Chapter 1.
Introduction C

History Of C :

१९९६ च्या दशकाच्या उत्तराध्धारात operating system आणि hardware यावर अवलंबून नसणाऱ्या Language च्या गरजेपोटी ,'B' Language तयार केली गेली व नंतर ती BCPL मध्ये रुपांतरीत झाली .मात्र Unix Operating System वर अवलंबून असल्यामुळे तिचेही रुपांतर ,१९७२ मध्ये Martin Richards व Dennis Rithie यांनी 'C Language' चा उपयोग Compiler,operating system तसेच software development साठीही होता.या Language मध्ये Program code (instruction चा set) '.c' extention असलेल्या File मध्ये ASCII Text च्या स्वरुपात या फाईलमधील Program Instructions म्हणजे Source Code ,Compiler द्वारें executable करता येतो.'C Language' मधील प्रोग्रामचे Documentation Section,Link Section,Global Section इ. भाग पडतात.

Document Section मध्ये Progam शी संबंधित Documentation तसेच Program चे author या विभागात लिहिता येतात .

Link Seciton मध्ये Program साठी आवश्यक header files specify केल्या जातात,ज्यामध्ये define केलेली functions program मध्ये वापरले जातात.Defination section मध्ये user defined function तसेच library मधील function define करता येतात .

Global Declaration Section मध्ये declare केलेले variables,functions structures इ. आपल्याला संपूर्ण Program मध्ये कोठेही वापरता येतात.main() हे एक pre-defined function आहे व ते कोणत्याही C program मध्ये असावे लागते.({})महिरपी कंस हे प्रत्येक function मध्ये लिहिलेल्या code ची सुरुवात व शेवट दर्शवतात.

C Language चे वैशिष्टय :

१. Procedural Oriented Programming Language(POP) :
२. Case Sensitive Language.
३. Structure Oriented Programming Language :

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

```
void main()
{
 .....;
 .....;
}
```

४. Middle Level Language.

५. Portable Language.

६. C मधील सर्व statement semi colon ने end होतात.

प्रत्येक C Language ची सुरुवात main() ने होते.main() function इतर function ला call करते. #include<stdio.h> हे statement compiler ला standard input output बद्दल सांगते.जर header file दुसऱ्या directory मध्ये असेल तर आपल्याला double quote marks (“ “) वापरावे लागतात. #include command ने standard files main() available होतात.वरील Program मध्ये main() function कोणतेही arguments accept करत नाही.printf() statement,double quote मधील argument screen वर दाखवते.

#include<conio.h> हे statement compiler ला console input output screen चे content clear करण्यासाठी जे clrscr() व ouput screen hold करण्यासाठी जे getch() function use केले जाते त्यांची defination execute करण्यासाठी console input output function मदत करते.

How to start C :

1 Start PC
2 Double click on desktop icon TurboC++ IDE.
3 File -> New
4 Write your program.
5 Save program as File -> Save(program name नंतर तो .c ने save करावा.)Shortcut Key : F2
6 Compile your program(Alt+F9)
7 Solved any mistakes or errors.
8 Repeat step 6.
9 Run your program.(Ctrl+F9).
10 Check your output.(Alt+F5).
11 Stop.

Escape Sequence Character :

1. **\n :** Output Screen ला cursur नवीन ओळीत move करण्यासाठी \n चा use केला जातो.
2. **\t :** Output screen ला result display करताना प्रत्येक शब्दात अंतर निर्माण करण्यासाठी \t चा use केला जातो.

3. **\b :** Output Screen ला डावीकडील alphabets remove करण्यासाठी \b चा use केला जातो.
4. **\a :** alarm.

Input & Output Functions :

Output screen ला data read & write करण्यासाठी जे function use केले जातात.त्या functions ला input output function म्हणतात.

Input & Ouput Functions चे प्रकार :

1. Formatted Input Output Functions : printf(),scanf(),...etc.

1 printf() : printf() हे एक output function आहे.printf() चा use output screen data write करण्यासाठी केला जातो.

Syntax :

```
1 printf("Message");
2 printf("format specifier",variablename);
```

2 scanf() : scanf() हे एक input function आहे.scanf() चा use output screen data read करण्यासाठी तसेच runtime बदल करण्यासाठी केला जातो.

Syntax :

```
scanf("format specifier",&variablename);
```

scanf() मध्ये variable च्या address ला point करण्यासाठी '&'(address of operator) चा use केला जातो.

2. Unformatted Output Functions :putchar(),getchar(),putch(),getch() ,gets(),puts(). .....etc.

1 putch() : putch() हे unformatted console character output function आहे .putch() function चा use करून एकावेळी एकच character output screen ला print केले जाते.

Syntax :

```
putch(char);
```

2 puts() : puts() हे unformatted string output function आहे.puts() function चा use output screen ला string print करण्यासाठी केला जातो.

Syntax :

```
puts(string);
```

Q. Write a C Program to display the following output using Escape Sequece Character.

Output :

```
        Siddesh
  Computer        Dhule
```

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

```
#include<conio.h>
void main()
{
 clrscr();
 printf("\tSiddesh\nComputer\tDhule");
getch();
}
```

Chapter 2
Data Types & Expressions

Token in C :

1 Data Type
2 Variable or Identifier
3 Constant
4 Keywords
5 Operators

1.Data Type : एखादया variable साठी computer मध्ये किती memory space म्हणजेच किती जागा व्यापावी किंवा वापरावी याच्यासाठी compiler ला मदत व्हावी म्हणून Data Type use केला जातो.
Data Type चे प्रामुख्याने ४ प्रकार पडतात.

1 int
2 float
3 char
4 double

1. Integer : ज्यावेळी आपल्याला numerical म्हणजेच number जर output screen ला display करायचे असतील तर integer data type चा use केला जातो.integer data type ला २ bytes memory space use केली जाते.integer data type साठी int keyword आणि "%d" format specifier वापरला जातो.
2. Float : ज्यावेळी आपल्याला दशाश अपूर्णांकाचा number computer screen display करायचा असेल तर float data type चा use केला जातो. float data type ला 4 bytes memory space use केली जाते.float data type साठी float keyword आणि "%f" format specifier वापरला जातो.
3. Double : double data type ला 8 bytes memory space use केली जाते.double data type साठी double keyword आणि "%lf" format specifier वापरला जातो.
4. Character : ज्यावेळी आपल्याला single alphabet ,characterजर output screen ला display करायचा असेल तर character data type चा use केला जातो.character data type ला 1 bytes memory space use केली जाते.character data type साठी char keyword आणि "%c" format specifier वापरला जातो.

2. Variables : ज्यावेळी एखादी value store करण्यासाठी आपल्याला जागेची किंवा location ची गरज असते तर value storage location साठी आपण ज्याचा use करतो.त्याला Variable म्हणतात.

Variable Declaration चे नियम :

१. Variable चे नाव हे alphabet,digit आणि underscore चे एकत्रीकरण असते.
२. Variable ची length हि १ ते ८ character पर्यंत असते.
३. Variable name ची सुरुवात हि alphabet ने केली जाते.
४. Variable name मध्ये semi colon,space अशा spacial symboles चा use केला जात नाही.
५. Underscore चा use केला जातो.

variable declaration:

```
datatype    var1,var2...varN;
```

For e.g :

```
int a,b,c,.....;
```

3. Constants : एखादया Variable ला permenent value म्हणजेच कधीही change न होणारी value set करायची असेल तर त्याच्यासाठी Constant चा वापर केला जातो.

for e.g :

```
const pi=3.14;
```

4.Keywords : Keywords हे असे शब्द आहेत कि ज्यांचा अर्थ आधीच C च्या compiler ला सांगण्यात आलेला आहे.Keyword हे Variable म्हणून आपण वापरू शकत नाही.Keywords ला reserved असेही म्हणतात. C Language ने आपल्याला 32 Keywords provides केले आहेत.

5. Operators : Operator हे special चिन्हं असतात जे mathematical calculation किव्हा दोन variable मधील relation define करण्यासाठी use करतात.

Types Of Operator :

1. **Arithematic Operator :**
2. **Assignment Operator :**
3. **Logical Operator :**

4. Relational Operator :

5. Conditional Operator :

1. **Arithematic Operator :** कोठल्याही प्रकारचे mathematical operation perform करण्यासाठी Arithematical Operators चा use केला जातो.
   for e.g. : +,-,*,/,%,++,-- etc.
2. **Assignment Operator :** जेव्हा आपल्याला एखादया variable ला value assign करायची असेल तर Assignment Operator चा use केला जातो.
   for e.g : =
3. **Logical Operator :** जेव्हा आपण एकापेक्षा जास्त conditions एकाच वेळी execute करायची असतील तर Logical Operators use केला जातो.
   for e.g : AND,OR,NOT etc.

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is true.

4. **Relational Operator :** दोन variable मधील relation define करण्यासाठी Relational Operator use केले जातात.
   for e.g :

Operator	Description	Example
==	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.

<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

5. **Conditional/Ternory Operator : जेव्हा Program हे conditions नुसार execute होतात.त्यांना Conditional Operator म्हणतात.**

for e.g :

? :	Conditional Expression	If Condition is true ? Then value X : Otherwise value Y

Q. Write C Program to print the addition of two numbers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int num1,num2,result;
  clrscr();
  printf("Enter the value of num1 & num2  : ");
  scanf("%d %d",&num1,&num2);
  result=num1+num2;
  printf("Addition : %d\n",result);
  getch();
}
```

Chapter 3

Control Flow Statement

ज्यावेळी user समोर काही तरी condition असेल आणि त्या condition मध्ये जर त्याला काही निर्णय घ्यायचा असेल तर त्यासाठी जे statement use केले जातात.त्यांना **Decision Making** म्हणतात.

Decision Making statement चे प्रकार :

1 if : if हा keyword condition साठी वापरला जातो. जर if मधील condition true असेल तर true block execute होतो आणि जर condition false असेल तर तो direct block च्या बाहेर येतो.

Syntax :

```
if(condition)
{
  true;
}
```

Q. Write a C Program to check the greater number.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int a,b;
  clrscr();
  printf("Enter the value of a & b : ");
  scanf(“%d %d”,&a,&b);
  if(a>b)
    printf("a is greater than b");
  getch();
  }
```

2 if....else : जर if मधील condition true असेल तर true block execute होतो आणि जर condition false असेल तर तो else block मधील statement execute होतील.

Syntax :

```
   if(condition)
   {
     true;
   }
```

```
  else
  {
    false;
  }
```

Q. Write a C Program to check given number is armstrong or not.

```
#include<stdio.h>
#include<conio.h>
int main()
{
  int num,a,b,c,sum,num1;
  clrscr();
  printf("Enter the three digit number : ");
  scnaf(“%d”,&num);
  num1=num;
  a=num%10;
  num=num/10;
  b=num%10;
  num=num/10;
  c=num;
  sum=a*a*a+b*b*b+c*c*c;
  if(num1==sum)
     printf("Number is armstrong");
  else
     printf("Number is not armstrong");
```

```
  getch();
}
```

3 nested if : जेव्हा if मध्ये if लिहितात तेव्हा त्याला nested if म्हणतात.

Syntax :

```
    if(condition)
    {
      if(condition)
      {
        true;
      }
    }
```

Q.Write a C Program to print the marksheet & display the classes.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int rn,m1,m2,m3,tot;
  float per;
  clrscr();
  printf("Enter the roll no,m1,m2,m3 : ");
 scanf(“%d %d %d %d”,&rn,&m1,&m2,&m3);
  tot=m1+m2+m3;
  per=tot/3.0;
  printf("ROLLNO\tM1\tM2\tM3\tTOTAL\tPERCENT\tCLASS\n\n");
  printf(“rn\tm1\tm2\tm3\ttot\tper\t");
  if(per<35)
  {
    printf("Fail");
  }
  if(per>=35)
  {
```

```
   if(per<50)
   {
     printf("Pass");
   }
  }
  if(per>=50)
  {
   if(per<60)
   {
     printf("Second");
   }
  }
  if(per>=60)
  {
   if(per<75)
   {
     printf("First");
   }
  }
  if(per>=75)
  {
   printf("Dist");
  }
getch();
}
```

4 nested if...else : जेव्हा if else statement मध्ये दुसरे if else statement लिहितात.त्याला nested if...else statement म्हणतात.

Syntax :

```
   if(condition1)
   {
     if(condition2)
     {
```

```
      block1;
    }
    else
    {
      block2;
    }
  }
  else if(condition3)
  {
    block3;
  }
  else
  {
    block4;
  }
```

Q. Write a C Program to check the greatest of three numbers.

```
#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)
{
 if(a>c)
 {
  printf("a is greater ");
 }
 else
 {
  printf("c is greater ");
 }
}
else if(b>c)
{
 printf("b is greater ");
}
else
{
 printf("c is greater ");
}
getch();
}
```

5 switch case : ज्याप्रमाणे nested if else चा वापर करून आपण multiple condition check करतो.त्याप्रमाणे switch case चा वापर करून multiple case check करतो.Multiple cases मध्ये आणि user ने दिलेल्या choice मध्ये equality check करण्यासाठी switch case statement चा use केला जातो.

Syntax :

```
switch(choice)
```

```
{
   case 1 : statement 1;break;
   case 2 : statement 2;break;
          :
          :
   case n : statement 3;break;
   default : invalid block;
}
```

प्रत्येक case साठी statement दिली जातात आणि शेवटी break keyword लिहिला जातो त्यामुळे दोन cases एकमेकांपासून seperate वेगळे केले जातात.जर एकही case true नसेल तर तेव्हा default case execute होते.

Q. Write a C Program to check given alphabets are Vowel or Consonent.

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

void main()
{
  char ch;
  clrscr();

  printf("Enter the alphabet : ");
  scanf(“%c”,&ch);
```

```
 switch(ch)
 {
   case 'a' : printf("Vowel"); break;
   case 'e' : printf("Vowel");break;
   case 'i' : printf("Vowel"); break;
   case 'o' : printf("Vowel");break;
   case 'u' : printf("Vowel"); break;
   default : printf("Consonent");

 }
 getch();
}
```

Loop : Programs मध्ये instuctions set अनेक वेळा execute करावे लागतात.त्यासाठी Loop वापरतात.

Loop चे प्रकार :
1 while loop/Pretested loop/Entry control loop :
2 for loop/Pretested loop/Entry control loop :
3 do while loop/Post tested loop/Exit control loop :
4 nested for loop :

1 while loop/Pretested loop/Entry control loop : while loop प्रत्येक वेळी पहिले condition check करतो आणि जो पर्यंत condition false होत नाही तो पर्यंत while loop repeat होतो.त्यामुळे while loop ला pretested loop किवा entry control loop म्हणतात.
Syntax :

```
 initialization;
 while(condition)
 {
   statement;
   increment/decrement;
 }
```

Q. Write C Program to print the following message 5 times “MKCL”

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int i;
  clrscr();
  i=1;
  while(i<=5)
  {
    printf(“MKCL\n”);
  }
}
```

Output :

MKCL
MKCL
MKCL
MKCL
MKCL

2 for loop/Pretested loop/Entry control loop : Program मधील statement विशिष्ट वेळा एखादया counter मदतीने execute करायचे असतील तर for loop वापरतात.for loop मध्ये पहिले condition check होते म्हणून for loop ला pretested loop किवा entry control loop म्हणतात.

Syntax :

```
for(initialization;condition;increment/decrement)
{
  statement block;
}
```

for loop मध्ये 3 section असतात.यामध्ये प्रत्येक section नंतर semicolon (;) दिला जातो.

1 Variable initialization Section : यामध्ये variable ला initial value assign केली जाते.पण हे compulsory नसते.

2 Condition Section : यामध्ये counter वर based condition दिली जाते.condition true आहे तो पर्यंत loop मधील statement चा block execute होत राहतो.

3 Incerment & Decrement Section : यामध्ये counter ची value वाढवली किवा कमी केली जाते.

Q. Write a C Program to print the factorial of given number.

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

3 do while loop/Post tested loop/Exit control loop : do while loop हा आधी statement block execute करतो ,मग increment or decrement perform करतो आणि शेवटी condition check करतो.म्हणून do...while loop ला post tested loop किंवा exit control loop असेही म्हणतात.do....while loop मध्ये condition जरी false असेल तरी कमीतकमी एक वेळा तरी loop execute होतो.do while च्या while ला compulsory semicolon दिला जातो.

Syntax :

```
initialization;
do
{
   statement block;
   increment/decrement;
}while(condition);
```

Q. Write a C Program to the 1 to 10 numbers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int i;
 clrscr();
 printf("1 to 10 numbers : \n");
 i=1;
 do
 {
  printf("%d\n",i);
  i++;
 }
while(i<=10);
getch();
}
```

4 nested for loop : ज्यावेळी user ला rows आणि column च्या form मध्ये data represent करायचा असेल त्यावेळी nested for loop चा use केला जातो.nested for loop मध्ये for मध्ये for use केला जातो.म्हणून त्याला nested for loop म्हणतात.

Syntax :

```
 for(initialization;condition;increment/decrement)
 {
   for(initialization;condition;increment/decrement)
   {
     statement block;
   }
 }
```

Q.Write a C Program to print the following series .

1

1 2
1 2 3
1 2 3 4
1 2 3 4 5

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

Chapter 4
Array

आपण आता Array हे Chapter strat करणार आहोत. Array हा derived data type आहे. Arrays चा use & need काय हे आपण आज एका example वरून शिकणार आहोत.

Suppose मला 10 student चे roll no.s print करायचे आहेत.तर आपण काय करू.सगळ्यात आधी roll no. चा data type integer असल्यामुळे आपण int data types असलेल 10 variables declare केले.प्रत्येक variable साठी values read करणार आणि write करणार.पण त्यात एक problem असा येईल की, Suppose 10 student चा data increse झाला आणि 100 student झाला मग तुम्ही प्रत्येक वेळी 100 variables declare करणार का? पुन्हा त्या 100 variables साठी values read & write करणार .जर तुम्हाला विचारले की 46 roll no.s चा student कोणत्या variable मध्ये store आहे याच answer आपल्याला लगेच देता येणार नाही.मग या सर्व problems चे solution म्हणून आपण Array Chapter शिकणार आहोत .

Defination : Array हा एकाच data type म्हणजेच similar data type च्या values चा set असतो.

Array चे प्रकार :

1 One Dimentional Array :
2 Two Dimentional Array :

1 One Dimentional Array : ज्यावेळी आपल्याला no. of elements एकाच variable मध्ये store करायचे असतील त्यावेळी आपण एकच subscript use करतो.त्याला One Dimentional Array म्हणतात.

Declaration Of Array :

```
datatype arrayname[size_of_arrays];
```

For e.g. :

```
int a[10];
float b[10];
```

Array declare करताना त्याचा data type,त्याचे नाव आणि size द्यावी लागते.Array चा पहिला element हा a[0] ने ओळखला जातो. 0 ला offset असे म्हणतात.ही array element ची memory allocation मधील relative position असते.Offset ची सुरुवात ही नेहमी 0 ने होते, त्यासाठी एखादे variables हे वापरता येतात.

Initializing Arrays :

Array initialize करताना माहिती कंसात एका sequence मध्ये elements ने separate करून लिहिता येते. प्रत्येक element हा comma ने separate होतो.

for e.g. :

```
int a[5]={2,4,34,3,4};
```

Q. Write a C Program to print the array element.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int a[5],i;
  clrscr();

  printf("Enter the array element : \n");
  for(i=0;i<5;i++)
  {
    scanf("%d",&a[i]);
  }

  printf("Array Element : \n");
  for(i=0;i<5;i++)
  {
    printf("%d\n",a[i]);
  }
getch();
}
```

Output :

Array Element :

1
2
3
4
5

2 Two Dimentional Array : 2D Array ला अधिक dimentions असू शकतात.Two Dimentional Array ला matrix असेही म्हणतात.कारण 2D array हा rows आणि columns च्या form मध्ये store केला जातो.

Declaration Of 2-D Array :

datatype arrayname[size_of_rows][size_of_columns];

for e.g. :

int a[3][3];

Initializing 2-D Arrays :

for e.g. :

int a[2][4]={{1,2,3,4},{4,5,6,9}};

Q. Write a C Program to print the matrix element.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int a[3][3],i,j;
  clrscr();
   pritnf("Enter the matrix element : \n");
  for(i=0;i<3;i++)
  {
    for(j=0;j<3;j++)
    {
      scanf("%d",&a[i][j]);
    }
  }

 printf("Array of matrix Element : \n\n");
 for(i=0;i<3;i++)
 {
   for(j=0;j<3;j++)
   {
```

```
    printf("%d ",a[i][j]);
   }
  printf("\n");
 }
 getch();
}
```

Output :

```
 Array of matrix element :
 11  22  33
 44  55  66
 77  88  99
```

Chapter 5
String

Definition : String म्हणजे no. of characters चा संच .आतापर्यंत आपण single alphabets,character print करण्यासाठी character data type चा use करत होतो.पण त्यामुळे फक्त single alphabet,symbole,character print होत होते.जर आपल्याला no. of alphabet चे collection म्हणजेच एखादा word print करायचा असेल तर त्याच्यासाठी आपण array ची concept use करतो.Alphabets चा संच तयार करण्यासाठी आपल्याला character चा array तयार करावा लागेल .Character चा array म्हणजेच collection of one dimentional character its called as String.

String ही नेहमी double quotation mark मध्येच declare केली जाते.String हा one-dimentional character array असून तो नेहमी '\0' ने terminate होतो.Character array continueous memory location मध्ये store केला जातो. '\0' या NULL Character नेच string संपते. '\0' आपण देण्याची गरज नाही तो आपोआप घेतला जातो.Character array हा loop चा वापर न करताही %s या format specifier चा वापर करून accept करता येतो.तसेच वापरता येतो.

Syntax :

```
datatype stringname[size_of_string];
```

for e.g. :

```
char city[20];
```

Initialization of String Variable :

तुम्ही character चा array पुढीलप्रमाणे initialize करू शकतात.

for e.g. :

```
char city[20]="Siddesh";
```

Read the String : scanf() चा वापर करून string पुढीलप्रमाणे लिहितात.

```
scanf("%s",city);
```

String read करताना आपल्याला (&) address of operator ची गरज नसते.

C च्या Library ने आपल्याला ready made string function provide केले आहेत. हे सर्व Library Functions जर आपल्याला use करायचे असतील तर string.h नावाची header file आपल्याला include करवी लागते.

Types Of String Function :

1. **strlen() :** String Length म्हणजेच strlen() चा use दिलेल्या string ची length mean दिलेल्या string मध्ये no. of किती characters present आहेत हे count करण्यासाठी केला जातो.

   Syntax :

   ```
   strlen(string);
   ```

2. **strcpy() :** String Copy म्हणजेच strcpy() चा use एका string ची value दुसऱ्या string मध्ये copy करण्यासाठी केला जातो.

   Syntax :

   ```
   strcpy(target,source);
   ```

3. **strcat() :** String Concantenation म्हणजेच strcat() चा use जेव्हा आपल्याला दोन string एकत्र करायच्या असतील किवा दोन string ची addition करायची असेल.अशा वेळी केला जातो.

   Syntax :

   ```
   strcat(target,source); or strcat(source,target);
   ```

4. **strcmp() :** आपल्याला जेव्हा दोन string compaire करायच्या असतील ,त्या दोन string equal आहे की ,greater हे check करण्यासाठी string compaire म्हणजेच strcmp() चा उसे केला जातो.

   0 ---------- > both strings are equal
   >0 ---------- > first string is greater
   <0 ---------- > second string is greater

5. **strrev() :** आपल्याला जेव्हा दिलेली string reverse म्हणजेच उलट print करायची असेल,त्यासाठी strrev() function चा use केला जातो.

   Syntax :

   ```
   strrev(string);
   ```

6. **strupr() :** आपल्याला जेव्हा दिलेली string upper case मध्ये print करायची असेल,त्यासाठी strupr() function चा use केला जातो.

Syntax :

strupr(string);

7. **strlwr() :** आपल्याला जेव्हा दिलेली string lower case मध्ये print करायची असेल,त्यासाठी strlwr() function चा use केला जातो.

Syntax :

strlwr(string);

Q.Write a C program to perform the all string operation using menu driven.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  char str1,str2;
  int ch,len=0;
  clrscr();
  printf("*************** Menu Driven ******************\n\n");
  printf("1 : strlen()\n2 : strcpy()\n3 : strcat()\n4 : strcmp()\n5 : strrev()\n6 :  strupr()\n7 :
strlwr()\n");
  printf("Enter your choice : ");
  scanf("%d",&ch);
 switch(ch)
 {
   case 1 : printf("Enter the string : ");
            scanf("%s",str1);
            len=strlen(str1);
            printf("String Length : %d\n",len);break;
   case 2 : printf("Enter the String : ");
            scanf("%s",str1);
            strcpy(str2,str1);
            printf("Copied String : %s\n",str2);break;
   case 3 : printf("Enter the 1st & 2nd String : ");
            scanf("%s %s",str1,str2);
            strcat(str2,str1);
            printf("Concate String : %s\n",str2);break;
   case 4 : printf("Enter the 1st & 2nd String : ");
            scanf("%s %s",str1,str2);
            if(strcmp(str1,str2)==0)
                 printf("Both Strings are equal");
            else if(strcmp(str1,str2)>0)
```

```
            printf(“1st string is greatest”);
        else
            printf(“2nd string is greatest”);break;
  case 5 : printf(“Enter the String : ”);
        scanf(“%s”,str1);
        strrev(str1);
        printf(“Reverse String : %s\n”,str1);break;
  case 6 : printf(“Enter the String : ”);
        scanf(“%s”,str1);
        strupr(str1);
        printf(“Upper Case String : %s\n”,str1);break;
  case 7 : printf(“Enter the String : ”);
        scanf(“%s”,str1);
        strlwr(str1);
        printf(“Lower Case String : %s\n”,str1);break;
 default : printf(“Invalid Case”);
 }
getch();
}
```

Chapter 6

Structure

Defination of Structure : Structure म्हणजेच different data type च्या values चा set असतो.

Structure हा user defined data type तयार करण्यासाठी struct हा keyword आहे.

Declaraction Syntax :

```
struct structurename
{
    Datatype 1 Member1;
    Datatype 2 Member 2;
    Datatype 3 Member3;
}objectname;
```

Accessing Variables of Structure Type :

For e.g :

```
struct student
{
   char name;
   float per;
   int roll;
} s1,s2,s3;
```

Structure declare केल्यानंतर आपण Structure ची Variable declare करू शकतो.

e.g. : Structure definition नंतरचे s1,s2,s3.Structure variables declare करताना त्यांचा Tag name आपण exclude करू शकतो,पण असे केल्यास variables ही Structure declaration च्या बरोबर दयावी लागतात.ज्यामुळे आणखी Variables आपण नंतर add करू शकत नाही.Structure Variables access करण्यासाठी आपण dot(.) operator चा वापर करतो जसे s1.per,s2.roll etc.

Write a c program to display the book information using structure.

```
#include<stdio.h>
#include<conio.h>
struct book
{
```

```
        char title[20];
        int pages;
        float price;
} b;

Void main()
{
        clrscr();
        printf("Enter the book title,pages,price : ");
        scanf("%s %d %f",&b.title,&b.pages,&b.price);
        printf("Title : %s\n",b.title);
        printf("Pages : %d\n",b.pages);
        printf("Price : %f",b.price);
       getch();
}
```

Structure Arrays :

Structure मध्ये कोणताही elements असतील तर ती नेहमी contiguous memory locations मध्ये store होतात.यामध्ये Structure array ची Row & column elements ही structure members च असतात.

For e.g :

```
struct student
{
   char name;
   float per;
   int roll;
} s1[5];
```

Write a c program to display the book information using structure array.

```
#include<stdio.h>
#include<conio.h>
struct book
```

```
{
        char title[20];
        int pages;
        float price;
} b[5];

void main()
{
        int I;
        clrscr();

        for(i=0;i<5;i++)
        {
                printf(“Enter the book title,pages,price : ”);
                scanf(“%s %d %f”,&b[i].title,&b[i].pages,&b[i].price);
        }
       for(i=0;i<5;i++)
       {
                printf(“Title : %s\n”,b[i].title);
                printf(“Pages : %d\n”,b[i].pages);
                printf(“Price : %f”,b[i].price);
       }
       getch();
}
```

Structure with Function : structure चा variable as argument आपण function ला pass करू शकतो & त्याचा structure variable मध्ये function calling होताना changes reflect होत नाही .

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

struct student
{
   char name[50];
```

```
    int roll;
};
void Display(struct student stu);
void main()
{
    struct student s1;
    printf("Enter student's name: ");
    scanf("%s",&s1.name);
    printf("Enter roll number:");
    scanf("%d",&s1.roll);
    Display(s1);                    // passing structure variable s1 as argument
    getch();
}
 void Display(struct student stu)
{
  printf("Output\nName: %s",stu.name);
  printf("\nRoll: %d",stu.roll);
}
```

Structure within structure :

एका structure मध्ये दुसरे structure लिहिणे याला structure within structure म्हणतात.

```
/* STRUCTURE WITHIN STRUCTURE ANOTHER WAY PROGRAMM */

#include<stdio.h>
#include<conio.h>
struct date
{
   int dd;
   int mm;
   int yy;
};

void main()
{
  struct  inv
  {
    char inm[15];
    int qty;
    int rate;
    int amt;
    struct date d;
  }list;

  clrscr();
```

```
    printf("Enter the item name, qty and rate,date\n");
            scanf("%s%d%d%d%d%d",      list.inm,&list.qty,&list.rate,&list.d.dd,&list.d.mm,
&list.d.yy);
    list.amt =  list.qty*list.rate;

    printf("ITEM NAME      QTY     RATE      AMT     date\n");
printf("%-5s%5d%5d%7d                                                    %d-%d-
%d\n",list.inm,list.qty,list.rate,list.amt,list.d.dd,list.d.mm,list.d.yy);
  getch();
}
```

Union :

Structure सारखे unions ही derived data type आहेत unions आणि structrues सारखेच दिसतात पण त्यांचा उपयोग वेगवेगळ्या गोष्टीसाठी होतो.Structure आणि union दोन्हीही वेगवेगळ्या प्रकारचे Variables वेगवेगळ्या memory location मध्ये store करतात पण union वापरून एक memory section एकावेळी एका प्रकारचे तर दुसऱ्यावेळी दुसऱ्या प्रकारचे variable म्हणून वापरू शकतो.

Usage of unions : Structure मधले elements वेगवेगळी memory locations वापरतात तर union मधील elements एकच memory location वापरतात म्हणून union memory space वाचवतात .हे multiple elements च्या application मध्ये useful आहेत जिथे सगळ्या elements ना एकाच वेळेला value assign करावी लागत नाही.

```
// Example of Union :
#include<stdio.h>
#include<conio.h>

union birthdate
{
 int day;
 char month[20];
 int year;
}b;

void main()
{
 clrscr();
 printf("UNION SIZE : %d bytes\n\n",sizeof(union birthdate));
```

```
 printf("Enter the day,month,year : ");
 scanf("%d %s %d",&b.day,&b.month,&b.year);
 printf("\n\t\tBirth Date .....\n\n");
 printf("%d-%s-%d\n",b.day,b.month,b.year);
 getch();
}
```

Union elements सुद्धा Structure elements प्रमाणे ' . ' operator नी access करतात.

Union of Structures : ज्याप्रमाणे एक Structure दुसऱ्या Structure मध्ये nest करता येते.त्याचप्रमाणे एक union दुसऱ्या union मध्ये nest करता येत.हेच नाही तर Structure मध्ये union आणि union मध्ये Structure असू शकते.

Chapter 7
Function

Defination : ज्यावेळी आपल्याला particular task perform करायचे असेल .तेव्हा function चा वापर होतो.याच function ला user defined function म्हणतात.

Function Declaration :

```
returntype  functionname(argumentlist);
```

Function Calling :

```
funtionname(argumentlist);
```

Function Defination :

```
returntype  functionname(argumentlist)
{
    Function definition;
}
```

Types of function :

1. No Pass No Return :

   ज्या वेळी आपण function ला कुठल्याही प्रकारचे argument pass केलेले नसतील व function कुठल्याही प्रकारची value return करत नसेल .त्यावेळी ते function "No Pass No Return" type मध्ये येते.

   Declaration Syntax :

   void functionname(void);

   e.g :

   void add(void);

2. Pass But No Return:

   ज्या वेळी आपण function ला argument pass केलेले असतील पण function कुठल्याही प्रकारची value return करत नसेल .त्यावेळी ते function "Pass But No Return" type मध्ये येते.

   Declaration Syntax :

   void functionname(argumentlist);

   e.g :

   void add(int,int);

3. Return But No Pass:

   ज्या वेळी आपण function ला argument pass केलेले नसतील पण function value return करत असेल.त्यावेळी ते function "Return But No Pass" type मध्ये येते.

   Declaration Syntax :

   returntype functionname(void);

   e.g :

   int add(void);

4. Pass and Return:

   ज्या वेळी आपण function ला argument pass केलेले असतील & function value पण return करत असेल .त्यावेळी ते function "Pass & Return" type मध्ये येते.

   Declaration Syntax :

   returntype functionname(argumentlist);

   e.g :

   int add(int,int);

WACP to demonstrate the all user defined function type using Menu Driven.

1 No Pass No Return
2 Pass But No Raturn
3 No Pass But Return
4 Pass & Return.
Solution :

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void add1();
void add2(int,int);
int add3();
int add4(int,int);
int a,b,c;
void main()
{
	int ch;
	clrscr();
	while(1)
	{
		printf("\t\t\tMenu Driven\n");
		printf("1 : No Pass No Return\n");
		printf("2 : Pass But No Return\n");
		printf("3 : No Pass But Return\n");
		printf("4 : Pass & Return\n");
		printf("Default : Exit\n");
		printf("Enter your choice : ");
		scanf("%d",&ch);
		switch(ch)
		{
			case 1 : add1();break;
			case 2 : printf("Enter the two numbers : ");
				 scanf("%d %d",&a,&b);
				 add2(a,b);break;
			case 3 : c=add3();
				 printf("Addition3 : %d\n",c);break;
			case 4 : printf("Enter the two numbers : ");
				 scanf("%d%d",&a,&b);
				 c=add4(a,b);
				 printf("Addition3 : %d\n",c);break;
			default : exit(0);
```

```
                }
        }
  getch();
}
void add1()
{
        printf("Enter the two numbers : ");
        scanf("%d%d",&a,&b);
        c=a+b;
        printf("Addition1 : %d\n",c);
}
void add2(int a,int b)
{
        c=a+b;
        printf("Addition2 : %d\n",c);
}
int add3()
{
        printf("Enter the two numbers : ");
        scanf("%d%d",&a,&b);
        c=a+b;
        return c;
}
int add4(int a,int b)
{
        c=a+b;
        return c;
}
```

Recursion Function : जे function स्व:ताचाच definition मध्ये स्व:तालाच पुन्हा पुन्हा call करत असेल,त्याला Recursive Function म्हणतात.

Program : WACP to calculate the factorial of given number using recursive function.

Solution :

```
#include <stdio.h>
#include<conio.h>
long int factorial(long int i)
{
```

```
   if(i <= 1)
   {
      return 1;
   }
   return i * factorial(i - 1);
}
void main()
{
    printf(“Enter the number : ”);
    scanf(“%ld”,&i);
    printf("Factorial of %ld is %ld\n", i, factorial(i));
    getch();
}
```