकते हैं यही बाद में आवश्यकता होने पर यह डॉक्यूमेंट किसी को भी संशोधन करने की अनुमति प्रदान करता है link box सिस्टम लाइब्रेरी से function से link करने के लिए कम्पाइलर को निर्देश देता है

कुछ variable की आवश्यकता विभिन्न function में होती है तथा इसे variable को global variable कहते हैं इसे variable global Declaration ब्लॉक में डिक्लेअर किये जाते हैं c प्रोग्राम में एक main function ब्लॉक होता है इस ब्लॉक के अंतर्गत मुख्यतः दो ब्लॉक होते हैं जिनके नाम डिक्लरेशन और executable statement हैं पहले भाग में वे सभी variable होते हैं जो main function block में use किये जाते हैं कम से कम एक statement दूसरे भाग में होना चाहिए जो executable statement ब्लॉक होता है दोनों

भाग { } कोष्ठक द्वारा बंद किये जाते हैं main () function ब्लॉक के बाद में यूजर की आवश्यकता अनुसार कई सब प्रोग्राम होते हैं जिन्हें यूजर द्वारा परिभाषित function कहते हैं

उपरोक्त structure को समझने के लिए के संख्याओं का योग ज्ञात करने का प्रोग्राम देखते हैं

```
#include<stdio.h>      \* compiler directive*\nmain()\n{\n\n\* indicates beginning of a block *\nint sum ;      \* deceleration of integer variable*\nint i=10,j=20;      \*deceleration and definition of a variables i,j *\nsum=i+j;      \* executable statement*\nprintf("Sum %d",sum);\n\*library function printf called by main to print the contents of the brackets*\n}      \*indicate the end of block*\n
```

प्रोग्राम में कमेंट /\* एव \*/ अन्दर बंद है

printf एक function है वो variable में स्टोर value को screen पर display करने का कार्य करता है

## Pre-Procisic or Directiv

प्रीप्रोसिसर c Language का एक खास गुण है जो इसे अन्य उच्चा स्तरीय भाषा से विशिष्ट बनाता है c भाषा में प्रीप्रोसिसर का प्रयोग यूजर अपने प्रोग्राम को अधिक सरल बनाने तथा बेहतर बनाने में मदद करता है इसके आलावा यह प्रोग्राम को संशोधित करने में मदद करता है

प्रीप्रोसिसर वह प्रोग्राम है जो सोर्स कोड को कम्पाइलर से पहले पढता है वहा प्रोग्राम कमांड लाइन निर्देशों के अंतर्गत लिखे जाते है ये सही निर्देश # चिन्ह से शुरू होते है इसके अंत में सेमीकॉलन नहीं लगाया जाता है प्रेप्रोसिसर डाइरेक्टिव किसी भी सोर्स प्रोग्राम में main () से पहले लिखा जाता है सामान्य रूप से प्रचलित प्रीप्रोसिसर को निम्नलिखित भागो में विभाजित किये जा सकते है

1. Macro Substitution directives
2. File inclusion Directives
3. Compiler control Directive

### Character set in c

कंप्यूटर की अन्य भाषाओ के समान c language की भी अपनी शब्दावली व व्याकरण है इस भाषा में करैक्टर सेट को मुख्य चार भागो में विभाजित किया गया है

- Letter
- Digits
- special characters
- White space

#### 1. Letter:

इस भाग में अग्रेजी के सभी अक्षर आते है जो इस प्रकार है A to Z  
और a to z

#### 2.Digits

c language 0 से लेकर 9 तक सभी digits का उपयोग किया जा सकता है उदाहरन के लिये

0,1,2,3,4,5,6,7,8,9

### 3. special characters

C language में सभी special character का उपयोग किया जाता है इसे हर एक special characters का अपना एक अलग अर्थ होता है C language में उपयोग किये जाने वाले special character निम्न लिखित है

Example : @ # \$ % ^ & \* ( ) : ; " Etc

### 4. White Space

White space characters set निम्नलिखित है

- Blank space \v
- Horizontal Tab \t
- carriage return \r
- new line \n

### *Identifiers :*

पहचान चिन्ह या Identifiers वेरिएबल समूह के अंतर्गत आते हैं इनका प्रयोग वेरिएबल ,Constant, function , Array आदि की पहचान के लिए किया जाता है जिनका use यूजर अपनी आवश्यकता के अनुसार करता है

Identifiers का नाम कुछ सामान्य नियमों के अंतर्गत निम्नलिखित लिखित प्रकार से रखा जा सकता है –

- Identifiers का नाम किसी latter से शुरू होना चाहिये
- key-word को Identifier नाम की तरह प्रयोग नहीं कर सकते हैं
- विशेष चिन्हों में Underscore का use कर सकते हैं
- English के सभी latter c language में use किये जा सकते हैं  
लेकिन c एक case Sensitive language है इस लिए c language में  
Capital और Small latter अलग – अलग मने जाएंगे

पहचान चिन्ह या Identifiers वेरिएबल समूह के अंतर्गत आते हैं इनका प्रयोग वेरिएबल ,Constant, function , Array आदि की पहचान के लिए किया जाता है जिनका use यूजर अपनी आवश्यकता के अनुसार करता है

**Identifiers का नाम कुछ सामान्य नियमों के अंतर्गत निम्नलिखित लिखित प्रकार से रखा जा सकता है –**

- Identifiers का नाम किसी latter से शुरू होना चाहिये
- key-word को Identifier नाम की तरह प्रयोग नहीं कर सकते हैं
- विशेष चिन्हों में Underscore का use कर सकते हैं
- English के सभी latter c language में use किये जा सकते हैं  
लेकिन c एक case Sensitive language है इस लिए c language में  
Capital और Small latter अलग – अलग मने जाएंगे

### Key-word in c

c language में सभी शब्द या तो कोई key –word होता है या तो Identifier key word का एक निश्चित अर्थ होता है जिसे यूजर द्वारा बदला नहीं जा सकता है इन्हें पूर्व निर्धारित Syntax के आलावा अन्य अर्थों में प्रयोग नहीं किया जा सकता है इन्हें अंग्रेजी के small Letter में लिखा जाता है

key – word को reserve word भी कहते हैं कुछ c compiler अन्य key word को भी सपोर्ट करते हैं c language में 32 key word होते हैं

c language के सभी key –word निम्नलिखित

हैं

---

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

### Variable In C

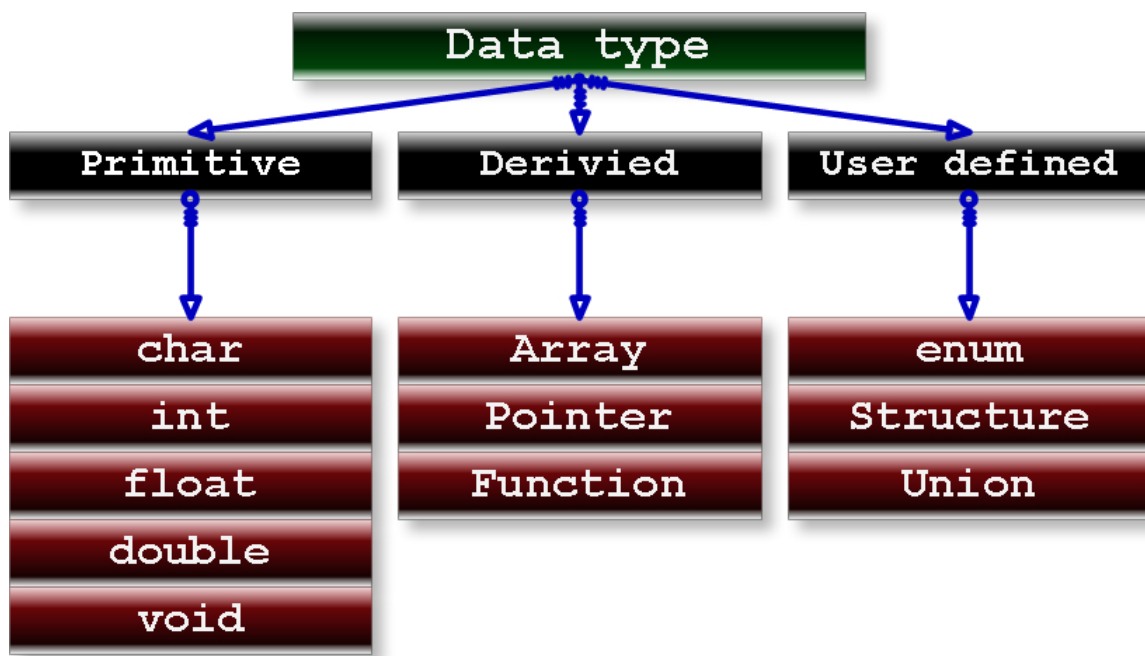
variable different type के data को represent करने के लिए use होते हैं एक प्रोग्राम को रन करते समय इनका अर्थ अपरिवर्तित रहता है किन्तु इनकी value में अंतर आ सकता है वैरिएबल उस नाम को कह सकते हैं जो किसी Memory Location का पता दे रहा हो जहा कोई data स्टोर किया है

उदाहरन के लिए अगर area और Radius दो variable प्रोग्राम की लिए

use किये है तो इनका इनका अर्थ पुरे प्रोग्राम में अपरिवर्तित रहेगा किन्तु Radius का मान बदलने से area का मान भी बदलता रहेगा  
c language में वेरिएबल int,float,char,double आदि हो सकते है

### Data Type In C

c language में data को स्टोर करने के लिए दो अलग अलग भागो में विभाजित किया है जिन्हें क्रमशः Primitive data Type और Drivied data type , User Define Data Type कहते है





## **Operator in C**

c language में ऑपरेटर वे संकेत होते हैं जिनका use Arithmetic और लॉजिकल कैलकुलेशन्स करने के लिए use किये जाते हैं इनका प्रयोग c language में प्रोग्राम data में data और variable को संचालित करने के लिए किया जाता है ऑपरेटर कहलाते हैं  
c language में निम्नलिखित प्रकार के ऑपरेटर होते हैं

1. Arithmetic Operator
2. Relational Operator
3. Logical Operator
4. Assignment Operator
5. Conditional Operator
6. Bitwise Operator

### **1. Arithmetic Operator**

इसका उपयोग Mathematic कैलकुलेशन करने के लिए किया जाता है इसके लिए दो या दो से अधिक operand की आवश्यकता होती है इसे Binary Operator भी कहते हैं

Addition +

Subtraction -

Multiply	*
Division	/

## 2. Relational Operator

रिलेशनल ऑपरेटर का use दो operando के मानों की तुलना करने के लिए किया जाता है

Grater Then	<
Less Then	>
Grater Then equal	>=
Less Then equal	<=
Not Equal	!=

## 3. Logical Operator

एक से अधिक शर्तों की परीक्षा करके किसी निर्णय तक पहुंचने के लिए इन operator का use किया जाता है  
Logical Operator तिन प्रकार के होते हैं

AND	&&
OR	
NOT	!!

## 4. Assignment Operator

c language में किसी value का कैलकुलेशन करने के बाद उसे किसी variable में असाइन करने के लिए Assignment ऑपरेटर का use किया जाता है

EX :

x=a+b ;

## 5. Conditional Operator

c language में इस ओपरेटर को टर्नरी ऑपरेटर कहते हैं इसके साथ दो चिन्ह ? और : का use किया जाता है

Conditional Operator

में कम से कम तिन operand की आवश्यकता होती है

Ex :

```
x = a : b;
```

## Printf () function

Printf c language का एक प्री डिफाइन function है जो stdio.h नाम के हैडर फाइल में है इसका use Message को output screen पर print करने के लिए किया जाता है

printf function का syntax निम्नलिखित है

```
printf("format String ",arg1,arg २,....arg n);
```

Example Of Printf() Function In C

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("engineersworld.in");
    getch();
}
```

## Scanf() Function In C

scanf एक function है यह c language में एक प्री डिफाइन function है जिसका use key bord के data इनपुट लेने के लिए किया जाता है

Syntax of scanf () function

```
scanf("format string ",arg1 ,arg 2 ,arg n);
```

### Example Of Scanf() Function

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    clrscr();
    printf("Enter Value Of A :");
    scanf("%d",&a);
    printf("Enter Value Of B :");
    scanf("%d",&b);
    c=a+b;
    printf("Sum = %d",c);
    getch();
}
```

## Decision Control statements

c language में कुछ कथनों का प्रयोग किया जाता है जिसमे कोई न कोई condition दी जाती है decision making statement में condition के सत्य होने पर ही इन कथनों के अन्दर दिय गय कथनों का ब्लॉक एकजीक्युट होता है decision making statement में दी गई condition

का मान true या false होता है यदि दी गई condition true है तब decision making statement के अन्दर दिए गए कथनों का ब्लॉक एक्जीक्यूट होता है अगर condition false है तब नियंत्रण decision making statement के अन्दर दिए गए कथनों में न जाकर प्रोग्राम के अगले कथनों पर चला जाता है इन कथनों का प्रयोग करने से true या false के आधार पर नियंत्रण के लिए दो मार्ग होते हैं दो मार्गों में से कोई एक मार्ग प्रोग्राम के execution के लिए condition पर आधारित होता है जिसके लिए दो condition का प्रयोग किया जाता है true और false c language में decision making statement निम्नलिखित है -

- if ..... कथन,
- if ..... else कथन,
- nested if कथन ,
- Switch कथन

## **If Statement In C**

if statement के सामान्य दो भाग होते हैं सामान्य if statement एवं if else statement c language में इन्हें नियंत्रण या निर्णय करने वाले statement कहते हैं if statement पहले test expression की testing करता है फिर उसके true और false के आधार पर यह दो भागों में से किसी एक का चयन करता आगे बढ़ जाता है इस तरह यह मुख्यतः द्विमर्गीय पद्धति का सहारा लेता है if statement का use निम्नलिखित प्रकार से किया जा सकता है

सामान्य: if statement में कंप्यूटर if से start होने वाले एक्सप्रेशन या शर्त की परीक्षा करता है यही यह true है तो वह statement block में जा कर परिणाम अंकित करता है यही false है तो वह उसे छोड़ कर वह दूसरे statement को test करता है इसका फ्लोचार्ट निचे दर्शाया गया है

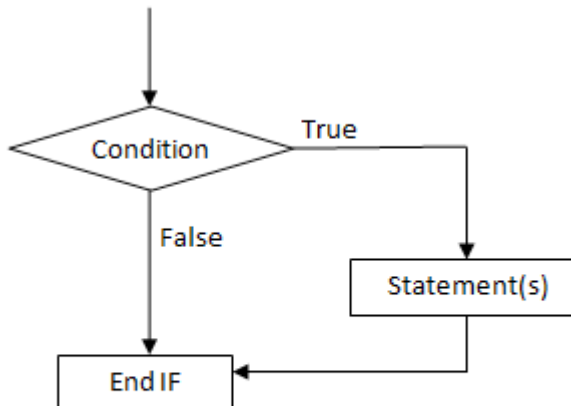


fig: Flowchart for if statement

## Example Of If Statement

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int x;
    clrscr();
    printf("Enter A Number :");
    scanf("%d",&x);
    if(x%2==0)
    {
        printf("%d Is Even");
    }
    getch();
}
```

### If else :

if...else statement if statement का ही एक विस्तृत रूप है इसमें condition के true और false होने पर एक statement या कई statement का ग्रुप एक्सीक्यूट होता है if else statement का Syntax निचे दिया गया है

```
if(test expression)
{
    Statement block1;
```

```
}  
else  
{  
    Statement block 2;  
}
```

If else statement का फ्लोचार्ट निचे इया गया है इसमें यदि condition true है तो statement ब्लॉक 1 एक्सीक्यूट होगा नहीं तो condition false होने पर statement block 2 एक्सीक्यूट होगा if else statement का फ्लोचार्ट

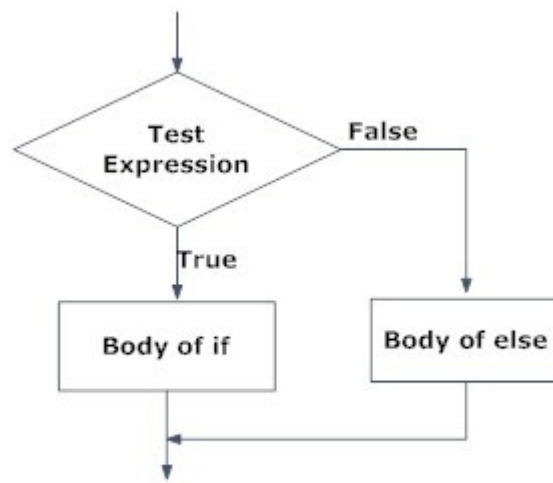


Fig: Operation of if...else statement

## Example Of if else Statement

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    int x;  
    clrscr();  
    printf("Enter A Number :");  
    scanf("%d",&x);  
    if(x%2==0)  
    {  
        printf("%d Is Even");  
    }  
    else  
    {  
        printf("%d is Odd Number");  
    }  
}
```

```
    getch();  
}
```

## Nested if else Statement

if और nested if else statement Control और decision statement हैं यह statement प्रोग्राम की दिशा निर्धारित करते हैं  
nested if statement का use उस Condition में किया जाता है जब एक ही प्रोग्राम में if else के कई statement एक दुसरे से गुथे हुए रूप में प्रस्तुत किये गई हो तब इसे nested if else statement कहा जाता है इसका सामान्य syntax निचे दर्शाया गया है

```
if(condition)  
{  
    Statement;  
}  
else if(condition)  
{  
    Statement;  
}  
else  
{  
    Statement;  
}
```

nested if statement को निचे दिए गई फ्लोचार्ट के द्वारा भी समझ सकते हैं



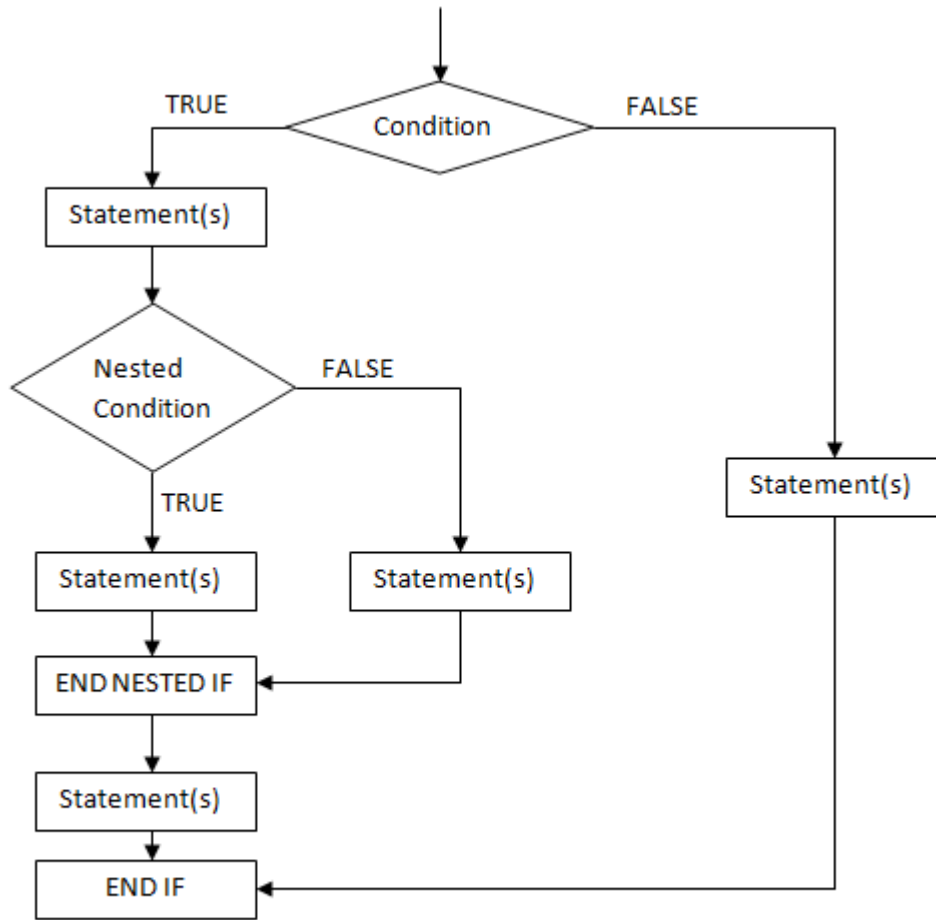


fig: Flowchart for nested if statement

### Example Of Nestedif Statement :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a,b,c;
    printf("Enter 3 Number :");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
    {
        printf("%d is Grater ",a);
    }
    else
    {
        if(b>c)
        {
```

```
        printf("%d Is Grater",b);
    }
    else
    {
        if(c>a)
        {
            printf("%d is grater",c);
        }
        else
        {
            printf("%d is grater ",a);
        }
    }
}
getch();
}
```

## Switch Statement

c language में कभी - कभी एक ही प्रोग्राम में if else का रूप इतना सघन हो जाता है की प्रोग्रामरस्वय ही भ्रमित हो जाता है उन इस्तिथियो में भ्रान्ति से बचने के लिए switch statement का प्रयोग किया जाता है इसमें विभिन्न कार्य करने के लीये statement को अलग – अलग Group में बाट दिया जाता है व हर group को एक case-label के अंतर्गत रखा जाता है switch statement में कोई संख्या या कैरेक्टर variable का मन दिया जाता है जिसमे सभी जिसे सभी case –labels के साथ मेच किया जाता है जिस case label का मान switch

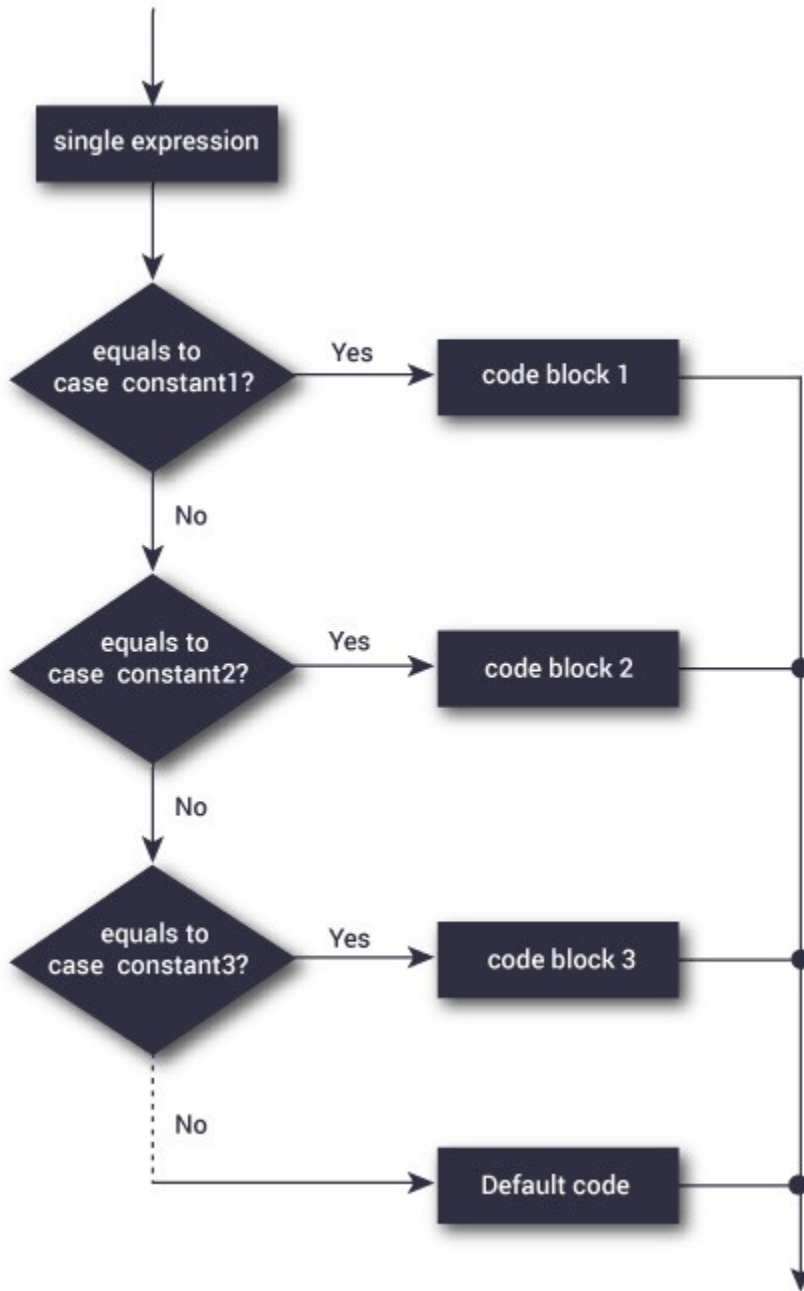
statement में दिए गए मान से मेच करा जाता है उसी के अंतर्गत लिखे गए statement Execute किये हो जाते है कभी कभी प्रोग्राम में केसर label के साथ default label भी रख देते है जब switch statement में दिया गया मान किसी भी case label से मैच नहीं होता है जब default label के अंतर्गत लिखे गए statement execute किये जाते है

switch statement का syntax निचे दिया गया है

```
switch(expression)
{
    Case label 1:
        Statement sequence 1:
        break;

        case label 2 :
        statement sequence 2:
break;
case label 3 :
    statement sequence 3:
break;
:
:
Case label n-1:
Statement sequence n-1;
break;
default:
statement sequence n;
}
```

Switch statement को अधिक स्पष्ट रूप से समझने के लिए फ्लोचार्ट निचे दर्शाया गया है



## Example Of Switch Case Statement

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c,ch;
    clrscr();
    printf("Press 1. Add \n");
    printf("Press 2. Sub \n");
```

```
printf("Press 3. Mul \n");
printf("Press 4. Div \n");
printf("Enter Ur Choice \n");
scanf("%d",&ch);
switch(ch)
{
case 1 :
    printf("Enter Value Of A : ");
    scanf("%d",&a);
    printf("Enter Value Of B : ");
    scanf("%d",&b);
    c=a+b;
    printf("Sum =%d",c);
break;

case 2:
    printf("Enter Value Of A : ");
    scanf("%d",&a);
    printf("Enter Value Of B : ");
    scanf("%d",&b);
    c=a-b;
    printf("Sub =%d",c);
break;

case 3:
    printf("Enter Value Of A : ");
    scanf("%d",&a);
    printf("Enter Value Of B : ");
    scanf("%d",&b);
    c=a*b;
    printf("Mul =%d",c);
break;

case 4:
    printf("Enter Value Of A : ");
    scanf("%d",&a);
    printf("Enter Value Of B : ");
    scanf("%d",&b);
    c=a/b;
    printf("Div =%d",c);
break;
default :
    printf("Wrong Input ");
}
getch();
}
```

## Loop Statement In C

प्रोग्रामिंग के दौरान कई बार कुछ कथनों को शर्त के आधार पर बार – बार दोहराना पड़ता है इस लिए c language control loop statement का use किया जाता है control loop statement दो भागों से मिलकर बने होते हैं

1. loop की body
2. control statement

लूप की बाँड़ी वही भाग होती है जिसे प्रोग्राम में बार बार दोहराया जाता है control statement में दी गई शर्त का परीक्षण करते हैं और इसी शर्त के अनुसार लूप की बाँड़ी का execution बार – बार होता है c language में साधारणतः control loop statement को दो भागों में विभाजित किया जाता है

1. Entry – control Loop :

Entry – control loop में loop की body के एक्सीक्यूट होने से पहले loop के अन्दर दी गई शर्त का परीक्षण पहले होता है उसके बाद ही शर्त के आधार पर loop की body execute होती है जैसे for और while loop entry control loop को प्री टेस्ट loop भी कहते हैं entry control loop का फ्लोचार्ट चित्र में दर्शाया गया है

1. Exit Control loop :

exit control loop में कम से कम loop की body एक बार एक्सीक्यूट होते हैं उसके बाद control Condition test होती है जैसे do while loop exit control loop को post test loop भी कहते हैं

### for Loop :

for loop एक entry control loop है यह c language में सबसे जादा use होने वाला loop है इसका syntax निम्लिखित है

```
for(initialization;condation;updation)
{
    Body of loop'
}
```

जब For loop use किया जाता है तो सर्वप्रथम loop संगणक का प्रारम्भिक मान निश्चित किया जाता है तत्पश्चात loop संगणक का मान परीक्षण शर्त से मिलाया जाता है यदि परीक्षण सत्य है तो loop एक्सीक्यूट होता है for संगणक का मान बढ़ाया या घटाया जाता है परीक्षण शर्त का पुनः परीक्षण किया जाता है आगे for भी संगणक कका मान परीक्षण में सत्य आया जाता है तो loop एक बार for चलाया 'जाता है यह क्रम तब तक चलता है जब तक शर्त का परीक्षण असत्य न हो जाए एसा होते ही loop स्वतः ही समाप्त हो जाता है loop के end होने पर program का control loop के बाहर लिखे गए statement पर आ जाता है for loop का फ्लोचार्ट निचे दर्शाया गया है

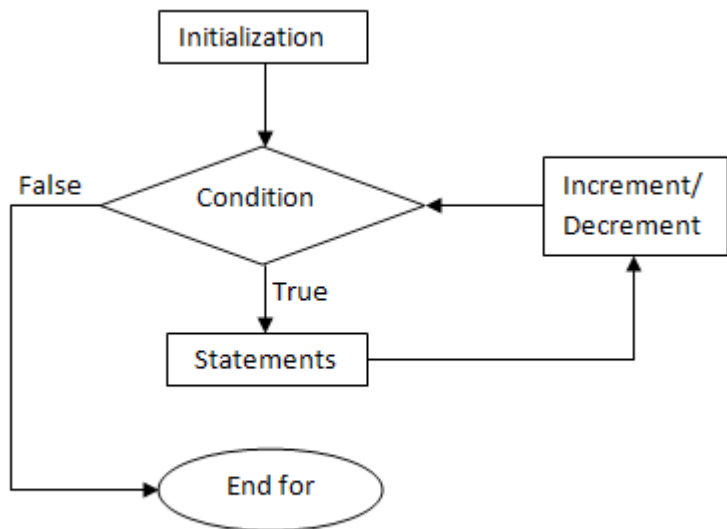


fig: Flowchart for for loop

### Example Of For Loop

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

### While Loop :

while control statement c language का महत्वपूर्ण statement है इसका सिंटेक्स निम्नलिखित है

```
while(test condition)
{
    body of loop ;
}
```

यह पर loop की body में सिर्फ एक ही statement भी हो सकता है व एक से ज्यादा भी हो सकता है एक से जादा statement होने की दशा में कार्ली ब्रेसेस का use किया जाता है ये सभी statement अर्थात loop को body तभी एक्सीक्यूट होगी जब test condition का परीक्षण true होता है इस condition में loop की body में लिखे गए statement रन किये जाते है तत्पश्चत test condition का एक बार पुनः परीक्षण होता है अगर यह पुनः सत्य होता है तो फिर body execution किया जाता है यह क्रम तब तक चलता रखता है तब तक की test condition false नहीं होती है test condition false होने पर loop समाप्त हो जाता है तब प्रोग्राम का control loop के बाद जो पहले कथन होता है उस पर चला जाता है

while loop को entry control loop भी कहते है यह test condition का परीक्षण



पहले होता है और इसके बाद loop की body एक्सीक्यूट होती है  
while loop का syntax :

```
initialization  
while(condition)  
{  
    Statement;  
    Update;  
}
```

while loop का फ्लोचार्ट

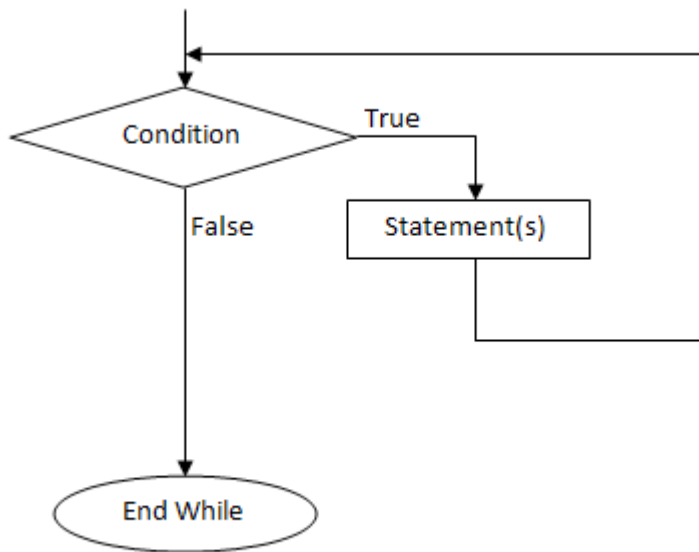


fig: Flowchart for while loop

## Example Of while Loop

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

### do – while Loop :

do ...while loop एक exit control loop है इसका syntax निम्नलिखित है

```
do  
{  
body of loop;  
}while(condition);
```

do .. while loop में पहले loop की body एक्सीक्यूट होती है और अंत में condition test होती है

do while loop को निचे दिए गए फ्लोचार्ट की सहायता से समझा जा सकता है

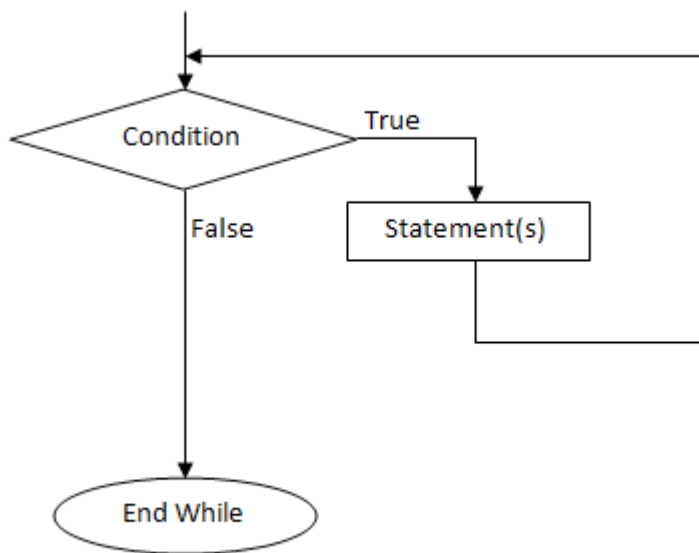


fig: Flowchart for while loop

### Example Of do-while Loop

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
clrscr();  
int a;  
a=0;
```

```
while(a<10)
{
    printf("%d\n",a);
    a++;
}
getch();
}
```

## Break statement

C language break statement का use किसी loop की body के अन्दर किया जाता है है या switch statement की body के अन्दर किया जाता है जब किसी loop के अन्दर break statement मिलता है तो उस जगह loop समाप्त हो जाता है तथा प्रोग्राम का control loop को body के बहार पहले लिखे statement पर आ जाता है जब break statement switch की body के अन्दर मिलत है तो प्रोग्राम का control switch की body के बहार आ जाता है

जब कई loop एक के अन्दर एक करके अर्थात nested loop रो में प्रयुक्त किया जाता है तब break के use से केवल एक loop से बहार निकलने के लिए use किया जाता है फ्लोचार्ट के रूप में break statement को फ्लोचार्ट के रूप में दर्शाया गया है

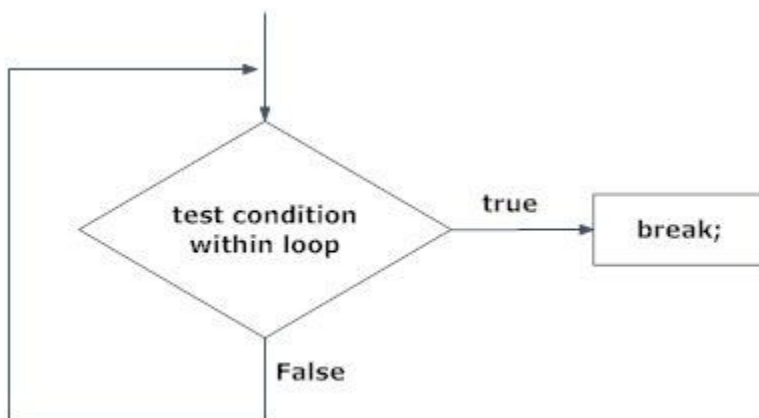


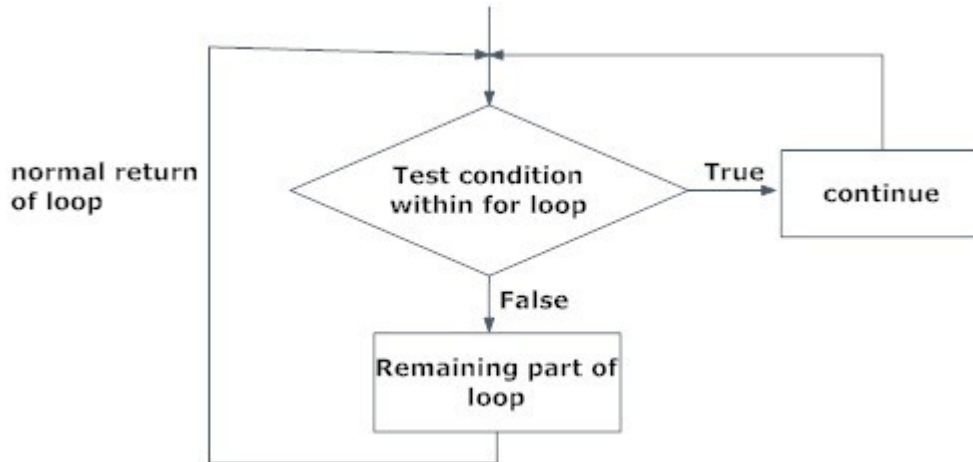
Figure: Flowchart of break statement

## Example Of Break Statement

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i;
    for(i=1;i<=10;i++)
    {
        if(i>6)
        {
            break;
        }
        printf("%d\n",i);
    }
    getch();
}
```

## Continue statement

continue statement का use किसी भी loop का control वापस loop की starting पर पहुच जाता है इसका use भी किसी loop के साथ किया जाता है जब किसी शर्त के परीक्षण के आधार पर यह तय किया जाता है loop के बाकि कथन छोडना है अथवा नहीं अगर परीक्षण सत्य हो तो continue कथन loop के बाकि statement छोड़ दिए जाते है व प्रोग्राम का control वापस loop की starting में चला जाता है continue statement फ्लोचार्ट के द्वारा समझाया गया है continue statement निचे दिए गए उदाहरन से स्पस्ट समझा जा सकता है



## Example Of continue Statement

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

## Array in c

c language में एक प्रकार के data समूह को अरे कहा जाता है यह data कंप्यूटर की memory में क्रमबद्ध तरीके से एक साथ रखा जाता है अरे की एक इकाई को एलिमेंट कहा जाता है किसी भी अरे में सभी data एलिमेंट एक ही प्रकार के होना चाहिए यदि एरे intiger संख्या का है तो तो उसमे

सिर्फ int value हो स्टोर हो सकती है एक अरे में कुछ एलिमेंट int ,कुछ float और कुछ char नहो हो सकते है एरे के किसी भी अवयव को प्रदर्शित करने के लिए एरे की नाम के बाद कोष्टक में index लिखते है उदहारण की लिए यदि किसी एरे का नाम marks है तो एरे marks[5] use एरे में पाचवे एलिमेंट को प्रदर्शित करेगे

एरे का use प्रोग्रामिंग में बहुत लाभ दायक होत है क्योकि इसके मध्य में एक समूह के हजार अवयव के विषय में data भरा जाता है संगणना की जाती है और परिणाम को भी मुद्रित किया जाता है परीक्षाफल और अंकतालिका एरे के द्वारा ही तयार की जाती है एरे की index 0 से start होती है

एरे का syntax निम्लिखित है  
type variable name [size];

उदाहरन :

```
int marks [5];
```

c language में एरे के प्रकार

- one dimensional array
- two Dimensional array
- multi dimensional array

## one dimensional Array

एक one dimensional Array का use साधारण लिस्ट इत्यादि बनाने में किया जाता है तथा एक ही index variable का use किया जाता है इसको single - subscripted variable भी कहते हैं

उदाहरण के लिए यदि ऐरे का नाम x है तो उसके n अवयव इस प्रकार प्रदर्शित कर सकते हैं

$x[0], x[1], x[2], \dots, x[n]$  इत्यादि one dimensional Array का syntax निम्नलिखित है

type variable name [size];

यह type ऐरे के अवयव का डेटा type प्रदर्शित करता है जैसे int , float ,char आदि

variable name ऐरे का नाम है जिसे ऐरे का memory में शुरुआती एड्रेस (base Address) भी कहते हैं और size ऐरे का size प्रदर्शित कर रही है size के अनुसार ही ऐरे के लिए memory में सुरक्षित होगी

उदाहरण : int group [10];

उपरोक्त उदाहरण में एक group नाम का ऐरे लिया गया है जिसका type int है इसका अर्थ यह है की ऐरे में अभी एलिमेंट int type के स्टोर होंगे

Program Of One Dimensional Array

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10],i;
    printf("Enter Element:");
    for(i=1;i<=10;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=1;i<=10;i++)
    {
        printf(" %d",a[i]);
    }
    getch();
}
```

## Two dimensional Array

two dimensional Array मैट्रिक्स या टेबल को प्रदर्शित करने के लिए use किया जाता है इसमें दो subscript use होते हैं एक रो की संख्या बताने के लिए और दूसरी कोलम की संख्या बताने के लिए इसके अवयव निर्दिष्ट करते समय पहले रो की संख्या लिखी जाती है for कोलम की इसका syntax निम्नलिखित है

```
type array name [row-size][column-size];
```

### Program Of Two Dimensional Array

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a[2][3],i,j;
    printf("Enter Elment :");
    for(i=1;i<=2;i++)
    {
```



```
        for(j=1;j<=3;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    for(i=1;i<=2;i++)
    {
        for(j=1;j<=3;j++)
        {
            printf(" %d",a[i][j]);
        }
        printf("\n");
    }
    getch();
}
```

## Multi - dimensional Array

c language में तिन या तिन से अधिक आयाम के ऐरे के ऐरे का use किया जाता है इस प्रकार के ऐरे को Multi - dimensional Array ऐरे कहते हैं Multi - dimensional Array का syntax निम्लिखित है

type array – name [s1], [s2], [s2].... [sm];

यह s1 ,s2 ...sm Multi - dimensional Array के सभी एलिमेंट का size है

Multi - dimensional Array को निम्लिखित त्रि - आयामी अर्र के द्वारा समझा जा सकता है

```
int arr [4][2][3];
```

## String

c language में करैक्टर या अंको और संकेतो के समूह को string कहते हैं जिसे quotation mark के अन्दर लिखा जाता है एक string को char के एरे के रूप में देखते हैं जिसे string का एरे भी कहते हैं हर string का अंतिम char null करैक्टर (`\0`) होता है जिससे string की समाप्ति का पता चलता है इसलिए null करैक्टर को स्टोर करने के लिए एक करैक्टर एरे का size  $n+1$  होना चाहिए क्योंकि अंतिम जगह पर null "`\0`" स्टोर किया जाता है

उदाहरन के लिए निम्नलिखित statement को देखते हैं

```
while(*p!='\0')
```

इस statement का मान तब तक true रहेगा जब तक string अंत नहीं होगा

उदाहरण :

```
char s[]={ 'i','n','d','i','a','\0' };
```

किसी भी string का प्रत्येक char memory में 1 बाइट लेता है और string का अंतिम char हमेशा null होत है जिसे compiler द्वारा string के अंत में जोडा जाता हो

c language में use होने वाले 5 string library function निम्नलिखित हैं

- strcat()
- strcpy()
- strlen()
- strcmp()
- strlwr()

## gets() function

इस function का use इनपुट string function को पढने के लिए किया जाता gets function में वाक्य अथवा एक या एक से अधिक शब्दों को इनपुट किया जाता है इनपुट string में रिक्त स्थान और tab का use भी किया जा सकता है

इसका syntax निम्नलिखित है

```
gets(string);
```

### Example Of gets() Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char ch[50];
    clrscr();
    printf("Enter A String");
    gets(ch);
    printf("\n%s",ch);
    getch();
}
```

## puts() function in c

puts() function का use string को आउटपुट के रूप में प्रदर्शित करने के लिए किया जाता है यह function भी एक समय में एक ही करैक्टर या variable का मान प्रदर्शित करने के लिए use किया जा सकता है अगर puts() function के कोष्टक में एक से जादा argument देना हो तो तो यह केवल पहले argument का मान ही प्रदर्शित करेगा

जैसे :

```
puts(ch,ans); अथवा puts("Hello");
```

यह statement क्रमशः ch का मान व string Hello का मान ही प्रदर्शित करेगा

puts() function के statement निम्नलिखित प्रकार के हो सकते हैं

```
puts(ans);
```

```
puts("Computer Science");
```

इसके सिंटेक्स का सामान्य रूप निम्नलिखित है

```
puts(string);
```

## Example Of puts() Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char ch[30];
clrscr();
printf("Enter A String");
gets(ch);
puts(ch);
getch();
}
```

## strcat() Function

इस function के द्वारा दो स्ट्रिंग को एक साथ जोड़ा जाता है इसका

syntax निम्नलिखित है strcat(s1,s1);

उदाहरन

```
fname= DEEPENDRA\0
```

```
sname=SINGH \0
```

```
lname =BAGHEL\0
```

परस्पर जोड़ने पर इसका अर्थ निम्नलिखित होगा

```
strcat(fname,sname,lname);
```

आउटपुट:

```
DEEPENDRASINGHBAGHEL\0
```

### Program Of strcat Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
main()
{
char text1[30],text2[30];
clrscr();
printf("Enter First String :");
scanf("%s",&text1);
printf("Enter Second String :")
scanf("%s",&text2);
strcat(text1,"");
strcat(text1,text2);
printf("%s\n",text1);
getch();
}
```

### strcpy() Function

यह function एक string के अवयव को दूसरी string में कॉपी करता है इस function में स्रोत तथा लक्ष्य string के base address का argument में रूप में पास करते हैं

इसका syntax निम्नलिखित है  
strcpy(s1,s2);

### Program Of strcpy Function

```
#include<string.h>
void main()
{
    char ch1[20],ch2[20];
    printf("Enter name");
    gets(ch1);
    strcpy(ch2,ch1);
    printf("%s",ch2);
}
```

### strlen () Function

इस function के द्वारा string के char गिने जाते है जिसे string की लम्बाई कहते है इसका syntax निम्नलिखित है  
n =strlen(string);  
यह पर n एक int variable है जिसे string की लम्बाई कहते है यह function पहले null char तक की string के char को गिनता है

### Program Of strlen Function

```
#include<stdio.h>
#include<string.h>
void main()
{
    char name[20];
    int a;
    printf("Enter name");
    gets(name);
    a=strlen(name);
    printf("Length=%d",a);
}
```

## strcmp () Function

string के लिए use होने वाला library function strcmp() दो string की तुलना करने के लिए use होता है यह तुलना दोनों string के एक एक char के ASCII मान के आधार पर होती है जैसे ही किन्हीं तो char के ASCII मान में अंतर प्राप्त होता है तुलना वहीं रोक दी जाती है एसी स्थिति में दोनों char के ASCII मानों का अंतर function द्वारा लौटा दिया जाता है

syntax :

```
strcmp(s1,s2);
```

### Example Of strcmp() Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    char ch1[]="jerry";
    char ch2[]="ferry";
    int i=strcmp(ch1,ch2);
    printf("%d",i);
    getch();
}
```

## strlwr () Function

इस function का use uppercase string को lowercase string में बदलने के लिए किया जाता है इस function में पैरामीटर के रूप में lowercase string को पास किया जाता है इसका syntax निम्नलिखित है

```
strlwr(string);
```

### **Program Of strlwr Function**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    char upper[50];
    printf("Enter A String");
    scanf("%s",&upper);
    printf("\n After Strlwr : %s",strlwr(upper));
    getch();
}
```



## Function :

### What is function

c language का हर प्रोग्राम एक या एक से अधिक function का समूह है c language के प्रत्येक प्रोग्राम में एक main function होता है यह यह संकेत करता है की प्रोग्राम कहा से शुरू हुआ है और कहा समाप्त हुआ है c language में function 2 प्रकार के होते है

1. library function
2. user – define function

library function पहले से डिफाइन function है जिन्हें यूजर द्वारा परिभाषित करने की आवश्यकता नहीं होती है इस प्रकार के function को सीधे प्रोग्राम में use कर सकते है

जैसे : print,scanf,getch,clrscr,cos,pow,strlen आदि

user define function प्रोग्रामर द्वारा बनाए जाते है एक बार डिजाइन करने के बाद ये function उस प्रोग्राम के लिए library function बन जाते है

### Definition of Function :

एक function स्वय एक पूर्ण प्रोग्राम नहीं होता है जो मुख्य प्रोग्राम के किसी एक उपकार्य को संपदित करता है

अत : function को एक मुख्य प्रोग्राम में use होने वाली तथा एक विशेष कार्य को सम्पादित करने वाली इकाई के रूप में use किया जाता है

सामान्य: एक function को निम्लिखित भागो में बाटा गया है

- फंक्शन डिक्लेरेशन

- फंक्शन कॉल
- फंक्शन डेफिनेशन

## 1. function Declaration

variable की तरह function को भी use करने से पहले डिक्लेअर करना अवश्यक होता है जिसे function प्रोटोटाइप कहते हैं function में function का नाम function का return type argument का type और argument की संख्या घोषित की जाती है

इसका syntax निम्नलिखित है

```
data type function_name (type,type,.....,type);
```

उदाहरन :

```
float xyz(int a,int b);
```

यह a और b argument है और xyz function का नाम है और float function का return type है इसका अर्थ यह है की function float type की value return करेगा

## 2. function call :

मुख्य प्रोग्राम में function को use करने के लिए function के नाम का use करते हैं function के नाम के बाद कोष्टक में पैरामीटर को कोमा के द्वारा विभक्त करते हैं

function को call करने के लिए syntax निम्नलिखित है

```
function_name(argument, argument..... ) ;
```

उदाहरन :

```
add(x,y);
```

x,y वास्तविक पैरामीटर (actual parameter) है

### 3. function Definition

function Definition में function हैडर तथा function की body होती है function की body में वे सभी निर्देश होते हैं जो function में आवश्यक कार्य को पूरा करने में आवश्यक है function में फार्मल पैरामीटर का use करते हैं

उदाहरन :

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

### 4. Function Prototype :

function प्रोटोटाइप का use compiler को ये बताने के लिए किया जाता है की सम्बन्धित function में कितने argument है व् उनका data type क्या है अतः अन्य शब्दों में कहा सकते हैं की function प्रोटोटाइप एक ऐसा statement होता है जो argument list और argument declaration को निम्नलिखित तरह से प्रस्तुत करा है  
data type function\_name(type a1, type a2,..... type an);

उदाहरण :

```
float xyz (int a,float b);
```

चूँकि यह एक statement है अतः इसके अंत मे सेमीकोलन लगाया जाएगा उपरोक्त statement एक function प्रोटोटाइप है जो compiler को बता रहा है की function xyz में दो argument है जो क्रमशः int व float type के है अतः function एक मान वापस

करेगा जिसका data type float है यदि function कोई और data का मान वापस करेगा तो compiler एरर देगा

### actual parameter

वे parameter जो कालिंग function में use होते हैं actual parameter होते हैं जिनका use main function में किया जाता है

### farmal parameter

वे parameter जो function की डेफिनिशन में use किये जाते हैं उन्हें farmal parameter कहते हैं

वास्तविक तथा फार्मल पैरामीटर को निम्नलिखित उदाहरण के द्वारा समझा जा सकता है

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int avg(int,int);
    int x,y,z;
    clrscr();
    printf("Enter x :");
    scanf("%d",&x);
    printf("Enter y :");
    scanf("%d",&y);
    z=avg(x,y);
    printf("Average is : %d",z);
    getch();
}
int avg(int a,int b)
{
    int c;
    c=(a+b)/2;
    return c;
}
```

## Local variable

Local variable किसी function या ब्लॉक के अन्दर डिक्लेअर किये जाते हैं इन variable का स्कोप स्थानीय होता है अर्थात् in variable का use कोई अन्य function नहीं कर सकता है ये variable केवल उसी function में use किये जाते हैं जहा इन्हें डेक्लेअर किया गया है यदि in variable का प्रारम्भिक मान यूजर द्वारा निश्चित नहीं किया जाता है तो वे कोई भी वंचित मान ले सकते हैं इसे मानो को garbage value कहते हैं

### Example Of Local Variable

```
#include<stdio.h>
#include<conio.h>
fun();
void main()
{
    int b=10,c=5;
    clrscr();
    printf("\n in main B=%d C=%d",b,c);
    fun();
    getch();
}
fun()
{
    int b=20,c=10;
    printf("\n in fun B=%d B\C=%d \n",b,c);
}
```

## Global variable

वे variable जो पुरे प्रोग्राम के दौरान जीवित या सक्रीय रहते हैं वे variable global variable कहलाते हैं ये variable पुरे प्रोग्राम में कही भी use किये जा सकते हैं तथा प्रोग्राम का कोई भी function इसे use कर सकता है सामान्यत : यह main function के बहार डेक्लेअर किये

जाते हैं कोई भी function इसे use करके इसका मान बदल सकता है अतः : global variable का use सावधानीपूर्वक करना चाहिये इसके आलावा global variable पुरे प्रोग्राम में जीवित रहते हैं और memory में जगा घेरते हैं अतः : आवश्यक variable को हो global variable डिक्लेअर करना चाहिये

### Example Of Global Variable

```
#include<stdio.h>
#include<conio.h>
int b=10,c=5;
void fun();
void main()
{
    clrscr();
    printf("\n in main B=%d c=%d \n",b,c);
    fun();
    b++;
    c--;
    printf("\n again in main B=%d c=%d \n",b,c);
    getch();
}
void fun()
{
    b++;
    c--;
    printf("\n in fun B=%d c=%d \n",b,c);
}
```

### call by value

c language में किसी function को मुख्य प्रोग्राम में call करते हैं तो argument भी भेज सकते हैं सामान्यतः : ये argument भेजने को call by value कहते हैं यह कॉलिंग function अपने actual argument की एक कॉपी called function को भेजता है

### Example Of Call By Value

```
#include<stdio.h>
#include<conio.h>
void main()
{
    void changer(int);
    int num,p;
    clrscr();
    printf("\n Enter The Value Of Num\n");
    scanf("%d",&num);
    printf("\n value before call :%d \n",num);

    changer(num);
    printf("\nValue After Call :%d\n",num);
    getch();
}
void changer(int n)
{
    n=n*10;
    printf("\nValue Of Number In Function :%d\n",n);
}
```

### Call by Reference

C language में जब किसी function को मुख्य प्रोग्राम में call करते हैं तो तो call करते समय कुछ argument भी भेज सकते हैं लेकिन इस विधि में पॉइंटर का use किया जाता है call by referance विधि में वास्तविक argument variable का memory address called function को पास करता है अर्थात called function सीधे वास्तविक argument की value में change करता है

उदहारण:

```
#include<conio.h>
#include<stdio.h>
void swap(int *p1,int *p2);
void main()
{
    clrscr();
    int a,b;
    printf("Enter Two Number :");
```

```
scanf("%d%d",&a,&b);
printf("Simple \n");
printf("A=%d \nB=%d\n",a,b);
swap(&a,&b);
printf("Swaped\n");
printf("a=%d \nb=%d",a,b);
getch();
}
void swap(int *p1,int *p2)
{
    int t;
    t=*p1;
    *p1=*p2;
    *p2=t;
}
```

## pointer

c language में प्रोग्रामिंग करने के लिए कई अधिक शक्तिशाली और प्रभावशाली उपकरण (tool) उपलब्ध हैं तथा pointer एक ऐसा ही माध्यम है अन्य variable की तरह ज=ही pointer variable भी use कर सकते हैं जो किसी data का memory में address स्टोर करता है अतः यह कह सकते हैं की pointer एक ऐसा variable है जो किसी अन्य variable के address को होल्ड कर के रखता है c language pointer के use से प्रोग्राम को जटिलता और लम्बाई कम होती है तथा उसकी क्रियाशीलता में वृद्धि होती है pointer कहलाता है

pointer के use को समझने से पहले memory की संरचना को समझे कंप्यूटर की memory में एक एक बाइट की सेल श्रृंखला बंद तरीके से जुड़ी होती है memory स्टोरेज सेल्स का संग्रह होती है memory बढ़ने पर बाइट की संख्या बढ़ती है जब भी किसी variable या निर्देश को फीड करते हैं तो तो कंप्यूटर use उक्त बाइट की सेल में स्टोर कर देता है चूंकि हर बाइट का अपना एक नंबर होता है अतः स्टोर किये जाने वाले variable और उसकी value को बाइट के उस नंबर से अर्थात् उस variable memory



address से जाना जा सकता है हर सेल का अपना एक यूनिक address होता है इस address को pointer की सहायता से ज्ञात कर सकते हैं

```
int i=8;
```

### Example :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i=8;
    clrscr();
    printf("Address Of I :%u\n",&i);
    printf("Address of I :%d",i);
    getch();
}
```

अतः स्पष्ट है की & की सहायता से किसी भी variable का memory location ज्ञात की जा सकती है

### declaration of pointer :

c language में अन्य variable की तरह ही pointer variable भी डिक्लेअर किये जाते हैं pointer variable को डिक्लेअर करने के नियम सामान्य variable को डिक्लेअर करने की रतर्ह ही होते हैं pointer variable को डिक्लेअर करने का syntax :

```
datatype *pvar;
```

pointer variable का declaration उपयुक्त उदाहरन द्वारा समझा जा सकता है

उपयुक्त उदाहरण में ptr int pointer variable है और ptr1 char pointer variable है तथा ptr2 एक यूजर डिफाइन data type है

## Address operator :

संकेत & एक ऑपरेटर है जिसका use किसी variable का address एक्सेस करने के लिए और pointer variable में असाइन करने के लिए किया जाता है

उदाहरण :

```
int m=15,*mptr;
float x=2.54,*xptr;
mpr=&m;
xptr=&x;
```

## Indirection operator :

संकेत \* indirection operator कहलाता है इसका use pointer के द्वारा किसी variable के मान को एक्सेस करने के लिए किया जाता है

उदाहरण :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int m=15,*mptr;
    float x=2.54,*xptr;
    clrscr();
    mptr=&m;
    xptr=&x;
    printf("\n value of m: %d",*mptr);
    printf("\n value of x: %f",*xptr);
    getch();
}
```

जब प्रोग्राम एक्सीक्यूट होता है जब साधारण variable का मान pointer variable के द्वारा प्रदर्शित होता है

## malloc () function

इस function का use कर memory को रन टाइम पर allocate किया जाता है यह function बताए गई size का memory ब्लॉक सुरक्षित रखता है और void type का pointer return करता है लेकिन जरूरत के अनुसार इसका type चंगे कर सकते हैं

इसका syntax निम्नलिखित है :

```
ptr=(cast type*)malloc(size);
```

यह पर ptr cast type का pointer है

## calloc () function

यह एक अन्य memory एलोकेशन function है इसका use प्रोग्राम रन करे के दौरान derived data type जैसे ऐरे स्ट्रक्चर आदि द्वारा मांगे गए memory space को allocate करने के लिए किया जाता है यह पर स्टोरेज के multiple ब्लॉक को allocate करता है और फिर बाइट को 0 पर set कर देता है

इसका syntax निम्नलिखित है

```
*ptr=(cast type*)calloc(n,elem-size);
```

## realloc () function

इस function का use पूर्व में allocate memory space को परिवर्तित करने के लिए किया जाता है इसका syntax निम्नलिखित है

```
ptr= realloc(ptr,size);
```

## free () function

इस function का use रन करते समय allocate की गई memory को रिलीस करने के लिए किया जाता है इसका syntax निम्नलिखित है

```
free(ptr);
```

## Structure :

c language में ऐरे का use एक ही प्रकार के data की संगना के लिए किया जाता है ये data int float या char हो सकता है परन्तु एक array के सभी एलिमेंट एक ही प्रकार के data type के होते हैं ऐरे का data अमिश्रित होता है जबकि structure मिश्रित होता है जो यूजर द्वारा परिभषित होता है अतः structure में एक साथ कई भिन्न प्रकार के data type का use किया जा सकता है

उदाहरण के लिए एक लाइब्रेरी में विभिन्न पुस्तकों का रिकॉर्ड रखना है तो उसके लिए पुस्तक का नाम प्रष्ट क्रमांक व पुस्तक का मूल्य इत्यादी का data एक साथ तैयार करना होगा यह पुस्तक का नाम तथा लेखक का नाम char data है जबकि प्रष्ट क्रमांक एक int data है और पुस्तक का मूल्य float data है c language में इस तरह के different type का data स्टोर करने के लिए structure का use करते हैं एक structure का syntax निम्नलिखित है

```
Struct book  
{  
Char name[20];  
Char author[20];  
Int page;  
Float price;  
};
```

## Union In C

यूनियन भी c language का एक यूजर डिफाइन data type है जिसका syntax structure के सामान होता है परन्तु structure के हर सदस्य की स्टोरेज location अलग - अलग होती है जबकि यूनियन के सभी सदस्यों की स्टोरेज location एक होती है इसका अर्थ है की structure में अनेक प्रकार के data type के सदस्य हो सकते हैं किन्तु उनकी स्टोरेज location एक होने के कारण एक समय पर उसमे से किसी एक का use एक बार में किया जा सकता है

जैसे :

```
Union abc  
{  
Int a;  
Char c;  
Float d;  
}code;
```

## File Handling In C

जब भी हम c में कोई प्रोग्राम बनाते हैं तो उस प्रोग्राम में हम किसी ना किसी प्रकार के data का use करते हैं वह data तब तक ही कंप्यूटर की memory में रहता है जब तक प्रोग्राम एक्सीक्यूट होता है जैसे ही प्रोग्राम एंड होता है डाटा कंप्यूटर की memory से नष्ट हो जाता है

यदि हमें अगली बार data की आवश्यकता होती है तो हमें वह data नहीं मिल पाता है इस समस्या को खतम करने के लिए c में फाइल हैंडलिंग का use किया जाता है c में फाइल हैंडलिंग का use से c प्रोग्राम में use किया जाने वाला data फाइल के रूप में स्टोर किया जाता है ताकि बाद में प्रोग्राम के end होने के बाद भी यदि हमें उस data की आवश्यकता होती है तो उसे हम use कर सकते हैं

दुसरे शब्दों में कहे तो c में data को परमानेंट फाइल के रूप में स्टोर करने के लिए फाइल हैंडलिंग का उपयोग किया जाता है

### What Is File :

file का तात्पर्य एक ऐसे ऑब्जेक्ट से होता है जो किसी ना किसी प्रकार के data को कन्टेन्ट कर के रखती है

अर्थात दुसरे शब्दों में कहे तो फाइल का उपयोग किसी ना किसी प्रकार के data को स्टोर करने के लिए करते हैं और c language में फाइल handling का use कर के फाइल में ही data को write का read किया जाता है

## fopen () Function

fopen() function का use फाइल को open करने के लिए किया जाता है यह function दो पैरामीटर लेती है पहला पैरामीटर उस फाइल का नाम होता है जिसे open करना होता है तथा दूसरा पैरामीटर फाइल opening मोड होता है file opening मोड से तात्पर्य यह है की फाइल को open किस उद्देश्य से किया गया है फाइल opening मोड कहलाते हैं

### **W :**

इस opening मोड का use तब किया जाता है जब फाइल में data को write करना हो किन्तु w opening मोड का use करने पर फाइल में पहले से मौजूद data डिलीट हो जाता है तथा new data write होता है और यदि जिस फाइल में data write करना है वह फाइल निर्धारित पाथ पर मौजूद है तो new फाइल क्रिएट हो जाती है

### **syntax :**

```
fp=fopen("a.txt","w");
```

### **r :**

इस मोड का use फाइल से data को read करने के लिए किया जाता है

### **syntax :**

```
fp=fopen("a.txt","r");
```

### **a :**

इस मोड का use फाइल में data को write करने के लिए किया जाता है किन्तु इसका use करने पर फाइल में पहले से लिखा हुआ data डिलीट नहीं होता है बल्कि पुराने data के बाद new add हो जाता है

### **syntax :**

```
fp=fopen("a.txt","a");
```

### **w+ :**

इस मोड का use फाइल में बहुत से ऑपरेशन परफॉर्म किये जाते हैं जिसे फाइल में data की write करना फाइल से data को read करना data को modify करना in सभी कार्य को करने के लिए इस मोड का use किया जाता है

### **syntax :**

```
fp=fopen("a.txt","w+");
```

### **r+ :**

इस मोड का use करने पर फाइल से data read करने के साथ - साथ फाइल में data को write भी किया जा सकता है और यही फाइल निर्धारित location पर नहीं मिलती है तो यह NULL return करता है इसका use करने से फाइल में पहले से मौजूद data डिलीट हो जाता है

### **syntax :**

```
fp=fopen("a.txt","r+");
```

## **fclose() Function**

इस function का use फाइल को close करने के लिए किया जाता है इस function के एक्सीक्यूट हो जाने के बाद फाइल प्रोग्राम से कनेक्ट नहीं रहती है यह function 1 पैरामीटर लेता है वह फाइल pointer होता है जो उस फाइल को पॉइंट करता है जिसे close करना है

### **syntax :**

```
fclose(file_pointer);
```



**Example :**

```
fclose(fp);
```

## **fprintf()Function**

fprintf() c language का library function है यह stdio.h हैडर फाइल में डिफाइन है इस function को फॉर्मेटेड आउटपुट function भी कहा जाता है इस function का use के फाइल में data को write किया जाता है fprintf() function कहलाता है fprintf() function 3 पैरामीटर लेता है

1. file pointer जो फाइल को represent करता है जिसमें data Write करना है
2. control string जो data को represent करती है की data किस type का है
3. data या variable name जिसमें उपस्थित data को फाइल में write करना है

fprintf() Function का Syntax :

```
fprintf(file_pointer,control_string,variable_name);
```

### **Example Of fprintf() Function**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *fp;
    int a,b,c;
    fp=fopen("d1.txt","w");
    printf("Enter Two Number");
```

```
scanf("%d%d",&a,&b);
fprintf(fp,"a= %d b=%d c= %d",a,b,a+b);
fclose(fp);
}
```

## fscanf() Function

fscanf() c language का library function है यह stdio.h हैडर फाइल में डिफाइन है इस function को फॉर्मेटेड इनपुट function भी कहा जाता है इस function का use कर के फाइल से data को read किया जाता है fscanf() Function कहलाता है fscanf() Function 3 पैरामीटर लेता है

1. file pointer जो फाइल को represent करता है जिसमे से data read करना है
2. control string जो data को represent करती है की data किस type का है
3. variable जिसमे फाइल से data को read कर के store किया जाता है

### fscanf() Function का Syntax :

```
fscanf(file_pointer,control_string,variable_name);
```

### Example Of fscanf() Function

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a,b,c;
    FILE *fp;
    fp=fopen("d1.txt","r");
    fscanf(fp,"%d,%d,%d",&a,&b,&c);
    printf("a= %d b=%d c=%d",a,b,c);
    fclose(fp);
    getch();
}
```

## fputc() Function In C

यह c language का library function है fputc() function stdio.h हैडर फाइल में है इसका use कर के फाइल में data को write किया जाता है fputc() function character by character data को फाइल में right करता है character by character से तात्पर्य यह है की fputc() function का use कर के फाइल में एक बार में एक character ही write किया जाता है fputc() function दो परमिटर लता है

1. पहला पेरामीटर data होता है जिसे फाइल में right करना है
2. तथा दूसरा पेरामीटर file pointer होता है जो फाइल को represent करता है

### fputc() function का syntax :

```
fputc(data,file_pointer);
```

### Example Of fputc() Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    int i;
    FILE *fp;
    char s[]="Deependra Baghel";
    fp=fopen("d1.txt","w");
    if(fp==NULL)
    {
        printf("File Not FOUND");
    }
    else
    {
        for(i=0;i<strlen(s);i++)
        {
            fputc(s[i],fp);
            fclose(fp);
        }
    }
}
```

```
    }  
  }  
  getch();  
}
```

## fputs() Function

fputs() function c का library function है यह function stdio.h हैडर फाइल में है इस function का use फाइल में data को write करने के लिए किया जाता है इस function के द्वारा फाइल एक बार में पूरी एक string write की जाता है fputs() function कहलाता है fputs() function दो पेरामीटर लेती है

1. पहले पेरामीटर data होता है जिसे फाइल में write करना होता है
2. तथा दूसरा पेरामीटर file pointer होता है जो फाइल को represent करता है

### fputs() function का syntax

```
fputs(data,file_pointer);
```

### Example Of fputs() Function

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    char s[10];  
    FILE *fp;  
    fp=fopen("d1.txt","w");  
    printf("Enter Name");  
    gets(s);  
    fputs(s,fp);  
    fclose(fp);  
}
```