PAGE:
DATE: / /

C Programming

Dennis Ritchis
(1972) (1970's)

To write unix O/S

A- Algol (Algorithm)
B- BCPL (Basic Combined Programming language)
C- Structured language.

Structure of C Programming

```
#include <stdio.h>
#include <conio.h>
   main()          - first executable statement of C program.
   {
    - - -
    - - -
    - - -
   }
```

Non executable statement

- Pre processor directive
<> - Start and end delimiter of a file name.

PAGE:
DATE: / /

Translator is of two types

1) Assembler
2) Interpreator
3) Compiler

Compiler :— (i) Syntactical Analysis (Syntax check)
(ii) Lexical Analysis (Grammer check)
(iii) Passing

obj → library link → exe
↓
execute.

There are 32 Keyword in C. All keyword written by small letter.

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

pre processor

<stdio.h> → standard input - output header file

```
main()
{
  //body (Documentation part)
}
```

WAP in C to print message

Printf (" ");
it is general purpose output function.

```
#include <stdio.h>
#include <conio.h>
main()
{
printf ("MCR 1st year");
}
```

User Code : ABC.C
Machine code : ABC.obj
Operating system code : ABC.exe

Scanf (" "),
it is general purpose input function.

Data type in C

Char – A to Z – 1 byte
a to z
Special char
#, @
int 0 to 65556 – 2 byte
float – 4 byte (By default 6 decimal place)
Double – 8 byte (14 decimal)

PAGE:
DATE: / /

$-2^{n-1} - 1$ to 2^{n-1}

1 byte = 8 bit

char $-2^{8-1} - 1$ to 2^{8-1}

-127 to 128

i) Modifiers use to modify data type

char — %c — 1B
int — %d — 2B
float — %f — 4B
Double — %4 — 8B

How to write C program

#include <iostream.h>
#include <conio.h>
main

ii) Operator Arithmatic

+
−
*
/

iii) To show menage on Monitor

printf ("msg");

iv) To accept a value from keyboard

Scanf ("%d", & Var);

PAGE:
DATE: / /

Write a program to check the number whether it is even or odd:-

```
#include<stdio.h>
#include<conio.h>
  main()
  {
  int a;
  clrscr();
printf("Enter the value of a:");
scanf("%d",&a);
if(a%2==0)
  {
printf("a is a even number");
        }
else
    {
printf("a is a odd number");
        }
  getch();
        }
```

v) to print message along with value on Monitor

printf ("Addition = %d", Var);

How to write C program

```
#include <stdio.h>
#include <conio.h>
  main()
  {
  int a;
  clrscr();
  printf ("Enter the value of a");
  Scanf ("%d", &a);
  printf (" Value of a = %d", &a)
        getch();
  }
```

Dry Run:

a	output
50	Enter the value of a : 50 ↵
	Value of a = 50

W.A.P in C to add two numbers.

```
#include <stdio.h>
#include <conio.h>
  main()
  int a, b;
  clrscr();
```

```
printf ("Enter the value of a");
Scanf ("%d", &a);
Printf ("Enter the value of b:");
scanf ("%d" &b);
    c= a+b;
printf ("Addition =%d ", c);
    getch();
        }
```

4 →

```
#include <stdio.h>
#include <conio.h>
  main()
  {
  float C,F;
  clrscr();
  printf("Enter the value of F");
  Scanf("%f",&F);
    C= (5*(F-32))/9;
  printf("Celcius =%f", C);
    getch();
        }
```

(1) $V = (4 \times 3.14 \times r \times r \times r) / 3$

(2) $C = (2 \times 3.14 \times r)$

(1)

```
#include <stdio.h>
#include <conio.h>
  main()
    {
    float r;
```

clrscr();
printf("Enter the value of r");
scanf("%f",&r);
V = (4 X 3.14 X r X r X r) / 3
printf("Volume = %f", V);
getch();
}

→ (2) #include <stdio.h>
#include <conio.h>
main()
{
float r;
clrscr();
printf("Enter the Value of r");
scanf("%f"; &r);
C = 2 X 3.14 X r
printf("Circumference = %f", C);
getch();
}

* If else if statement :—

#include <stdio.h>
#include <conio.h>
main()
int a, b;
clrscr();
printf("Enter the value of a");
Scanf("%d", & a);
printf("Enter the value of b");
scanf("%d"; &b);

PAGE: 08
DATE: / /

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

```
{
if ()
{
}
else
{
}
}
```

Block scope

→
```
#include <stdio.h>
#include <conio.h>
main()
{
int a, b, c;
clrscr();
printf ("Enter the value of a");
Scanf ("%d", &a);
printf ("Enter the value of b");
Scanf ("%d", &b);
if (a>b && a>c)
{
printf ("A is greater");
}
else
{
if (b>c)
{
printf ("B is greater");
}
```

{ — scope of a variable

}

Note:— :: Scope resolution operator (It is used when we want both global and local value).

```
if
{
printf (" C is greater");
}
getch();
}
```

W.A.P in C to check whether the year is leap year or not.

```
#include <stdio.h>
#include <conio.h>
main()
{
int y;
clrscr;
printf ("Enter the Year");
scanf ("%d, &y);
if (y%4==0 && y%100!=0 || y%400==
{
Printf (" Year is leap year");
}
else
{
printf (" Year is not leap year");
}
getch();
}
```

PAGE: 10
DATE: / /

Turboc 2	TurboC 3	Unix	ANSI

main()
{
}
 	
	#
	#
	main()
	{
	
	}
	
		#include
		<stdio.h>
		int main()
		{
		
		return 0;
		}
			#include
			<stdio>
			→ Using name
			space stdi
			→ int main()
			{
			
			return 0;
			}

If (num % 2 == 0)
    even
else
    odd;

C Token :- The smallest, independent and executable units are called Token.
All the Keyword and fundamental units are known as Token.

Keyword:- Reserved word which can not be used as variable and identifier.
→ 32 Keyword.
e.g:- int, float, char, double.

PAGE:
DATE: / /

Identifier :— The name given by a programmer to identify the variable name, a function or an array is known as identifier.

Variable :- The name given to a memory location is known as variable.

i) Arithmatic Operator

( )
/ * % } L to R
+ −
= } R to L
└ Copy Operator

ii) inclination / Declination

x++ } x=x+1 } R to L
++x }

```
#include<stdio.h>
  main()
  {
  int x= 5 ;
  printf ("%d","%d","%d", x++, x++,
                          x++)
          }
```

O/P 7, 6, 5

for loop (count controlled loop):

for (①;②;③)

① → Initialization

② → Condition Operator

③ → Increment decrement

Write a ~~proprar~~ program to find sum of three numbers given by user and find average of them.

```
#include <stdio.h>
#include <conio.h>
    main()
    {
    int a, b, c, d;
    float Avg;
     clrscr();
printf (" Enter the value of three
                numbers:");
scanf ("%d %d %d", &a, &b, &c);
        d = a+b+c;
printf (" Sum of three number = %d", d)
    Avg = (a+b+c) /3
printf ("Average = %f ", Avg);
        getch();
        }
```

Bitwize Operator

>> } Shift Operator
<<

5 >> 2
2.5

→ for loop :-

The for loop is another entry-controlled loop that provides a more concise loop control structure. The general form of the for loop is

```
for (initialization ; test condition;
      increment )
{
  body of the loop
}
```

It is count controlled loop.

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

PAGE :
DATE : / /

W.A.P to sum 10 natural numbers -

```
#include <stdio.h>
#include <conio.h>
  main()
  {
  int i, sum;
  clrscr();
  for (i=1; i<=1; i++)
    {
   sum = sum+i;
      }
  printf ("sum %d"; sum);
    getch()
      }
```

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

O/P

1
2
3
4
5
6
7
8
9
10

PAGE:
DATE: / /

Nested loop :- When a loop contains another loop in its body, then it is called Nested loop.

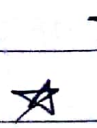
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

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

i	j	Output
1	1②	1
2	12③	2 2
3	123④	3 3 3
4	1234⑤	4 4 4 4
5	12345⑥	5 5 5 5 5
6		

* Write a program in c to find Simple Interest :-

```
#include <stdio.h>
#include <conio.h>
  main()
   {
   int SI, P, R, T;
    clrscr();
printf("Enter the value of P:");
scanf("%d", &P);
printf("Enter the value of R:");
scanf("%d", &R);
printf("Enter the value of T:");
scanf("%d", &T);
 SI = (P*R*T)/100;
printf("SI = %d", SI);
     getch();
      }
```

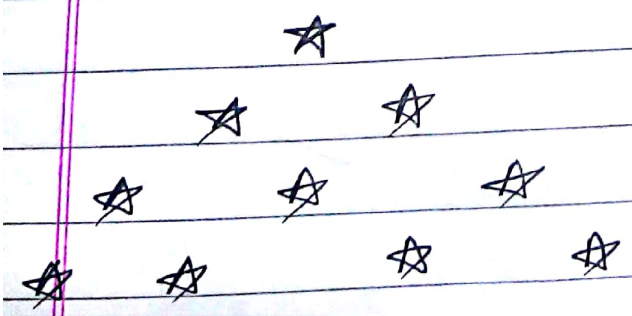


2.

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

```
*
* *
* * *
* * * *
* * * * *
```

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



PAGE:
DATE: / /

```
#include <stdio.h>
#include <conio.h>
  main()
   {
  int i, j, col, row;
  col = 40; row = 12;
  for (i=1; i<=7; i+=2)
   {
  gotoxy (col, Row);
  for (J=1; j<=I; J++)
   {
  printf<<"*";
      }
  col = col - 1;
  row = row + 1;
      }
  getch ();
      }
```

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

PAGE:
DATE: / /

Note :— All program needs of complexcity (Time and Space)

i.e., Space $\propto \frac{1}{\text{time}}$.

Write a programme to Reverse a number

num = 123
Reverse = 321

```
#include<stdio.h>
#include<conio.h>
 main()
 {
int num = 123; rem; num
for ( ; num>0; num/10)
 {
rem = num %10;
printf("%d", rem);
   }
get ch();
    }
```

```
# include <stdio.h>
#include<conio.h>
 main()
 {
int num, rem, rev=0;
clrscr();
printf(" Enter the value of num");
scanf("%d", &num);
for ( ; num>0 ; ) or while (num>0)
 {
 rem = rem %10;
 rev = rev * 10 + rem;
```

```
num = num / 10;
  }
printf ("\n Reverse = %d", rev);
    getch();
        }
```

31/08/17

W.A.P in C to check palindrome or not.

```
#include<stdio.h>
#include<conio.h>
 main()
{
int rev = 0, rem, num, num1;
clrscr();
printf("enter no:");
scanf("%d", &num);
num1 = num;
while (num>0)
 {
rem = num % 10;
rev = rev * 10 + rem;
num = num / 10;
  }
if (num1 == rev)
{
printf("\n palindrome");
  }
else
  {
```

printf ("\n Not palindrome");
}
getch ();
}

rev	num	num	num 1
	1	121	121
0		12	
1	2	1	
12	1	0	
121			

★ main ()
{
while ()
{

}
if (num 1 == num)
{

}
else
{
}
getch ();
}

PAGE:
DATE: / /

1 – D array

⊙ To display 5 array element :—

```
#
#
main()
{
int a[5], i;
printf("\n Enter array element");
for (i=0; i<5; i++)
  {
  scanf("%d", &a[i]);
  }
printf("\n displayed array");
for (i=0; i<5; i++)
  {
printf("%d", a[i]);
  }
  }
```

→ Write the program to sum 5 natural no. using for loop.

```
#
#
main()
{
int i, sum=0;
clrscr();
for (i=1; i<5; i++)
 {
sum = sum+i;
 }
```

```
printf("sum %d", sum);
getch();
}
```

Matrix

```
#
#
main()
{
int a[2][2], b[2][2], c[2][2], i, j, k;
clrscr();
printf("\n Enter matrix\n");
for (i=0; i<2; i++)
  {
for (j=0; j<2; j++)
  {
scanf("%d", &a[i][j]);
  }}
scanf("%d", &a[i][j]);
for (i=0; i<2; i++)
  {
for (j=0; j<2; j++)
  {
c[i][j]=0;
c[i][j]=0;
for (k=0; k<2; k++)
  {
c[i][j]=c[i][j]+a[i][k] x b[k][j];
  }}}
```

```
printf ("\n ");
    }
getch();
  }

#
#
#include <string.h>
  main()
  {
  int i;
   char a[30]; char b[30];
   clrscr();
   printf ("\n Enter string")
   scanf ("%s", a);
   strcpy (b, a);
   i= strcmp (a, strrev (b));
   if (i==0)
   {
   printf ("\n string is palindrome");
     }
   else
   {
   printf ("\n string is not palindrome");
     }
   getch();
     }
```

function :-

```
main()
{
void disp();
disp();
}
void disp() // fun^n decl^n
{
printf("\n welcome to function");
                          //fun^n call
```

main()
void dis (int a, int b) // function definition

(formal / Dummy argument)

i) Call by value / pass by value
ii) Call by reference.

9/10/17

1) What is a macro, explain the macro
2) What do you mean by mail merge
3) W.A.P in C 1 to 20.

PAGE:
DATE: / /

Function

A function is a named unit of a sub-programme that performs the specific task (defined). A function is invoked (called) from any part of the programme and after execution of the task it will returned back to the calling portion of the programme.

Advantage of using function :—

i) A large complex programme may be divided (broken down) into smaller sub-programme known as function.

ii) Smaller sub programme is easy to write.

iii) A function can call any no. of time from any part of the programme.

Types of function :—

C function can be classified into two categories, namely, library function and user-defined function.

i) Built in function / library function
ii) User defined function

→ library function :— These functions are the part of the language compiler and this are available with the standard library file provided by the compiler. These type of function are frequently used in the programme to perform the specific task.

for e.g :— pow(), sqrt(), strlen() strrev() etc. are the e.g of Built-in or library function.

→ User defined function :— The user defined function are created (defined) by the user. These function are defined as per the need of programme.

Terms associated with function.

1) function Prototype - A function prototype of the function i.e., the information to the compiler about the type of value returned by the function and the number and type of arguments.

General Syntax - for a function prototype is as follows :-

Return type function, name (argument list);

ex :- i) void Sum (void);
ii) int factorial (void);
iii) int sum (int A, int B, int C);
iv) float avg (float, float);
v) void cal (int A, int B, char ()';

2) function- defination - A function defination contain the actual coding means the task to be executed whenever the function is called (Invoked).

The general syntax for a function defination is as follows :-

PAGE:
DATE: / /

→ return type function name
(argument list)
{
body of function
(task)
}

3) Global function Prototype :— when function prototype appear before the dipination of calling function then it is called Global function prototype.

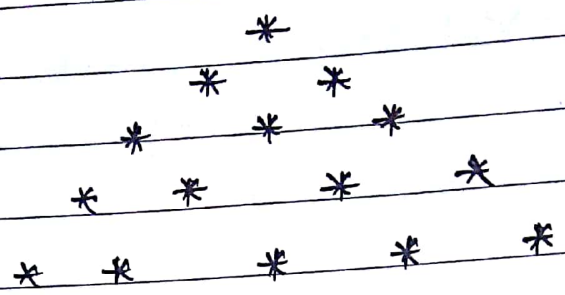
4) local Prototype :— when.

Write a program in c to check whether a number is prime or not :—

```
main()
{
int num, count=0, i;
clrscr();
printf("\n enter a no. for check:");
scanf("%d", &num);
for(i=1; i++; i<=num)
{
if (num % i==0)
{
count ++;
}
}
if (count ==2)
{
printf("No is prime");
}
else
{
printf("No is not prime");
}
getch();
}
```

DATE: / /

* Write a program to display the following pattern:—



```
#include <stdio.h>
#include <conio.h>
  main()
  {
  int i, j, K;
   clrscr();
  for (i=1; i<=5; i++)
  {
  for (K=5; K>=i; K--)
    printf(" ");
  for (j=1; j<=i; j++)
      {
    printf("*");
      }
    printf("\n");
        }
  getch();
        }
```

PAGE:
DATE: / /

* Write a program in C of fibonacci Series :-

```
#include <stdio.h>
#include <conio.h>
main()
 {
int a=0, b=1, c, i;
  clrscr();
  printf("\n %d", a);
  printf("\n %d", b);
  for (i=1; i<=10; i++)
    {
    c=a+b;
  printf("\n %d", c);
    a=b;
    b=c;
    }
  getch();
    }
```

PAGE:
DATE: / /

What is macro?

→ In microsoft office word 2017, you can automate frequently used tasks by creating macros.

A macro is a series of commands and instructions that your group together as a single command to accomplish a task automatically.

Steps to assign the macro to a keyboard shortcut, do the following :-

1) click Keyboard
2) In the commands box, click the macro that you are recording.
3) In the press new shortkut, key box, type the key. sequence that you want, and then click assign.
4) Click close to begin recording the macro.

Record macro — Keyword — ctrl + q.

PAGE:
DATE: / /

// call by value

```
#include <stdio.h>
main()
{
int a, b;
void swap (int, int);
printf("\n Enter the value of a :");
scanf("%d", &a);
printf("\n Enter the value of b:");
scanf("%d", &b);
swap (a, p);
printf("\n main () a = %d", a);
printf("\n main () b = %d", b);
}
void swap (int a, int b) // fun^n declaration
int t;
t = a;
a = b;
b = t;
printf("\n Now a = %d", a);
printf("\n Now b = %d", b);
```

// call by reference

```
#include <stdio.h>
main()
{
int a,b;
void swap (int * , int *);
                // function declaration
printf("\n Enter the value of a:");
scanf("%d", &a);
printf("\n Enter the value of b:");
scanf("%d",&b);
swap (&a,&b) // fun^ call.
printf("\n Main() a= %d", a);
printf("\n Main() b= %d", &b);
}
void swap (int *a , int *b) // fun^ definition
{
int t;
t = *a;
*a = *b;
*b = t;
printf("\n Now = a = %d", *a);
printf("\n Now = b = %d", *b);
}
```

PAGE:
DATE: / /

Recursive function

```
#include <stdio.h>
main()
{
int fact, num;
int factorial (int); // fun^n declaration
printf ("\n Enter the value of num:");
scanf ("%d", &num);
fact = factorial (num); // function
                           call.
printf ("\n factorial = %d", fact);
}
int factorial (int n) // function
                         definition
{
int (n==0)
return 1;
else
return (n * factorial (n-1));
}
```

→ W.A.P in C to reverse no. using recursive function.

```
→ #include <stdio.h>
  #include <math.h>
   int rev (int, int);
   int main()
   {
   int num, result;
   int length=0; temp;
```

```
printf("Enter an integer num to
        be reverse:");
scanf("%d", num);
  temp = num;
while (temp != 0)
{
length ++i
temp = temp/10;
      }
result = rev (num, length);
printf("The reverse of %d is
        %d.\n", num, result);
return 0;
      }
int rev (int num, int len)
  {
  if (len == 1)
    {
return num;
      }
  else
  {
return ((num % 10) * pow (10, len-1))
      + rev (num/10, --len));
```

31/10/17

PAGE:
DATE: / /

// Pointer : Keeps the address of another variable of its same type.

```
#include <stdio.h>
main()
{
int **ptr1;
int *ptr;
int a = 4;
ptr1 = &ptr;
ptr = &a;
printf("\n a = %d", a);
printf("\n a = %d", *ptr);
printf("\n a = %d", **ptr1);
}
```

1) // String

```
#include <stdio.h>
main()
{
char name[30];
printf("Enter String:");
scanf("%s", name);
printf("\n Your name = %s", name);
}
```

DATE: / /

2) // string

```
#include <stdio.h>
  main()
  {
  char name[30];
  int i;
  printf("enter string:");
  scanf("%s", name);
  for(i=0; name[i]!=NULL; i++)
  printf("%c" name[i]);
      }
```

// string

```
#include <stdio.h>
  main()
  {
  char name[30];
  int I, i, j, flag=0;
  printf("enter string:");
  scanf("%s", name);
  for(I=0; name[I]!=NULL; I++);
      I--;
    for(i=0; j=I; name[i]!=NULL;
            i++, j--)
    {
    if(name[i] != name[j])
      {
        flag=1;
        break;
      }}}
```

PAGE:
DATE: / /

Using String function

// String

#
#
main()
{
char name [30], name [30];
int l, i, j, flag = 0;
printf ("Enter String :");
scanf ("%s", name);
strcpy (name1, name);
i = strcmp (strrev (name1), name);
if (i == 0)
printf ("\n string is palindrome");
else
printf ("\n string is Not palindrome");
}

* Write a program in C to print armstorm no. between 100 to 1000.

* Write a program to check no. is armstrong or not;
$1^3 + 5^3 + 3^3 = 153$

#
#
main()
{
int num, i, sum = 0;

```
clrscr();
printf("Enter no");
scanf("%d", &num)
for (i=num; i>0; i=i/10)
 {
sum = sum+(i%10)*(i%10)*
        (i%10);
    }
if (sum==num)
 {
printf("No. is Armstrong")
    }
else
 {
printf("Not a Armstrong")
    }}
getch();
    }
```

OR

```
#include <stdio.h>
#include <conio.h>
main()
{
int num, rem, sum=0; num1;
clrscr();
printf("Enter no");
scanf("%d", &num);
num = num1;
while (num>0)
 {
rem= num%10;
```

```
Sum = sum + rem * rem * rem ;
num = num /10;
        }
if ( Sum == num1)
{
printf("\n No. is Aramstrong")
    }
else
{
printf("\n No. is not Aramstrong")
        }
        }
    getch ();
        }
```

for Series of Aramstrong nos:-

```
# include <stdio.h>
# include <conio.h>
# include <math.h>
  {
  int num, num1, rem, sum;
    clrscr ();
    for ( i= 100; i <= 1000 ; i++)
      {
      Sum= 0;
      num = i ;
      while (num>0)
      {
      rem = num% 10;
      Sum = Sum+ rem *rem * rem;
      num = num /10;
          }
```

```
if ( i == sum)
printf("\n %d", i);
}
getch();
}
```

~~Array in c~~

PAGE:
DATE: / /

Array in C.

Array is a collection of similar data. Array is a static homogeneous data structure that provides a way to store a multiple data of similar type in continous memory allocation pattern.

Type of Array :-

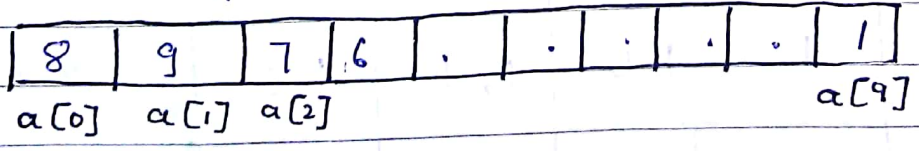
1) 1 D Array
2) 2 D Array (Matrices)
3) Multidimensional array.

① 1 D array :- In a one dimension array we have one dimension to store the data.

eg :- int a[10];

a is an integer array of size 10. so it can store 10 integer from starting Array index a[0] to a[9].

Memory allocation diagram



array index

a[0] = 8
a[1] = 9
. . . .

② 2D Array :- In a 2D array we have 2 dimensions to store the data. One is known as row and another is column. so it is also refer as matrix.

Ex :- int a[3][3];

a is an integer matrix of 3 row and 3 column. It can store 3 x 3 = 9 integer from starting matrix index a[0][0] to a[2][2]

Memory allocation of array diagram.

	0	1	2
0	0,0	0,1	0,2
1	1,0	1,1	1,2
2	2,0	2,1	2,2

Advantage of array :—

Using array we can store multiple data of similar type under a same name.

Disadvantage :—

1) By nature array is static (size is fixed)

2) Wastage of memory area.

3) Array can't store mixed data.

→ W.A.P in C to program to find duplicate value in an array.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int i, arr[20], j, no;
clrscr();
printf("enter size of array");
scanf("%d", &no);
printf("Enter any %d elements in
        array:", no);
for (i=0; j<no; i+1)
{
scanf("%d", &arr[i]);
}
```

1) → W.A.P in C to convert decimal to binary?

→ what is the difference between structure and union :-

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
main()
{
int num, rem, sum=0, i=0;
clrscr();
printf("\n Enter decimal number");
scanf("%d", &num);
while (num>0)
{
rem= num%2;
Sum= Sum + rem * pow (10, i);
num= num/2;
i++;
}
printf ("\n binary equivalent=%d", sum)
getch();
}
```

step 1:- declare a, b, c

step 2 :- Accept a and b

step 3:- compute $c = a+b$

Step 4 :- Print c.

while (n>0)
{
rem = n % 2
Sum = Sum * 10 + rem;
n = n/2
}

n	rem	Sum
6	0	0
3	1	1
1	1	11
0		

C:-

i	num	rem	Sum	Outp

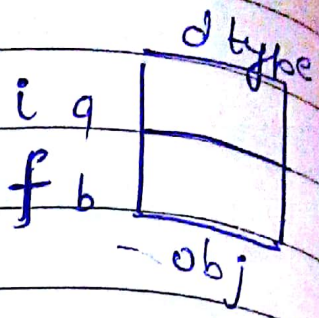
Binary to Decimal :—

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{
long int num, rem, sum=0, i=0;
clrscr();
printf("\n Enter binary number");
scanf("%d", &num);
while (num>0)
{
rem = num % 10;
sum = sum + rem * pow(2,i);
num = num / 10;
i++;
}
printf("\n decimal equivalent = %ld", sum);
getch();
}
```

PAGE:
DATE:

20/12/17

Structure :




```
#
#
struct dtype
{
int a;
float b;
};
main()
{
struct dtype obj;
clrscr();
printf("\n Enter integer value:");
scanf("%d", &obj.a);
printf("\n Enter float value:");
scanf("%f", &obj.b);
printf("\n Int displayed = %d", obj.a);
printf("\n float displayed = %f", obj.b);
getch();
}
```