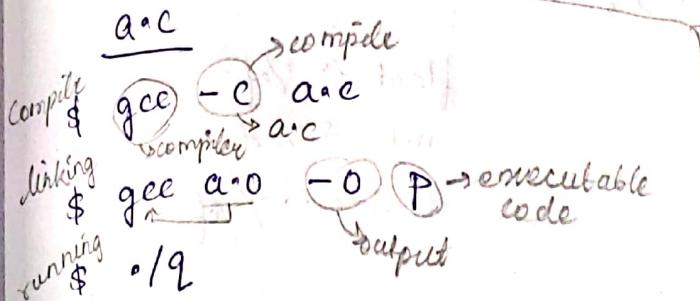
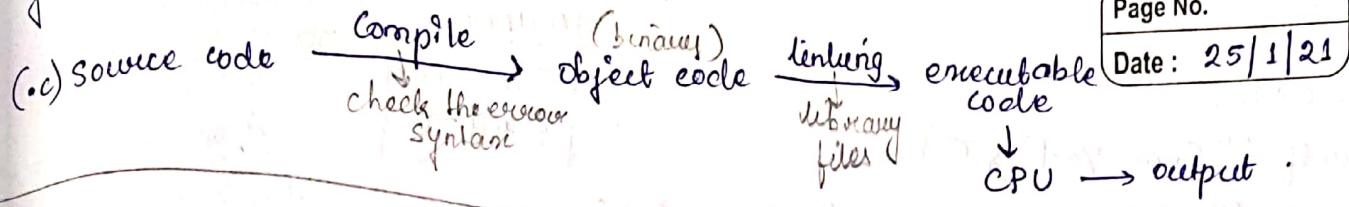


Program: Collection of statements written in a proper sequence which is given to a machine to perform a particular task.



gcc first.c
./a.out

a.out → default
p/q → explicitly created

→ \$ gcc p.c -o k
\$./k
\$ gcc q.c -o m
\$./m

a.c header file
#include <stdio.h> → preprocessor directive

I * this is my first program /* → multiple line comment

II this is a simple line → single line comment.

int main() { → main function

int a=2; → declaration as well as assignment.

int b;

b=a+4;

printf ("Simple program"); → function call.

printf ("Sum=%d", b); (%d will be replaced by b's value)

return 0;

statements

Preprocessor directive: It gives direction to the compiler to include or update the source code before the compilation starts.

printf → library file → detailed program of this function

printf → how it looks like → to check errors by Teacher's Signature

e.g. f(int, float); errors by the compiler.

```

→ int a = 2 ;
int b = 4 ;
int c ;
c = a + b ;
printf ("Sum %d", c) ;
printf ("%d + %d = %d", a, b, c) ;
(%d → format specifier)

```

Output	Page No.
Sum 6 2 + 4 = 6	Date :

float → %f
int → %d
double → %lf

```

float a = 2.5 ;
int b = 9 ;
printf ("%f %d", a, b) ;

```

→ Escape characters :

\n = new line char
\t = horizontally tab
\b = back space
\v = vertical tab
\r = carriage return
\f = form feed.

\n

```

printf ("ABC ");
printf ("DEF ");

```

Output
ABC DEF

```

printf ("ABC \n DEF");

```

Output
ABC DEF

\t

```

printf ("ABC \t DEF G \t KPL");

```

<u>Output</u>
ABC DEF G KPL

\v

```

printf ("ABCD\v pq");

```

→ bring the cursor to the beginning of the same line

ABCD -
↓
pq CD

<u>Output</u>
pq CD

```

(" \t ABCD \v pq");

```

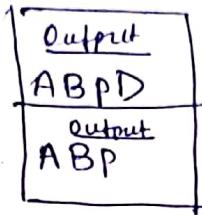
<u>Output</u>
pq ABCD

Teacher's Signature

\b printf ("ABCD\b\b p");

In gcc compiler :

AB CD
 \u2020



Page No.

Date :

In some other compilers :

AB PQ
 \u2020

\v printf ("ABCD\n PQ");

ABCD
 PQ

Output

A B C D E F
P Q \n

printf ("ABCDEF\n\n\b\b pq\n");

\n = \r + \n

printf ("ABC\r\nDEF");

ABC
DEF

\f

ABC\f Pq
↓
\n

ABC
 PQ

printf ("\n");

Output
 \n

Sum = %d

ABC" kp

printf ("Sum = % % d ");

printf ("ABC \"% kp");

Teacher's Signature

ASSIGNMENT-2 :

① Display your address in proper format

NAME :

AT :

POST :

PIN :

Page No.

Date :

② ABCD

EFGH

ABCD

MNOP

IJKL

XYZU

PQRS

PQRS

YZP

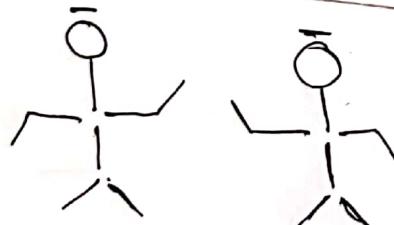
(3)



④

SL NO.	NAME	ROLL NO	CGPA
1.	XYZ	1245	96
2.	PQR	1563	96

⑤

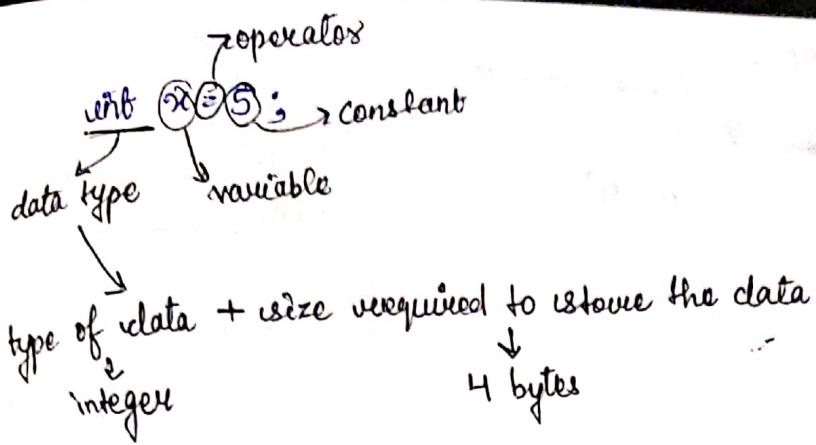


mkdir 2029023 → to create directory / folder

cd 2029023 → to enter into the folder

pwd

Teacher's Signature



Page No.

Date : 30/1/21

⇒ Types of constants :

→ Character = 'A', '2', '5', '\n'

→ integer = _{decimal} 25, _{octal} 025, _{hexadecimal} 0x25

→ real = 2.5, 6.25

→ string = "ABC", "A2BC", "235A"

(data type → type + size)

→ Character :

char → 1 byte

→ Integer :

should int → 2 bytes * To store 99 : size min req = 7 bits

int → 4 bytes

long → 8 bytes

long long → 16 bytes

$$6 \text{ bits} \rightarrow 2^6 = 64, \quad 7 \text{ bits} \rightarrow 2^7 = 128$$

→ Real :

float → 4 bytes

double → 8 bytes

long double → 16 bytes

* int a=5; → stored as 5

float b=5; → stored as 5.0.

→ Variable → a name given to a memory location where a value will be stored

Teacher's Signature

Naming Convention of variable name:

1. First letter : (A-Z), (a-z), ~~(-)~~ ^{underscore} (-)
2. Second letter : (A-Z), (a-z), (-), (0-9)
& onwards

Page No.

Date :

int qab ; X

int ab9 ; ✓

int a.b ; X

int a-b ; ✓

int ~~aab~~ ; ✓ ^(not preferable for application program)

3. int float ; X (\because float is a keyword)
 \hookrightarrow reserved words with specific meaning
Keywords shouldn't be used as a variable name.

4. Variable name is case sensitive
int a ; ~~int A ;~~ \rightarrow both are different.

$$* 6.25 = 110.01$$

$$6 = 110$$

$$0.25 \times 2 = 0.5 \rightarrow 0.1$$

$$0.5 \times 2 = 1.0$$

$$6.25 = 110.01$$

$$6.333$$

$$6 = 110$$

$$0.333 \times 2 = 0.666$$

$$0.666 \times 2 = 1.3322$$

$$0.332 \times 2 =$$

Teacher's Signature

char → %c

short int → %hd

int → %d

long → %ld

long long → %lld

float → %f

double → %lf

long double → %Lf

long → long int

long long → long long int

Page No.

Date :

→ float b = 7.5;
printf ("%f", b);

→ char p = 'B';
printf ("%c", p); B

→ short int k = 25;
printf ("%hd", k);

* char p = 'B'; → ASCII character → has corresponding ASCII value

↓
stored as
66 'B' → 66

char p = '@';
printf ("%c", p); @
printf ("%d", p); 25

int a = 65; / int a = 'A';
printf ("%d", a); 65

printf ("%c", a); A

printf ("%d %c", 'A', 'A'); 65 A

Teacher's Signature

→ char a = 2 ; → stores 2 00000010

char a = '2' ; → stores ASCII value of 2

Page No.

Date :

→ int a = 25 ;

printf ("%d", a) ; 125 → display in decimal

printf ("%o", a) ; → in octal 31

printf ("%x", a) ; → in hexadecimal 19

int a = 0296 ; → octal

printf ("%d", a) ; → display corresponding decimal of 296
hexadecimal
("%x", a) ; →

(%o → 1296 → octal

→ Signed, unsigned :

1. All real data types are always signed.

2. Character

char (1 byte) %c | unsigned char (1 byte) (%d)
(-128 to 127) | (0 - 255)
(2^7) ($2^7 - 1$) | ($2^8 - 1$)

3. Integer:

short int (%hd) | unsigned short int (%hu)

int (%d) | unsigned int (%u) (0 to $(2^{32} - 1)$)

long (%ld) | unsigned long (%lu)

long long (%lld) | unsigned long long (%llu)

(-2^{31}) to $(2^{31}-1)$

char a = 130 ;

-126

-128
 2^8

-127
 129

126
 130

unsigned char b = 130 ;

130

comes in range
0 to 255

Teacher's Signature

ASSIGNMENT-3

- ① Declare the variable of each data type and display the corresponding values.
- ② Display the ASCII values of the following :
 'A' → 'D', '2', '\n', '\', '\"', '\w' (all escape characters)
 'a' - 'd'
- ③ Display the ASCII character from 90 - 97 .
- ④ Take two variables x and y and compute $x^3 + 3x^2y - 2x + 3y + 4$.
int
float
- ⑤ Take three variables x , y and z and display .

x	z
y	y^2
z	z^2

- ⑥ Take two variables and interchange its values .

$$a=2, b=3$$

such that ~~use~~ use printf (" %d %d ", a, b); 3 2

- ⑦ x, y, z

$$\begin{array}{l} x=2, y=3, z=4 \\ \downarrow \\ x=4, y=2, z=3 \end{array}$$

temp
 $\text{temp} = x$
 $x = z$
 $z = y$
 $y = \text{temp}$

Example sheet

Page No.
Date :

Teacher's Signature

→ String Constant :

printf ("Name is %s", "xyz");

→ Constant of int type, by default is int

Constant of float type, by default is double.

int a = 25 ;
↳ int

long
long long
unsigned
↳ 25

float a = 2.5 ;
↳ float

float a = 2.5f ;
↳ float

float
2.5
long double
double.

First reads 2.5 as double (8 bytes), then truncates it to float size (4 bytes).

→ sizeof operator :

int p = sizeof(int);
printf ("%d", p); 14

→ Read data from Standard Input : (scanf function)

int main()

{
 int a, b, c;
 scanf ("%d", &a); input
 scanf ("%d", &b);
 c = a + b;
 printf ("%d", c);
 return 0;
}

scanf ("%d %d", &a, &b);

float b;
scanf ("%f", &b);
char d;
scanf ("%c", &d);

int a;
float b;
scanf ("%d %f", &a, &b);

Teacher's Signature

Page No.

Date: 1/2/2021

→ int %d %i - in scanf function.

%d → only decimal integer.

%i → can distinguish decimal, octal & hexadecimal integer.

Page No.

Date:

scanf ("%d", &a); ← 25 | ← 025
printf ("%d", a); → 25 | → 25

scanf ("%i", &a); ← 25 | ← 025 | ← 0x25
printf ("%d", a); → 25 | → 21 | → 37

→ Operations :

i) Arithmetic operators :

+ - * / %

int a = 5;

int b; $5/2 = 2.5$ takes only integer part

b = a/2;

printf ("%d", b); 2

int a = 5;

float b;

b = a/2;

printf ("%f", b); 2.0

→ making num/den real type

b = a/2.0;

2.5

int a = 5;

float b = a/2 - 3.0; $\frac{5}{2} - 3.0 = \boxed{-1.0}$

b = a/2.0 - 3.0;

$$= \frac{5}{2} - 3.0 \\ = \boxed{-0.5}$$

Teacher's Signature

char a = 25;

int b = a/2; 12

Page No.

Date:

int a = 5;

int b;

b = a % 2; 1
↓ remainder

b = -5 % 2; -1

$$-a \% b = -(\text{remainder})$$

$$-a \% -b = -(\text{rem})$$

$$a \% -b = +(\text{rem})$$

$$a \% b = +(\text{rem})$$

2) Assignment operator:

(=, +=, -=, *=, /=, %=)

a = 2 + 5; ① | a += 5; $\Rightarrow a = a + 5$

$$\begin{aligned} a + &= 5 - 3, \rightarrow \text{higher precedence} \\ a &= a + 5 - 3 \\ &= 2 + 5 - 3 = 4 \end{aligned}$$

3) Relational operator:

<, >, <=, >=, ==, !=

True $\rightarrow 1$
False $\rightarrow 0$

int a = 2;

int b = 5;

int c = a < b; 1
printf("%d", c);

int a = 2;

int b = 3;

int c;

c = b == a; 2
2↑ 2↑

c = b == a
↓ 0

printf("%d", c);

0

c = (b != a);

↑ T
1

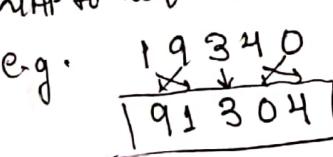
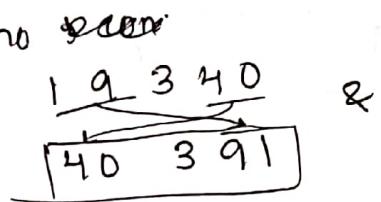
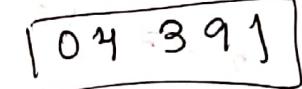
1

Teacher's Signature

ASSIGNMENT

Page No.

Date : 1/2/21

- ① Write a program that will take octal and hexadecimal value from keyboard and display its corresponding hexadecimal, octal and decimal. $\%x$, $\%o$, $\%d$
- ② Write a program to take the values from keyboard for each data type and display the corresponding values.
- ③ WAP to take 4 digit integer no (e.g 2396) and display each digit present in this integer.
- ④ WAP to take a 4 digit no and display the middle 3 digits.
- ⑤ WAP to take a 4 digit no and find its reverse.
(If $a = 23$, $b = 32$)
- ⑥ WAP to take a 4 digit no then display each digit in forward order.
- ⑦ WAP to take a 5 digit no ~~reverse~~
e.g.  &  & 

4) Logical operators :

&& || !
 ↓ ↓ ↓
 logical and logical or logical not

Page No.
Date : 2/2/21

int a=2, b=3;

int c = (a>2) && (b<3);

$$!(a \&\& b) \rightarrow !a || !b$$

$$!(a || b) \rightarrow !a \&\& !b$$

5) Increment and decrement operators : (++, --)

int a=2;
 a++; } $\Rightarrow a=a+1$;
 ++a;

int a=2;

int b;
 b=a++; \rightarrow post increment
 b=++a; \rightarrow pre increment

assign then increase
 $b=a++$; b 2 a 2 3
 increase then assign
 $b=++a$; a 3 b 3

6) sizeof operator

7) Ternary operator : (? :) (Conditional operator)

(Cond) ? statement 1 : statement 2

Teacher's Signature

int main()

{

 int a, b, c, d;

 scanf ("%d%d%d", &a, &b, &c);

 d = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

 printf ("the greatest number is %d", d);

 return 0;

}

Page No.

Date :

Bitwise Operator:

&	\sim	\ll	\gg
and	or	xor	not
		left shift	right shift
$a \Rightarrow 0110$	$00 \rightarrow 0$	$00 \rightarrow 6$	$00 \rightarrow 0$
$b \Rightarrow 0100$	$01 \rightarrow 0$	$01 \rightarrow 1$	$01 \rightarrow 1$
$a \& b \Rightarrow 0100$	$10 \rightarrow 0$	$10 \rightarrow 1$	$10 \rightarrow 1$
	$11 \rightarrow 1$	$11 \rightarrow 1$	$11 \rightarrow 0$

unsigned char a = 250; b; $\Rightarrow 00001111$

$b = a \& 15;$
 $\frac{15}{5} = 0x0F$

$b = 00001010$

250 $\rightarrow 11111010$

to get this part
149 $\rightarrow \begin{array}{r} 10010101 \\ 11110000 \end{array}$
 $10010000 \rightarrow c$

~~149~~ \rightarrow $c \gg 4$ 00001001
unsigned char a = 149;
unsigned char b, c;
 $b = a \& 0xF0;$
 $c = c \gg 4;$

Right shift \rightarrow that many zeros removed from right size & added to left side

$a = \underline{10110110}$
 $a \gg 3$

$\underline{00010110}$
 $\underline{10110110} a \ll 2$
 $\rightarrow \underline{11011000}$

Teacher's Signature

Scanned by CamScanner

ASSIGNMENT - 5

- ① Find the smallest among 2 nos.
- ② Find the smallest among 6 nos.
- ③ Take a 4 digit number and find its reverse and compare the original number with its reverse.
e.g. $1221 \rightarrow 1221$, if equal display 1, else display 0.
- ④ Find whether the 4 digit number is a valid number or not.
e.g. $\underline{2055} \rightarrow 05$ $25 = 5^2 \checkmark$, $3066 \checkmark$, $(6041)X$
- ⑤ Take a 4 digit number and check if it's valid as per the following condⁿ:
e.g. ~~2000~~ $\underline{\underline{1199}}$ $19 \rightarrow 19 \checkmark$
- Page No. _____
Date: _____
- $$\begin{array}{r} 1919 \\ \hline 91 \\ \downarrow \\ 19 \end{array}$$
 $91 \rightarrow \text{reverse of } 19 \checkmark$

student@kodak.org

Teacher's Signature

unsigned char a=234 ;

$$b = a \& 0x0F$$

$$b = b \ll 4;$$

$$c = (a \cdot \& 0xF0)$$

Page No.

Date : 6/2/21

* unsigned char a=10

00001010

$$a \ll 2$$

~~00000010~~

$$00101000 \Rightarrow 40 \Rightarrow 10 \times 2^2$$

$$a \ll 3$$

$$01010000 \Rightarrow 80 \Rightarrow 10 \times 2^3$$

* a=10

00001010

$$a \gg 2$$

$$000000010 \Rightarrow 2 \Rightarrow 10/2^2$$

$$a \gg 3$$

$$000000001 \Rightarrow 1 \Rightarrow 10/2^3$$

\therefore In \ll : If the number is n, after left shift it becomes
(s = no of left shift)

$$\boxed{n \times 2^s}$$

In \gg : if no is n, after s right shift it becomes

$$\boxed{n/2^s}$$

Comma operator : (,)

int x=3, y=4, z ; \downarrow left to right execution

$$x=3, y=4, z=x+y$$

Teacher's Signature

Scanned by CamScanner

char p = ('A', 'B', 'C');
printf ("%c", p); A

char p = ('A', 'B', 'C');
printf ("%c", p); C

Page No.

Date :

* Precedence and associativity:

associativity
left right

e.g. ~~order~~ int a = 2, b = 3;
right \rightarrow $a = b = 5$; left $\begin{cases} a = b \\ a = 3 \end{cases} = 5$;
 $a = 5, b = 5$

ASSIGNMENT-6

① Take an integer value and compute the second up value.

$$\begin{aligned} 1926 &\rightarrow 1930 \\ 1921 &\rightarrow 1920 \end{aligned}$$

② Take a real value and make a second off value.

$$\begin{aligned} 14.3468 &\rightarrow 14.35 \\ 14.3448 &\rightarrow 14.34 \end{aligned}$$

③ Find the second minimum among 5 nos.

5 ③ 2

④ Take an unsigned char and display the following:

i) each 4 bits ~~digit~~ bits ~~bits~~ bits with last 4 digits and generate the
ii) interchange the first 4 digits with last 4 digits and generate the
number

⑤ Take an integer and display the middle two bits

$$\begin{array}{r} 15 \quad 16 \quad 17 \quad 15 \\ \hline 32 \end{array}$$

⑥ Take 2 unsigned char

$$\begin{array}{l} a \rightarrow 0100101100 \\ b \rightarrow 11000010 \end{array} \rightarrow \begin{array}{l} 01001100 \\ 10110010 \end{array}$$

Teacher's Signature

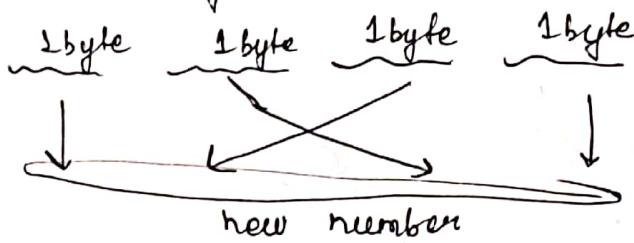
⑦ Take an integer and display each byte.

1 byte 1 byte 1 byte 1 byte $b = a \& \text{FF}00$
8 bits 00 FF 00 $b \gg 8$

Page No.

Date :

⑧ Take an integer and



⑧ int a = 754285456 ; 2C F5 76 90

int b, c, d, e ;
 ↓
 2C 76 F5 90

$b = a \& \text{0xFF0000FF}$;

$c = (a \& \text{0x00FF0000}) \gg 8$;

~~d = (a & 0x0000FF00) << 8~~ ;

2C 00 00 90 → b

00 00 F5 00 → c

00 76 00 00 → d

e = b | c | d ;

Teacher's Signature

C Operator Precedence Table

C operators are listed in order of *precedence* (highest to lowest). Their *associativity* indicates in what order operators of equal precedence in an expression are applied.

Operator	Description	Associativity
()	Parentheses: grouping or function call	left-to-right
[]	Brackets (array subscript)	
.	Member selection via object name	
->	Member selection via pointer	
++ --	Postfix increment/decrement	
++ --	Prefix increment/decrement	right-to-left
+ -	Unary plus/minus	
! ~	Logical negation/bitwise complement	
(<i>type</i>)	Cast (convert value to temporary value of <i>type</i>)	
*	Dereference	
&	Address (of operand)	
sizeof	Determine size in bytes on this implementation	
* / %	Multiplication/division/modulus	left-to-right
+ -	Addition/subtraction	left-to-right
<< >>	Bitwise shift left, Bitwise shift right	left-to-right
< <=	Relational less than/less than or equal to	left-to-right
> >=	Relational greater than/greater than or equal to	
== !=	Relational is equal to/is not equal to	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
	Logical OR	left-to-right
? :	Ternary conditional	right-to-left
=	Assignment	right-to-left
+= -=	Addition/subtraction assignment	
*= /=	Multiplication/division assignment	
%= &=	Modulus/bitwise AND assignment	
^= =	Bitwise exclusive/inclusive OR assignment	
<<= >>=	Bitwise shift left/right assignment	
,	Comma (separate expressions)	left-to-right

Decision Control Statement :

- if
- if else
- if else if

Page No.
Date : 8/2/21

ASSIGNMENT - 7

- ① Find the minimum among 2 nos.
- ② Display whether the no is +ve or -ve.
- ③ Find the minimum and second minimum among 3 nos.
- ④ Find the largest and smallest among 4 nos.
- ⑤ Enter two dates (dd/mm/yy)
Find which ~~date~~ date is the earliest one.
- ⑥ Take ~~two~~ two 24 hrs time and find the earliest one.
- ⑦ Take an integer and find out how many digits are present in this integer. (Input = 3 digit / 4 digit / 5 digit)
Find the middle digit.

ENGLISH THEORETICAL

Teacher's Signature

* $a \rightarrow 0110$

$b \rightarrow 1011$

$C = A \wedge B$

208(1)

$$\begin{array}{r} 0110 \\ 1011 \\ \hline 1101 \rightarrow C \end{array}$$

Page No.

Date:

Q. $C \wedge a = b \rightarrow$

$C \wedge b = a$

C	1	1	0	1
\wedge	a	0	1	1
b	1	0	1	1

* if...else if...:

if (exp 1)

{ }

else if (exp 2)

{ }

else if (exp 3)

{ }

else

{ }

} Optional

* Switch - case:

switch (expression)

{ case <value> :

case <value> :

default :

{ }

Constant value

↳ integer
character
constant expression

2+3

2/3 * 5

(2+a) X cannot use variable

www.camScanner.com

Teacher's Signature

```
→ int a;  
scanf ("%d", &a);  
switch (a)  
{  
    case 1 : printf ("MONDAY");  
    break;  
    case 2 : printf ("TUESDAY");  
    break;  
    case 3 : printf ("WEDNESDAY");  
    break;  
    default : printf ("invalid");  
}
```

Page No.

Date:

```
→ int mon-no, nd;  
scanf ("%d", &mon-no);  
switch (mon-no)  
{  
    case 1 :  
    case 3 :  
    case 5 :  
    case 7 :  
    case 8 :  
    case 10 :  
    case 12 : nd = 31;  
    break;  
    case 2 : nd = 28;  
    break;  
  
    case 4 :  
    case 6 :  
    case 9 :  
    case 11 : nd = 30;  
    break;  
    default : printf ("Wrong input");  
}
```

```
printf ("%d", nd);  
return 0;  
}
```

Teacher's Signature

ASSIGNMENT-8

- ① Take 3 items price → give the discount based on the following criteria:
- If $TA \geq 5000 \Rightarrow 30\% \text{ dis.} \rightarrow (\text{If any one item's price } > 3000, + 5\% \text{ discount})$
- If $TA \geq 3000 \Rightarrow 20\% \text{ dis.} \rightarrow (\text{one item } > 2000, + 5\% \text{ dis.})$
- $TA > 1000 \Rightarrow 10\% \text{ discount.}$

Page No.

Date: 9/2/21

Then display the amount the customer has to pay.

- ② Check the given year is a leap year or not.

$$\begin{array}{rcl} 2004 & 2000 & 1900 \\ \cancel{4} & \cancel{00} & \cancel{00} \\ \times & 100 & 100 \\ - & 400 & 00 \\ \hline & 400 & \end{array}$$

- ③ give the month and year as input and display the no of days in that month
(consider leap year).

- ④ Take a two digit number and display in words.

$$20 \leq \text{no} \leq 99$$

- ⑤ ~~Take two strings~~

Teacher's Signature

* Looping Statement:

It is a statement that repeatedly executes certain statements.

- while
- do...while
- for.

Page No.
Date: 15/2/21

while (expression)

{
 |||
}

ASSIGNMENT-9

- ① Display 1 to 10 using while loop.
- ② Display 24 to 5.
- ③ ~~Display~~ m to n ($m > n$) . Display $m, (m-2), (m-4) \dots \dots n$.
Exclude m, n.
- ④ Display all the nos which are divisible by 7. in between 1 to 200/n
- ⑤ Take a no and display its multiplication table.
- ⑥ Take a no and display all its digits
- ⑦ $1+2+3+\dots+10 = \underline{\quad}$
- ⑧ $1+3+5+7+\dots+n = \underline{\quad}$
- ⑨ $n * (n-1) * (n-2) \dots 1 = \underline{\quad}$
- ⑩ $n * (n-3) * (n-6) * \dots 1 = \underline{\quad}$
- ⑪ $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \underline{\quad}$
- ⑫ $\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots = \underline{\quad}$
- ⑬ $\frac{1}{1} + \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n-1}{n} = \underline{\quad}$

Subject: C Language

Teacher's Signature

ASSIGNMENT-10

Page No.

Date : 22/2/21

- ① Find the sum of numbers entered by the user.
& product
- ② Find the product of all even numbers among all the n numbers entered by the user
- ③ Let the user enter all the n odd numbers and display their summation.
- ④ Find the summation of all the numbers in between n and m which are divisible by 7 and 3.
- ⑤ Display all the integers whose square is in between n and m.
- ⑥ Find whether a number is prime or not -
- ⑦ $\frac{1}{0!} + \frac{2}{1!} + \frac{3}{2!} + \frac{4}{3!} + \dots + \frac{n}{(n-1)!}$
- ⑧ user will enter the day of 1st Jan 2000
Display : (if input is 4)

Sun	Mon	Tue	Wed	Thu	Fri	Sat
①	②	③	④	⑤	⑥	⑦
			1	2	3	4
5	6	7	8	9	10	
-	-	-	-	-	-	-
-	-	-	-	-	-	-

31 Jan - 31 31 August - 243

28 Feb - 59 30 Sept - 273

31 March - 40 31 Oct - 304

30 April - 120 30 Nov - 334

31 May - 151 31 Dec - 365

30 June - 181

31 July - 222

Teacher's Signature

For loop :

```

int a ;           Initialization
for ( a = 1 ; a <= 5 ; a ++ )    Condition
{   printf ("%d", a) ;
}

```

Page No.

Date :

ASSIGNMENT - 11

- ① Display 1 to n & n to 1.
- ② Display all the numbers which are divisible by 3 and 12. in between n & m
- ③ 4 3 2 1 1 2 3 4 → Display
& $n(n-1)(n-2) \dots 1 2 3 \dots n$
- ④ Fibonacci series : 1 1 2 3 5 8 13
- ⑤ $\frac{1}{3} + \frac{5}{7} + \frac{9}{11} + \frac{13}{15} + \dots + \frac{n-2}{n} = \dots$
- ⑥ 1, 10, 2, 9, 3, 8, 4, 7, 5, 6
- ⑦ $\frac{1}{2!} + \frac{2}{4!} + \frac{3}{6!} + \frac{4}{8!} + \dots$
- ⑧ You need to display the middle digit of any number.

$$\begin{array}{ccccc}
\underline{12} & \underline{1} & \underline{135} & \underline{1962} & \underline{\begin{matrix} 296472 \\ \downarrow \\ 64 \end{matrix}}
\\ \downarrow & \downarrow & \downarrow & \downarrow &
\end{array}$$

Teacher's Signature

ASSIGNMENT-12

- (1) *
 * *
 * * *
 * * * *
- (2) * * * *
 * * *
 * *
 *
- (3) 0 0 0 *
 0 0 * *
 0 * * *
 * * * *
- Page No.
 Date : 1/3/21
-
- (4) 1
 2 1
 3 2 1
 4 3 2 1
- (5) 0 0 0 1
 0 0 1 0 1
 0 1 0 1 0 1
 1 0 1 0 1 0 1
- (6) 1 1 2
 1 2 3
 1 2 3 4
-
- (7) 1
 2 2
 3 3 3
 4 4 4 4
- (8) 1 2
 1 1 1
 1 2 3 4
 1 1 1 1 1
 1 2 3 4 5 6
- (9) * *
 * *
 * *
 * *
 * * * * *
-
- (10) 0 0 1 0 0
 0 1 0 1 0
 1 0 1 0 1
 0 1 0 1 0
 0 0 1 0 0
- (11) 1 2 3 4 5
 6 7 8 9
 10 11 12
 13 14
 15
- (12) 1 2 3 4 5
 9 8 7 6
 10 11 12
 14 13
 15
-
- (13) 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
- (14)
- | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | * |
| * | 2 | 3 | 4 | 5 | 6 | 7 | * | * |
| * | 0 | 1 | 2 | 3 | 4 | 5 | * | * |
| * | 0 | 0 | 1 | 2 | 3 | 2 | 2 | * |
| * | * | * | * | * | * | * | * | * |

Teacher's Signature

ASSIGNMENT-13

2

Page No.

Date : 2/3/21

n

A grid of handwritten Chinese characters arranged in four rows and five columns. The characters are written in black ink on a light background. A circled number '3' is located in the top-left corner of the grid.

4

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

$$n=5, \quad \cdot \frac{n \times (n+1)}{2}$$

(5)	15	14	13	12	11
	9	8	9	10	
	6	5	4		
	2	3			

$$\textcircled{6} \quad \text{aaaaa}^n = \frac{n!}{r!(n-r)!}$$

(2)	1	1	1	1	1	1	1	1	1
0	1	0	2	0	1	0	2	0	1
0	1	3	3	0	1	0	4	0	1
0	1	4	6	0	4	0	5	0	1
1	0	5	10	10	0	5	1	0	1

$$\begin{array}{c}
 \text{?} \\
 \text{Co}^0 \\
 \xrightarrow{\quad} (\text{Co}^1) \quad (\text{C}^1) \\
 (\text{Co}^2) \quad (\text{C}^2) \quad (\text{C}^2) \quad (\text{C}^3) \\
 (\text{Co}^3) \quad (\text{C}^3) \quad (\text{C}^3) \quad (\text{C}^3)
 \end{array}$$

$$C_j = \frac{z_1}{z_1(z-j)}$$

⑧ take an odd number from well.

~~10000~~ ~~400~~ ~~400~~ ~~4000~~ ~~4000~~

e.g. $n = 7$

$$\frac{(n+1)}{2} - 2$$

6
2

e

Teacher's Signature

ARRAY :

It is a collection of same type of elements stored in a continuous memory location represented by a single name.

Page No.

Date : 8/3/21

Mark $\begin{array}{|c|c|c|c|c|c|c|} \hline & = & = & = & = & = & = \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline \end{array} \leftarrow \text{int Mark[7]}$

int $a[5] = \{9, 6, 2, 4, 1\}$; $a \begin{array}{|c|c|c|c|c|} \hline 9 & 6 & 2 & 4 & 1 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$

int $a[5] = \{9, 6\}$; $a \begin{array}{|c|c|c|c|c|} \hline 9 & 6 & 0 & 0 & 0 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$

int $a[5] = \{0\}$; $a \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$

int $a[5]$;

$a = \{2, 9, 6, 2, 7\}$ \times

\rightarrow 100 array $\rightarrow 0, 0, 0, 0 \longrightarrow 9, 6$

int $a[100] = \{0\}$;

$a[98] = 9$; $a \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \leftarrow \begin{array}{|c|c|} \hline 9 & 6 \\ \hline \end{array} \longrightarrow 98 \ 99$

$a[99] = 6$;

\rightarrow int $a[] = \{5, 4, 7\}$; - create array of size 3.

$a \begin{array}{|c|c|c|} \hline 5 & 4 & 7 \\ \hline 0 & 1 & 2 \\ \hline \end{array}$

\rightarrow int $a[]$; \times ~~size~~

\rightarrow size of array is fixed.

\rightarrow Variable size array:

```
int no;
scanf ("%d", &no);
int a[no];
```

Teacher's Signature

$\rightarrow \text{int } no = 6;$
 $\text{int } a[\underline{no}] = \{ 5, 9, 6, 2, 2, 4 \}; \quad X$

Page No.

Date :

$$\rightarrow \hat{i} = 1 ;$$

$$a[i] = a[i-1] + 3 ;$$

$$a[i++] = 7 ;$$

→ int b ; | int a [5] ;
 int c = sizeof (b) ; | with c = sizeof (a) ; → 20
 int c = sizeof (int) ; | int d = sizeof (a [1]) ; → 4

 int no = sizeof (a) / sizeof (a [0]) ;
 5 ← 20 / 4
 (no of elements)

Assignment - 14

① Take a 6 size array. Fill the array

Teacher's Signature

①	A	<table border="1"> <tr><td>2</td><td>9</td><td>6</td><td>4</td><td>5</td><td>7</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	2	9	6	4	5	7	0	1	2	3	4	5	→	A	<table border="1"> <tr><td>7</td><td>4</td><td>3</td><td>2</td><td>6</td><td>8</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	7	4	3	2	6	8	0	1	2	3	4	5
2	9	6	4	5	7																								
0	1	2	3	4	5																								
7	4	3	2	6	8																								
0	1	2	3	4	5																								
	B	<table border="1"> <tr><td>2</td><td>6</td><td>8</td><td>7</td><td>4</td><td>3</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	2	6	8	7	4	3	0	1	2	3	4	5		B	<table border="1"> <tr><td>4</td><td>5</td><td>7</td><td>2</td><td>9</td><td>6</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	4	5	7	2	9	6	0	1	2	3	4	5
2	6	8	7	4	3																								
0	1	2	3	4	5																								
4	5	7	2	9	6																								
0	1	2	3	4	5																								

$A[i] \leftrightarrow A[i+3]$

$A[i] \leftrightarrow B[i+3]$

$B[i] \leftrightarrow A[i+3]$

for ($i=0$; $i < no/2$; $i++$)

{

$c = A[i];$

$A[i] = B[i + (no+1)/2];$

$B[i + (no+1)/2] = c;$

$c = B[i];$

$B[i] = A[i + (no+1)/2];$

$A[i + (no+1)/2] = c;$

②	A	<table border="1"> <tr><td>5</td><td>9</td><td>6</td><td>8</td><td>2</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> </table>	5	9	6	8	2	0	1	2	3	4	→	A	<table border="1"> <tr><td>2</td><td>8</td><td>6</td><td>9</td><td>5</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>-4</td></tr> </table>	2	8	6	9	5	0	1	2	3	-4
5	9	6	8	2																					
0	1	2	3	4																					
2	8	6	9	5																					
0	1	2	3	-4																					

int a[5] = {5, 9, 6, 8, 2}

Teacher's Signature

ASSIGNMENT - 15

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

Page No.

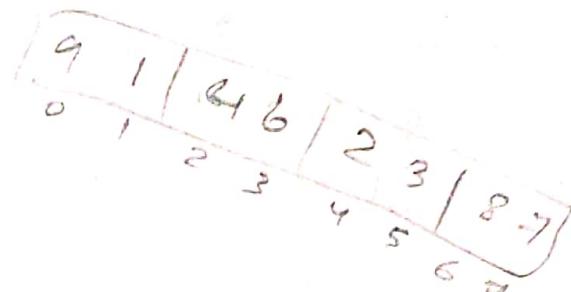
Date : 9/3/21

① Reverse the array.

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

(even sized array)

- 0 → 3
- 1 → 2
- 2 → 1
- 3 → 0
- 4 → 7
- 5 → 6
- 6 → 5
- 7 → 4



0 1 2 3 4 5 6 7

67 → 01
45 → 23

0 1 2 3 4 5 6 7

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

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

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

2	11	17	21	22	25	32
0	1	2	3	4	5	6

2+9 11+6 17+4 - -

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

0	2	7	-1	5	-4	7
4-5	1-(-4)	3-7				

⑥ Multiple of 4 sized array :

2	9	6	8	7	4	2	1	7	3	4	12
0	1	2	3	4	5	6	7	8	9	10	11

4	2	7	3	9	1	2	4	6	8	2	9
0	1	2	3	4	5	6	7	8	9	10	11

Teacher's Signature

(7)

2	9	6	8	4	7
---	---	---	---	---	---

9 2

9 6 2

9 6 8 2

9 6 8 4 2

9	6	8	4	7	2
---	---	---	---	---	---

Page No.

Date :

Teacher's Signature

(2-D ARRAY)

ASSIGNMENT-16

① Create a 2D array. Insert the values through scanf and display all the values.

Page No.

Date : 22/3/21

② Create an array, and fill it like :

1	4	7	10
2	5	8	11
3	6	9	12

③ a) Find out how many odd nos and even nos are present in the array.

b) Also, calculate the sum of all odd nos and all the even nos.

④

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

④ Find the sum of all the values present in each row and each column.

⑤ and find the greatest among all the summations.

⑤

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

→ Display.

⑥

1 ₀₀	2 ₀₁	4 ₀₂	7 ₀₃
3 ₁₀	5 ₁₁	8 ₁₂	6 ₁₃
6 ₂₀	9 ₂₁	5 ₂₂	3 ₂₃
10 ₃₀	4 ₃₁	2 ₃₂	1 ₃₃

Fill and display

⑦

0	1	8	10
7	2	6	9
8	6	3	5
10	9	5	4

7 ₀₀	8 ₀₁	1 ₀₂	9 ₀₃
8 ₁₀	6 ₁₁	2 ₁₂	7 ₁₃
1 ₂₀	9 ₂₁	5 ₂₂	3 ₂₃
9 ₃₀	7 ₃₁	8 ₃₂	4 ₃₃

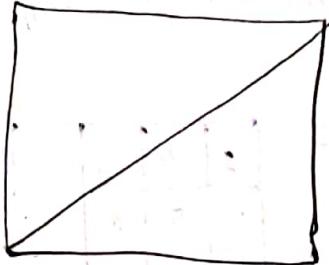
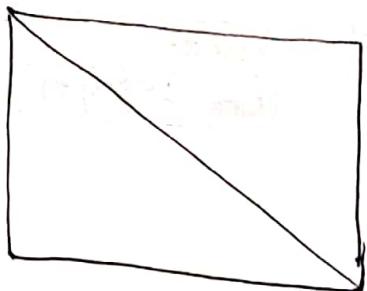
5 ₀₀	6 ₀₁	3 ₀₂	7 ₀₃
6 ₁₀	5 ₁₁	3 ₁₂	8 ₁₃
3 ₂₀	7 ₂₁	6 ₂₂	4 ₂₃
7 ₃₀	8 ₃₁	2 ₃₂	5 ₃₃

Teacher's Signature

ASSIGNMENT - 17

Page No.

Date : 23/3/21



add
square
matrix

- ② Multiply a no into a 2D matrix

③

$$A$$

$$C = 4A$$

Modify ② only : Take 2 matrices A and B

Compute $C = 3A + 4B - 2A$

④

$*_{00}$	$*_{01}$	$*_{02}$	$*_{03}$	$*_{04}$
$*_{10}$	11	12	13	$*_{11}$
$*_{20}$	21	22	23	$*_{21}$
$*_{30}$	$*_{31}$	$*_{32}$	$*_{33}$	$*_{34}$

$m = 4, n = 5$

Take $m * n$ and display only elements.

$0 \leq i \leq m-1$

$0 \leq j \leq n-1$

$(1 \leq i \leq m-1) \wedge (1 \leq j \leq n-2)$

$(m=1) \wedge (j \leq n-2)$

⑤

1	0	1	0	0
2	0	0	0	0
2	0	1	1	1
1	0	0	1	0
1	2	1	2	1

Find a position where a no is surrounded by opposite nos.

if $j=0$

→ Square matrix.



1 ₀₀	2 ₀₁	3 ₀₂	4 ₀₃
5 ₁₀	6 ₁₁	7 ₁₂	8 ₁₃
9 ₂₀	10 ₂₁	11 ₂₂	12 ₂₃

Teacher's Signature

* TYPE CONVERSION :

Page No.

Date : 27/3/21

char \leftrightarrow short

↓

int

↓

unsigned int

↓

long int

↓

unsigned long int

↓

short

↓

double

↓

long double

float a = 5/2 ; \Rightarrow 2.0

float a = (float) 5/2 ; $\boxed{a \Rightarrow 2.5}$

: TYPE CASTING OPERATOR

char a = 2 ;

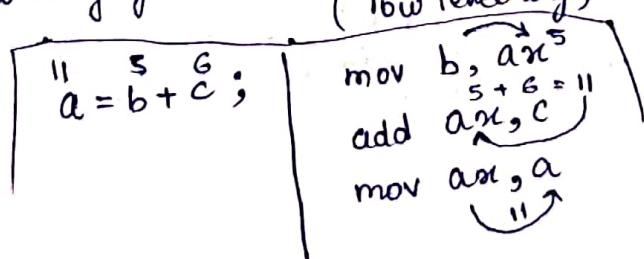
short int b = 2 ;

printf ("%d", sizeof (a+b)) ;

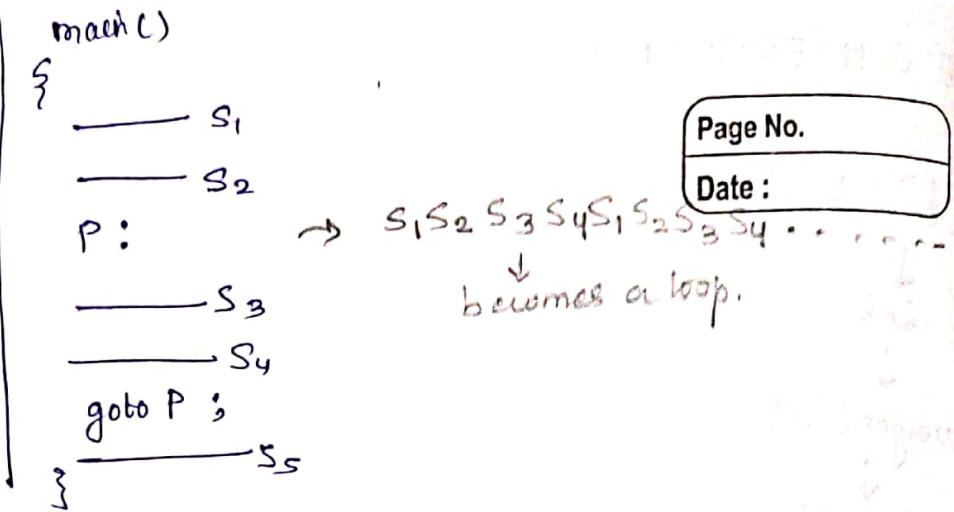
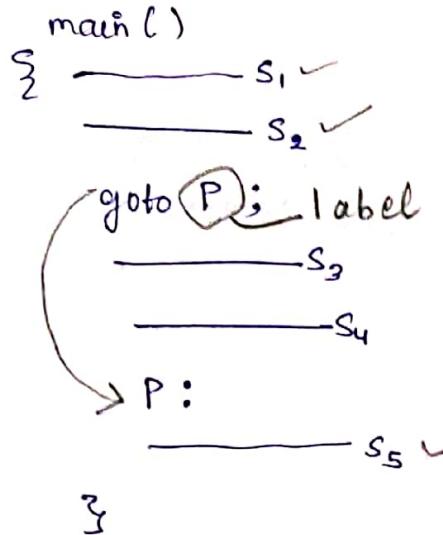
int

* Unconditional branching statement : (goto statement)

high level language \rightarrow Assembly language \rightarrow Binary language. \rightarrow CPU



Teacher's Signature



```

int a=1;
while (a<=5)
{
    printf("%d", a);
    a++;
}

```

main()

```

    {
        int a=1;
        P:
        printf("%d", a);
        a++;
        if (a<=5)
            goto P;
    }

```

```

for (i=0; i<3; i++)
{
    for (j=0; j<4; j++)
        printf("%d", a[i][j]);
}

```

int i, j;

```

    i=j=0;
    q:           j=0;
    P:
    printf("%d", a[i][j]);
    j++;
    if (j<4)
        goto P;
    i++;
    goto q;
}

```

Teacher's Signature

Matrix multiplication problem :

Page No.

Date :

ASSIGNMENT :

27/3/21

- ① Addition of two matrix
- ② Multiplication of two matrices
- ③ Display the following series using goto :

i) 10 9 8 7 ... -1

ii) 2 $\frac{4}{+2}$ $\frac{8}{+4}$ $\frac{14}{+6}$ $\frac{22}{+8}$ $\frac{32}{+10}$... n

- ④ Display the 2D matrix using goto in the following order.

①	④	⑦	⑩
②	⑤	⑧	⑪
③	⑥	⑨	⑫

- ⑤ Find the sum of all the digits present in a given no using goto

$$12537 \quad 1+2+5+3+7 = 18$$

- ⑥ Display multiplication table using goto

⑦

1	0	0	0	0	0
1	0	1	1	1	0
1	1	1	0	1	1
0	0	0	0	0	1
0	0	0	0	0	1
1	1	0	1	1	1
0	0	0	0	0	0

Teacher's Signature

Matrix multiplication problem :

Page No.

Date :

ASSIGNMENT : 18 27/3/21

- ① Addition of two matrix
- ② Multiplication of two matrices
- ③ Display the following series using goto :

i) $10 \ 9 \ 8 \ 7 \dots -1$
ii) $2 \frac{4}{+2} \frac{8}{+4} \frac{14}{+6} \frac{22}{+8} \frac{32}{+10} \dots n$

- ④ Display the 2D matrix using goto in the following order.

①	④	⑦	⑩
②	⑤	⑧	⑪
③	⑥	⑨	⑫

- ⑤ Find the sum of all the digits present in a given no using goto
 $1 + 2 + 5 + 3 + 7 = 18$

- ⑥ Display multiplication table using goto.

⑦

1	0	0	0	0	0
1	0	1	2	3	4
1	1	1	0	1	1
0	0	0	0	0	1
0	0	0	0	0	1
1	1	0	1	1	1
0	0	0	0	0	0

Teacher's Signature

String handling :

Collection of characters ended with null character.

"ABCD" → ABCD\0

char P; [A]

char a[4]; [ABCD]

ASCII value of
'\0' = 0

Page No.

Date : 30/3/21

Every string is stored through character array.

char a[50];

✓ char a[50] = "ABCD"; [A|B|C|D|\0|\0|\0|\0|\0]
0 1 2 3 4 5 6 7 8 9

✓ char a[50] = { 'A', 'B', 'C', 'D' }; [A|B|C|D|\0|\0|\0|\0|\0]
0 1 2 3 4 5 6 7 8 9

char a[10] = { 65, 66, 67, 68 }; [65|66|67|68|0|0|0|0|0|0]
Internally

char a[4] = "ABCD"; [A|B|C|D|\0]

char a[4] = { 'A', 'B', 'C', 'D' }; [A|B|C|D]

✓ char a[4] = { 'A', 'B', 'C', 'D', '\0' } [A|B|C|D|\0]

char a[5] = { 65, 66, 67, 68, 93, 0 } [A|B|C|D|?|1|\0]

char a[10];

scanf("%s", a);
↓
No &

Input
A B C D

storage

A B C D | 0

garbage

printf("%s", a); → A B C D

char a[10] = "AB PQRS \0"

printf("%s", a); [AB PQRS \0]

[A|B|P|Q|R|S|\0]

char a[10];
scanf("%s", a);

Input
A B P Q R S
storage [A B | 0]

printf("%s", a);

Output
AB

Teacher's Signature
scanf consider upto any
space or \n
first space

```

char a[10];
scanf ("%[^\\n]s", a);
    Input ABC pqrs
    Storage ABC pqrs
    (will take input until \\n comes)
    Output ABC pqrs
printf ("%s", a);
    
```

Page No.
Date:

```

char a[20];
scanf ("%[^\\n]s", a);
    abc, de fkp
    Input abcde fkp
    Storage a b c d e f k p | 1 2 3 4 5
    abcde fkp
    Output
    a[5] = '\0';
    printf ("%s", a);
    abcde
    → points till \0
    → data is still present
    but can't be displayed
    since \0 came first
    
```

```

char a[10];
a = "abcd"; X
a[0] = 'a';
a[1] = 'b'; ✓
a[2] = 'c';
a[3] = 'd';
    
```

```

char a[20];
scanf ("%[\\n\\n]s", a);
    abc kp
    Input abc kp
    Storage abc kp | 1 2 3 4 5
    abc kp
    Output
    for (i=0; a[i] != '\0'; i++)
        printf ("%c", a[i]);
    
```

Teacher's Signature

Length of array

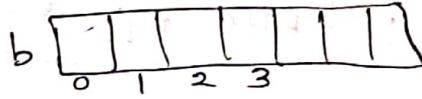
```
for (i=0; a[i] != '\0'; i++);
printf ("%c\n", i); 4
```

a	b	c	d	\0
0	1	2	3	4

Page No. _____
Date : _____

Copy strings

* a n y z \0 | | | |



→ to copy the content of a to b.

```
char a[20], b[20];
int i;
scanf ("%[^\\n]", a);
for (i=0; a[i] != '\0'; i++)
    b[i] = a[i];
b[i] = '\0';
```

```
int i=-1;
do {
    i++;
    b[i] = a[i];
} while (a[i] != '\0');
```

String Concatenation :



char a[20], b[10];

```
int i, j;
scanf ("%s", a);
scanf ("%s", b);
for (i=0; a[i] != '\0'; i++);
for (j=0; b[j] != '\0'; j++, i++)
    a[i] = b[j];
a[i] = '\0';
```

Teacher's Signature

ASSIGNMENT-189

- ① Receive a string that includes space using scanf and display it.
- ② find the length of the string
- ③ take a string of 3 words through scanf and display the middle word and its size.

Page No.

Date : 30/3/21

- ④ Find the number of words present in a string
- ⑤ Find the reverse of a string. (reverse the same array)
- ⑥ Find whether the given string is equal to its reverse or not.

⑦ A

A	B	C	D	P	\0
---	---	---	---	---	----

 → Copy in this manner.
B

P	D	C	B	A	\0
---	---	---	---	---	----

~~1 0 1 0 1 0~~

⑧ a

A	B	C	I	P	Q	R	S	T	U	\0
---	---	---	---	---	---	---	---	---	---	----

b

C	B	A	Q	P	U	T	S	R	\0
---	---	---	---	---	---	---	---	---	----

0 1 2 3 4 5 6

1	2	3	4
---	---	---	---

① 1 2 3

Assignment sheet no. 1

Teacher's Signature

POINTERS

Page No.

Date : 5/4/21

→ Pointer is a variable which can store the address of a memory location / it can store the address of a variable of the same type.

→ `int *P;` → pointer variable of integer type.

`int a;`
 $a = 5;$
 $P = \&a;$

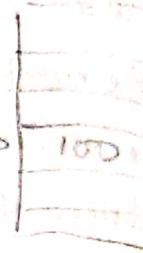
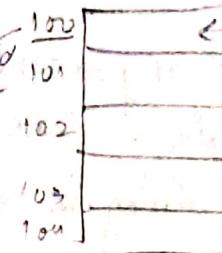
(P points to the value of a)

$\&P \Rightarrow 250 \rightarrow \text{address}$

$P \Rightarrow 100 \rightarrow \text{value}$

$\&a \Rightarrow 100$
 $a \Rightarrow 5$

address Memory location



$*(*(\&P)) \Rightarrow *(*250) \Rightarrow *100 \Rightarrow 5$

$*(\text{address}) \Rightarrow \text{Value presenting that address.}$

~~$*P \Rightarrow *(\&a) \Rightarrow *100 \Rightarrow 5$~~
 $\Rightarrow *P = 5$

$*(&P) \Rightarrow *250 \Rightarrow 100$

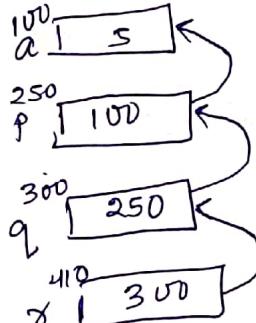
$\Rightarrow *250 \Rightarrow 100 \Rightarrow P$
So, $*\&P = P$

`int a = 5;`

`int *P = &a;`

~~Pointers~~
~~int * *q = &P;~~
~~Pointer~~
~~int pointer type~~

`int **x = &q;`



$a \rightarrow 5$

$\&a \rightarrow 100$

$P \rightarrow 100$

$*P \rightarrow 5$

$\&P \rightarrow 250$

$*\&P \rightarrow 100$

$q \rightarrow 250$

$*q \rightarrow *(250) \rightarrow 100$

$* *q \rightarrow *(*q) \rightarrow *100 \rightarrow 5$

$* * *q \rightarrow * *x \rightarrow *(*x) \rightarrow *(*(*x)) \rightarrow *(*(*(*x)))$

$\rightarrow *250 \rightarrow 100$

Teacher's Signature

int a[5];

9	6	2	7	4
0	1	2	3	4
100	104	108	112	116

Every array name represents the address of the first element of array

Page No.
Date:

printf ("%p", &a); → displays the address of a

printf ("%d", *P); → displays the value of pointer P

printf ("%p", &a[0]); → address of first element of array a

$$a+1 = 5$$

$$\&a+1 = 104 \quad (a+1 \times \text{size of data type})$$

int a[5];

a	9	3	7	8	6
0	1	2	3	4	5
100	104	108	112	116	120

$$a[0] \rightarrow 9$$

$$a \rightarrow 100$$

$$a+0 \rightarrow 100$$

$$*(a) \rightarrow 9$$

$$*(a+0) \rightarrow 9 \rightarrow a[0]$$

$$a+1 \rightarrow 104$$

$$*(a+1) \rightarrow 3 \rightarrow a[1]$$

$$a+2 \rightarrow 112$$

$$*(a+2) \rightarrow 8 \rightarrow a[2]$$

$$a[2][3] \rightarrow *(a[2]+3) \rightarrow *(*(a+2)+3)$$

$$\boxed{a[i] = *(a+i)}$$
$$= *((\&a[0] + i))$$

int b = 5;

int *P = &b;

printf ("%d", *P); → 5

printf ("%d", P[0]);

Every array internally is a pointer

Teacher's Signature

```

int *p = a;
for(i=0; i<5; i++)
    printf ("%d", p[i]);
    *(p+i)

```

Page No.

Date :

```

int *p = a;
for(i=0; i<5; ++i)
{
    printf ("%d", p[0]);
    p = p+1;
}

```

a | 9 | 8 | 7 | 8 | 6

Output
| 9 | 9 | 9 | 9 |

but

a[0]

a=a+1; X

}

5/4/21

ASSIGNMENT - 20

- ① Take a float pointer which will point to a float variable and display the value using float pointer.

```

float b = 3.9;
float *p;

```

- ② int a=10;
int *p;
int **q;

a, &a, P, &P, *P, *&P,
q, &q, *q, **q, **q, **&q

- ③ a | 9 | 5 | 7 | 2 | 6 | 8 | 4
Display the array using pointers in forward order and reverse order

using those pointers you swap the two values.

- ④ Take variables and two pointers are pointing to those two variables using those pointers you swap the two values.

⑤ Take an even size array. Take a pointer p & a pointer q.
Interchange the values as follows:

- 5 | 9 | 6 | 8 | 4 | 3 | 2 | 7

→ 4 | 3 | 2 | 7 | 5 | 9 | 6 | 8
Teacher's Signature

P → 9 | 3 | 2 | 7

ASSIGNMENT-2

① Find the max and second max from an integer array using pointers.

②

A	B	C	D	E	\n
---	---	---	---	---	----

 using pointer print the following patterns:

a) A B C D E
A B C D
A B C
A B
A

b) E
D E
C D E
B C D E
A B C D E

c) E D C B A
D C B A
C B A
B A
A

③ Take 20 size character array and fill 4 strings into same array

a)

A	B	C	\n	0		P	q	\n	0		P	T	L	B	\n	0	m	n	\n	0
---	---	---	----	---	--	---	---	----	---	--	---	---	---	---	----	---	---	---	----	---

b)

A	B	C	P	q		P	T	L	B	m	n	\n	0
---	---	---	---	---	--	---	---	---	---	---	---	----	---

④

A	B	C	\n	0		P	q	\n	0		P	T	L	B	\n	0	m	n	\n	0
---	---	---	----	---	--	---	---	----	---	--	---	---	---	---	----	---	---	---	----	---



A	B	C	P	q		P	T	L	B	m	n	\n	0
---	---	---	---	---	--	---	---	---	---	---	---	----	---

C	B	A	P	q		P	T	L	B	m	n	\n	0
---	---	---	---	---	--	---	---	---	---	---	---	----	---

Page No.	
Date:	6/4/21

Teacher's Signature

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

int *p = a; (P = 100)

printf ("%d", *p); → [2]

printf ("%d", *p++); → [2] then P becomes 104
Output so *P → [9]

printf ("%d", *++p) (first P becomes 104) → [9] output

printf ("%d", ++*p) → [3] output

↓
++ (*p)
next.

*p → a[0] = 2

++ *p → a[0] = 3

Page No.

Date: 12/4/21

precedence order:
++ > *

FUNCTION

→ A function is a series of statements combined together to perform a certain task.

return type

int add (int x, int y)

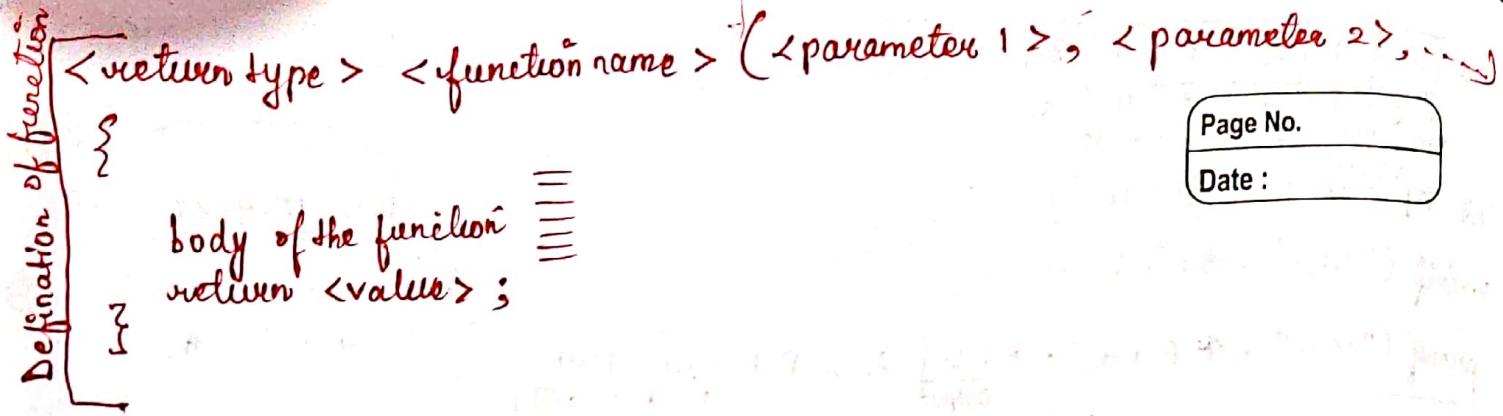
parameters

```
{  
    int z;  
    z = x + y;  
    return z;  
}
```

int main()

```
{  
    int a, b, c;  
    scanf ("%d %d", &a, &b);  
    c = add (a, b);  
    printf ("%d", c);  
    return 0;  
}
```

Teacher's Signature



→ Factorial of a number :

```

int factorial (int no)
{

```

→ ABC

— — —

KP

— — —

DE

— — —

void dis()

```

{
    printf ("---\n");
    // given ; even (you may or may not write this) :
}
```

int main()

```

{
    printf ("ABC\n");
    dis();
    printf ("KP\n");
    dis();
    printf ("DE\n");
    dis();
}
```

Teacher's Signature

ASSIGNMENT- 22

Page No.
Date. 12/4/21

- ① Write a function to find out minimum among two numbers.
- ② Write the function to find factorial.
- ③ Write a function to ~~return~~ display the fibonacci number upto 10th value.
- ④ Write a function to ~~return~~ 10th fibonacci number
- ⑤ Write a factorial function and use it for computing

$$\frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{15!}$$

- ⑥ Write a function which will take an integer along with character and display the following :

Input : 3
*

Output : * * *
* *
*

- ⑦ Write a function to find out no of digits present in the given input .

e.g. 12965 \Rightarrow 5

- ⑧ Write a function which will take 3 values

e.g. 9 - + \Rightarrow 16

e.g. 9 - * \Rightarrow 63

%

1

Teacher's Signature

```

int printf (char *, ...);
int main()
{
    printf ("ABC");
    printf ("---", a, b);
}

```

elipsis → variable size parameter

Page No.
Date : 13/4/21

Declaration printf
funⁿ is present in
stdio header file.
Definition of printf funⁿ is present
in library files.

```
float sum (int); → declaration
```

funⁿ name
type of parameter
no of parameters
return type

```
int main()
{
    sum(); → calling the function
}
```

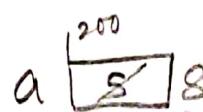
```
float sum(int a)
{
    → definition
}
```

* Call by Value and Call by reference:

```

void f1 (int a)
{
    a = a + 3;
    printf ("%d", a); 8
}

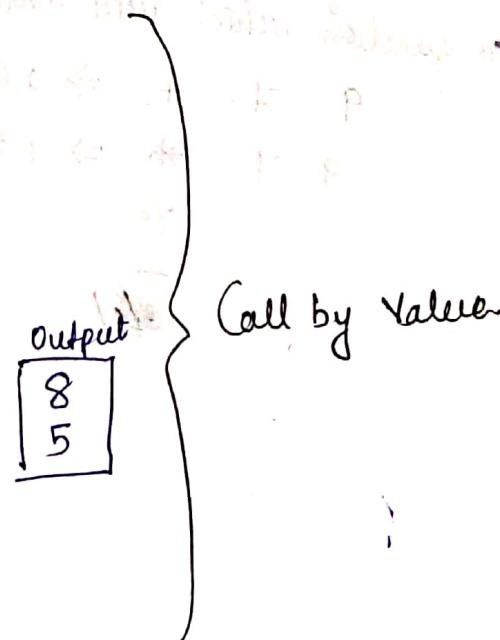
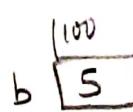
```



```

int main()
{
    int b = 5;
    f1(b);
    printf ("%d", b); 5
}

```



Teacher's Signature

```

void f( int *a )
{
    printf("%d", *a); // a [100]
    *a = *a + 3; // a + 3 = 8
    printf("%d", *a); // a [100]
}

int main()
{
    int b = 5; // b [5] [8]
    if (&b);
    printf("%d", b); // b [100]
}

```



Page No.	1
Date:	

Call by Reference .

* A function can return only one value . So call by reference is used for returning giving multiple values through pointers .

e.g. void cal (int x , int y , int *p , int *q)

```

{
    *p = (x+y)/2;
    if (x < y)
        *q = x;
    else
        *q = y;
}

```

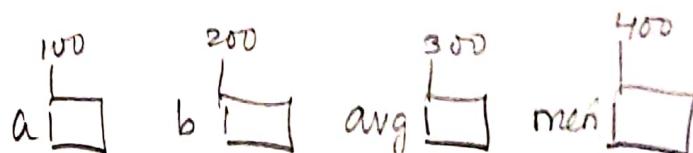


int main()

```

{
    int a, b, avg, min;
    scanf ("%d %d", &a, &b);
    cal (a, b, &avg, &min)
    printf ("%d %d", avg, min);
}

```



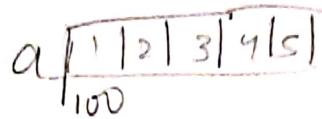
Teacher's Signature

→ Passing array :

int main()

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

 dis(a, 5);



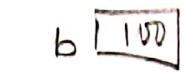
Page No.

Date :

?
void dis(int b[], int no)

{
 int i=0;
 for(i=0; i<no; ++i)
 printf("%d\n", b[i]);

void disp(int *b, int no)
{
 int i=0;
 for(i=0; i<no; ++i)
 printf("%d", *(b+i));
}



Teacher's Signature

ASSIGNMENT-23

Page No.

Date: 13/4/21

- ① Write a function which will take 3 values and return back the average, max, min.
- ② Write a function which will take an array and return back the summation of all odd elements present in the array.
- ③ Write a function which will receive a 4 digit no and swap the first digit with last digit. Display the resultant no in the main. (Call by reference)
- ④ Let a main function has two integer arrays of same size. Write a function which will store the first array's data in another array. in a reverse manner. Display the second array in main function.
 f1 (int *p, int *q, int s)
- a [] { 3 a [5] 9 6
b [] : 7 1 b [6] 9 5
- ```
scanf ("%d", &a); // this will take enter making the buffer empty
getchar(); // so this can take the next character & not
scanf ("%c", &c); // enter as character;
```

SWINTEX - 2021

Teacher's Signature

int main()

{

int a[2][3] = { {1, 2, 9},

{ 6, 4, 5 } }

call (a, 2, 3); des (\*a, 2, 3);

}

void des (int b[2][3], int x, int c)

{

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

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

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

b[1][2] = 7;

}

|   |   |   |   |   |
|---|---|---|---|---|
| a | 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 9 |   |
| 1 | 6 | 9 | 5 |   |

a → address of this row  
\*a → address of 00 element

int (\*b)[0] → pointer to an array

→ length of string

→ copy the string

→ compare the string

→ joining two strings

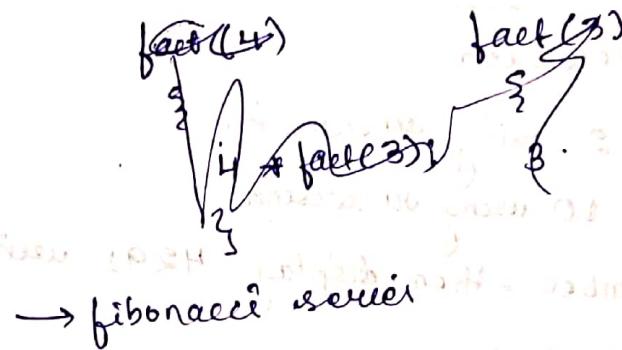
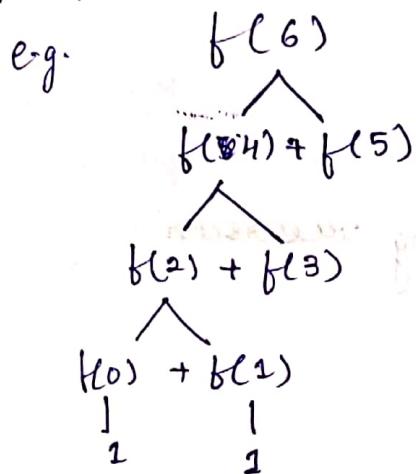
- string.h
- ① `strlen()` — length
  - ② `strcmp()` — compare
  - ③ `strcpy()` — ~~shallow~~ copy
  - ④ `strcat()` — concatenation

Page No.

Date :

### \* Recursion : (function calls itself)

Solution of a bigger problem can be computed by the solution of the same type problem with smaller values.



### Fibonacci

```

int main()
{
 int a, x;
 x = fact(4);
 printf("%d", x);
}

```

```

int fact(int no)
{
 int ws = 1;
 if (no == 0 || no == 1)
 return 1;
 else
 for (; no > 1 ; no--)
 ws = ws * no;
 return ws;
}

```

QUESTION - ANSWER

Teacher's Signature

int fact ( int no )

{

if ( no == 0 || no == 1 )

return 1 ;

else

return no \* fact ( no - 1 ) ;

}

Page No.

Date :

### ASSIGNMENT - 24

17/4/21

- ① Display 1-10 using recursion
- ② Display 10 8 6 4 2 using recursion.
- ③ Find summation of 1-10 using recursion.
- ④ If 1924 is a number, then display 4291 using recursion.
- ⑤ Find g.c.d using recursion.

Unacademy Student

Teacher's Signature

## STRUCTURE

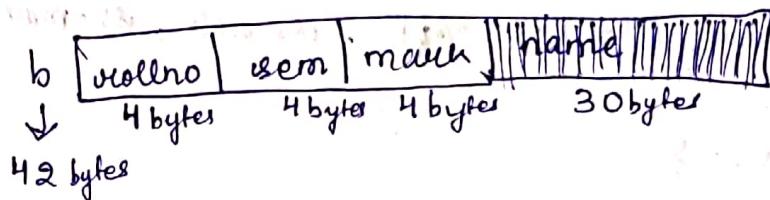
```
struct student {
 int rollno;
 int usem;
 int marks;
 char name[30];
};
```

Page No.

Date : 19/4/21

→ Structure is a user defined data type that consists of different types of member variables.

student student      b ;  
                        ↓  
                        data type      variable



`sizeof (b);` → 42

b·violino

b'userr

b · maulk

b.name

scanf ("%d", &b'rollno);

pointf ("%.d", b.vollno);

## \* Syntax?

street <name of streetview> {  
    member variables;  
    member variable;  
    :  
    :

۲۰

### Teacher's Signature

```
#include <stdio.h>
struct student {
 int roll;
 float mark;
 int sem;
```

```
};
```

```
int main()
{
 struct student s1;
 s1.roll = 100;
 s1.mark = 85;
 s1.sem = 3;
 printf("roll %d mark %f sem %d", s1.roll, s1.mark, s1.sem);
}
```

```
struct stu {
```

```
 int roll;
 float marks;
 float sub1;
 float sub2;
 float sub3;
```

```
};
```

```
int main()
```

```
{
 struct stu s1;
```

Teacher's Signature

```

struct stu {
 int roll;
 int name;
 float usub[3];
};

int main()
{

```

```
 struct stu us[4];
```

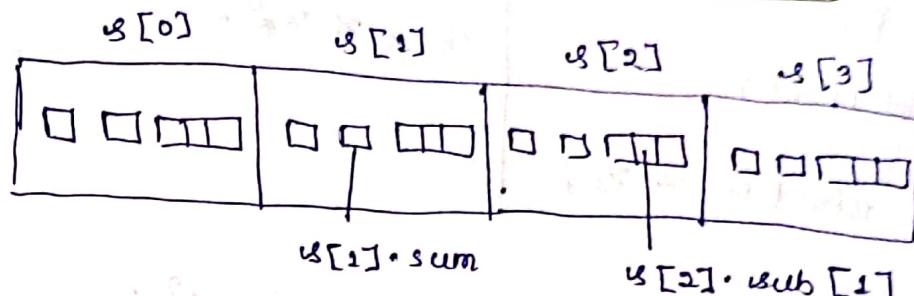
```
 int i, j;
```

```
 for (i = 0; i < 4; ++i)
```

```
{
 scanf ("%d %d", &us[i].roll, &us[i].sem);
}

for (j = 0; j < 3; j++)
```

```
 scanf ("%f", &us[i].usub[j]);
```



19/4/21

### ASSIGNMENT-25

- ① Collect the student information (roll no, name, CGPA) and display.
- ② Collect the student information (roll no, name, usub 1, 2, 3) and display the student detail with total percentage.
- ③ Collect two dates and find out which is earlier one. (difference)  
 ↓  
 (dd, mm, yy)
- ④ Take 3 employees information (empid, empname, basic salary)

Display empid, name, gross salary

$$\hookrightarrow \text{BS} + 10\% \text{ DA} + 5\% \text{ HRA}$$

$$100 + 10 + 5 = 115$$

Display what is the total amount paid to these 3 employees.

Ans:

Ques:

Challenged

Teacher's Signature

⇒ struct  $x \{$

int a ;

int b ;

$\} ;$

int main()

{

struct  $x \{ x_1, x_2 \};$

struct  $x \{$

int a ;

int b ;

$\} x_1, x_2 \};$

struct  $x \{ x_3 \};$

Page No.

Date : 20/4/21

⇒ struct  $x \{$

$x_4 = 5 ;$

$x_4 \cdot a = 5 ;$

struct  $x \{ x_4 = \{ 5, 7 \} ;$

$x_4 \cdot b = 7 ;$

struct  $x \{ x_4 = \{ 5, 7 \} ;$

⇒ struct stu {

int roll ;

char name [20] ;

float sub [2] ;

$\} ;$

struct stu s1 = {1096, "ABCD", {5, 19}};

⇒ typedef :

typedef is used to create an user defined data type.

typedef int P ;

P a, ;

P b, c ;

P replaces int .

typedef struct {

int roll ;

char name ;

} stu; data type

stu s1, s2, s3 ;

Teacher's Signature

```

struct stu {
 int roll;
 int marks;
} us1;

```

Page No.

Date:

```

typedef struct stu uss;
uss us2, us3;

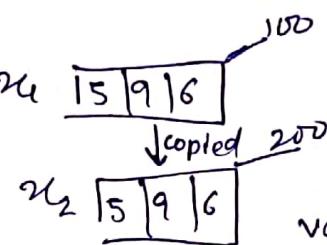
```

int main()

```

} struct x x1 = {5, 9, 6};
struct x x2;
x2 = x1;

```



values changed in x2  
won't reflect in x1.  
both are diff.

```

structs int a = 5; a 100
int *p;
p = &a; p 200
*p => 5
*(100) => 5

```

struct x {

int a;

float b;

} ;

int main()

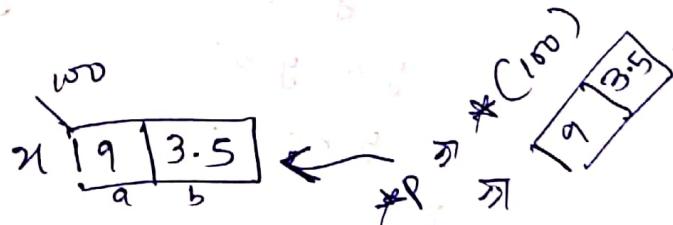
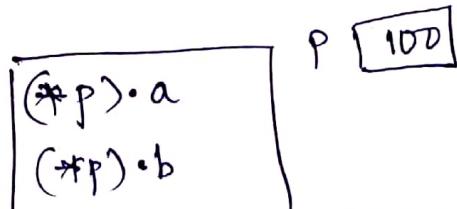
```

} struct x x1 = {9, 3.5}

```

struct x \*p;

p = &x1;



Teacher's Signature

struct  $x \{$

int a;

double b;

\};

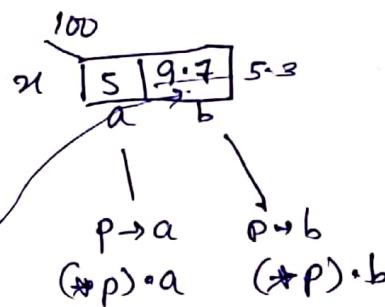
struct  $x x_1 = \{ 5, 9.7 \}$

struct  $x *p = &x_1;$

double \*q = &(x\_1.b);

$$*q \Rightarrow 9.7$$

$$*q = 5.3;$$



struct  $x \{$

int a;

int b;

\};

int main()

\{ struct  $x x_1 = \{ 7, 3 \}$ ;

f1(x1);

printf("%d %d", x1.a, x1.b);

printf("%d %d", x1.a, x1.b);

f2(\*x1);

void f1 (struct  $x y$ )

\{

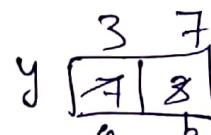
int c;

c = y.a;

y.a = y.b;

y.b = c;

\}



Teacher's Signature

void f2 (struct  $x$  \*y)

3. 7

{  
    int c ;

24 A) B

Page No.

Date :

    c = y ->a ;

    y ->a = y ->b ;

    y ->b = c ;

}

struct  $x$  {

    int a ;

    int b ;

}

int main ()

{  
    struct  $x$  x1 = { 5, 7 } ;

    struct  $xes$  x = cal (x1) ;

    printf ("%.1d %.1d ", x .avg , x .max ) ;

    return 0 ;

struct  $xes$  {

    float avg ;

    int max ;

}

struct sees cal (struct  $x$  y)

{  
    struct sees t ;

    t .avg = (y .a + y .b) / 2 ;

    if (y .a > y .b)

        t .max = y .a ;

    else

        t .max = y .b ;

    return t ;

}

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

## ASSIGNMENT - Q6

- ① Create a structure having 3 elements. Through a pointer, you need to display all the elements of the structure variable.
- ② Create the above structure and pass 3 values to a function and return back the average of 3 values.
- ③ Create item structure (item no, item name, quantity, unit price, discount (in %)).  
Take 3 items details and pass it to a function where you need to display the following

|          |         |
|----------|---------|
| Page No. | 20/4/21 |
|----------|---------|

| <u>Item no</u> | <u>Item name</u> | <u>quantity</u> | <u>unit price</u> | <u>dis</u> | <u>Amount</u> |
|----------------|------------------|-----------------|-------------------|------------|---------------|
|                |                  |                 |                   |            |               |
|                |                  |                 |                   |            |               |
|                |                  |                 |                   |            |               |
|                |                  |                 |                   |            |               |

- ④ Write a structure which will take student information which consists of (roll no, sem, sub[3], avg)  
Pass this student info to a function which will update the average mark. Display the detail about the student in the main.  
(Call by reference).

Teacher's Signature

## Nested Structure

```

struct date {
 int dd;
 int mm;
 int yy;
};

struct user {
 int uroll;
 int usem;
 int mark;
 struct date jdate;
 struct date dob;
};

};

ss.uroll = 1093;
ss.jdate.dd = 13;
ss.jdate.mm = 4;
ss.jdate.yy = 2020;

```

struct user {

int uroll;

int usem;

int mark;

struct date {

int dd;

int mm;

int yy;

? jdate, dob;

? ss;

Page No.  
Date: 26/4/21

?;

struct date jdate, dob;

\* struct info {

int x;

int y;

? P;

↳ variable

typedef struct info {

int x;

int y;

? P;

↳ data type

typedef struct {

int x;

int y;

? P;

P, P.1;

end of lesson

Teacher's Signature

## Union :

Used for creating user defined data type.

This is a collection of different types of variables which are sharing a single block of memory.

Page No. \_\_\_\_\_

Date: \_\_\_\_\_

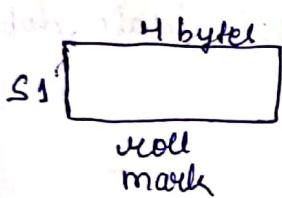
union stu {

int roll;

float mark;

?;

union stu us1 :



sizeof ( $S_1$ )  $\rightarrow$  4 bytes

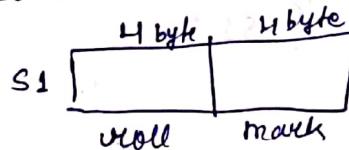
struct stu {

int roll;

float mark;

?;

struct stu ss1 ;



sizeof ( $ss1$ )  $\rightarrow$  8 bytes

The size of the union is  
the size of the variable  
with maximum memory

union stu {

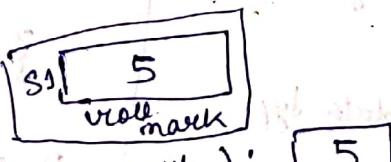
int roll;  $\rightarrow$  4 bytes (4 > 2)

float mark;  $\rightarrow$  2 bytes

?;

union stu s1 ;

$s1.roll = 5;$



printf ("%d", s1.mark);

[5]

Teacher's Signature

struct istu {

int sem ;

int tgrp ;

union {

int roll ;

char name [20] ;

{ id ;

} istu ;

us1.sem = 3 ;

us1.tgrp = 92 ;

strcpy ( us1.id.name , "ABC" );

Page No.

Date :

struct traveller {

char name [20] ;

int age ;

char state [15] ;

union identity {

int vid ;

int pes ;

int add ;

} id ;

char id-type ;

} p1 ;

} are required

} any one is required

} in union is used.

struct voterid {

int vot\_no ;

int .

int age ;

} ;

struct passport {

int passno ;

char PA [20] ;

int no\_T ;

} ;

struct idhar {

int ad\_no ;

int area\_code ;

int age ;

int bg ;

} ;

struct

( next page )

Handwritten assignment

Teacher's Signature

## struct traveller {

char name [20];

int age;

char state [20];

## union identity {

struct voterid vid;

struct passport p;

struct adhar adh;

{ id;

char id-type;

} p;

p1.id · pas · pass-no = - - ;

\* in main ( )

El autor de

## union info {

int a;

char p b;

};

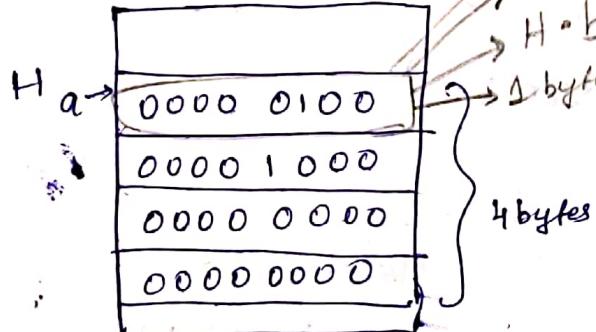
int main ()

{ union info H;

H.a = 260;

printf ("%d", H.b); } 4

}



Teacher's Signature

\* enum : This is used to declare named integer constants.

if (1)  
{  
} = } if (true)  
{  
}

enum valid { False, True };

enum valid {  
 False ;      False  $\leftarrow$  0  
 True ;        True  $\leftarrow$  1  
};

First name is assigned 0 value  
Second name is assigned 1 value

enum valid v1;  
data type      Variable

enum days {  
 0 - sun,  
 1 - mon,  
 2 - tue,  
 3 - wed,  
 4 - thurs,  
 5 - fri,  
 6 - sat.  
};

enum days {  
 sun = 2, — 2  
 mon = 5, — 5  
 tue, — 6 (previous value + 1)  
 wed = 0, — 0  
 thur, — 1  
 fri, — 2  
 sat = 2 — 2  
};

int p = thurs;  $\boxed{P \Rightarrow 4}$

Page No.

Date:

Printed by CamScanner

Teacher's Signature

## ASSIGNMENT-27

① Create a student structure (roll, dob, join)

Collect 2 students' information

joining date

[street date (dd, mm, yy)]

Display the age of two students

Find which student is the elder one.

Page No.

Date : 28/4/21

② Collect 3 employee's information. (empid, type of employee (part time / full time),

(dob / age), salary)

Display the 3 employees' information

union

street date dob  
int age.

empid

type

dob / age -

salary

③ Write the recursive function for finding gcd

④ Collect 4 student information in an array and pass it to a function which will return the highest average mark, highest

mark in each subjects

roll no

name

cgpa

physics marks

chemistry marks

## Visibility of Variables :

- block scope
- program scope
- function scope
- file scope.

Page No.

Date : 27/4/21

int main()

```
{ int a=6; a [6] ↗ ←
 a++; ↗
 printf ("%d", a); - [7]
```

```
{ int a=3, b=7;
 b--; ↗
 a++; ↗
```

a [3] 4  
b [7] 6

local variable  
function preference  
printf ("%d %d", a, b); [4 6]

a++; ↗  
printf ("%d", a); [8]

goto : Cannot move its control outside of the function

f1()

```
{
 =
 goto ABC:
 =
```

f2()

= original address

ABC:

=

Teacher's Signature

```

int main()
{
 four()
 {
 goto P;
 }
 four()
 {
 p:
 }
}

```

applicable  
present in different blocks in the same  
function.

### Program Scope :

- Global variable
- local variable

#### Global variable

- ① It is a variable which is declared outside of a function.
- ② Global variable is used in any function in the program.
- ③ Cannot be initialized inside a function. Can only be used for calculation.
- ④ Only one instance is created. (by default, the initialization value is 0)

#### local variable

- ① It is a variable which is declared inside a function.

- ② Multiple instances are created.

Teacher's Signature

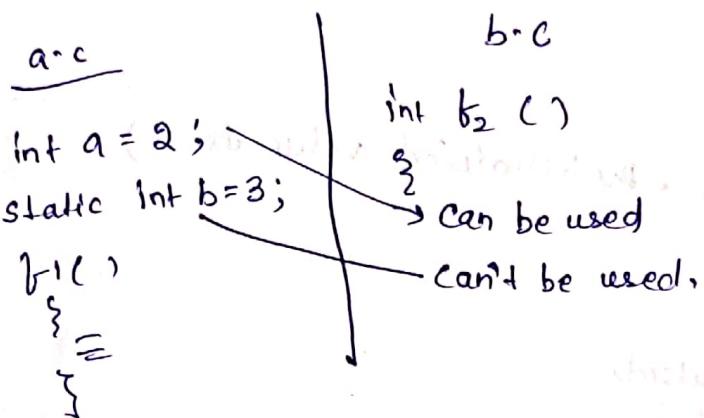
## File Scope :

If a global variable is declared with a `static` specifier then it cannot be used outside of a file.

A global variable specified with `static` maintains the file scope.

Page No. \_\_\_\_\_

Date: \_\_\_\_\_



## Storage Class :

Information provided by storage classes:

- ① Scope of the variable
- ② Life time of variable
- ③ The variable is allocated in which memory.
- ④ whether the variable is automatically initialized or not.

## Types of storage:

`auto`, `register`, `static`, `extern`

→ every local variable's default type is `auto`.

void f1()

{

int a=2; /auto int a=2;

a++;

printf ("%d", a);

}

Evidence & Solution

- Scope of this local variable : block / function scope.
- Life time : only during control in function `f1` block.
- allocated in the RAM. / garbage
- 

Teacher's Signature

## register variable :

```
void f() {
 register int a = 2;
 a++;
 printf ("%d", a);
}
```

→ stored in register in CPU  
address can't be traced  
&a can't be used,  
not automatically initialized

Page No.

Date:

 $\&a \rightarrow X$ 

static variable : (by default, the initialized value is 0)

static local → variable declared inside a function

static global → declared outside of a function.

```
void f() {
 static int a = 2; a [2]
 a++;
 printf ("%d", a);
}
```

(only one instance)      output: 3 4

```
int main()
```

```
{ f();
f(); }
```

```
}
```

Scope : block scope / fun<sup>n</sup> scope  
Life time : throughout the program  
Allocation : RAM  
Initialization : yes (0)

Teacher's Signature

```

static int a=5;
f1()
{
 a++;
 printf("%d", a);
}
f2()
{
 a=a+2;
 printf("%d", a);
}
int main()
{
 f1();
 f2();
 f1();
}

```

Scope : program  
 life time : throughout the program execution  
 Allocation : RAM  
 Initialization : Yes(0).

|          |
|----------|
| Page No. |
| Date:    |

extern : (is applicable only for global variable)  
extern int a; → tells the compiler that the variable a is present somewhere & look for it

```

void f1()
{
 a++;
 printf();
}
int a=2;
void f2()
{
 a++;
 printf();
}
int main()
{
 f1();
 f2();
 f1();
}

```

extern is used when multiple files are combined together

```

a.c
int t=5;
int main()
{
 printf("%d", t);
}
b.c
extern int t;
int main()
{
 printf("%d", t);
}

```

but if  
 t would have  
 been static,  
 that cannot be  
 accessed outside its file.

Teacher's Signature

**extern** :- scope : same file / diff file

Scope : same file / diff file  
Lifetime: throughout the program execution

memory : RAM.

Initialization : Yes ( $\emptyset$ )

Page No.

Date :

Q) Take a structure (int no, int [10], int count)

In main , create a variable of this structure

store the no , take from user

Pass the variable to a function

stone the number it get we in array

Then count the digits

main( )

۳

91 ;

$s_{1-20} \Rightarrow 29647$

A diagram illustrating a stack of integers. A bracket labeled 'S' indicates a stack of five elements. An arrow labeled 'f1 C)' points from the stack to a horizontal box containing the digits 2, 9, 1, 6, and 4, separated by vertical bars. Below the box, an arrow labeled 'count = 5' points to the value 5.

Teacher's Signature

Dynamic Memory Allocation  
We will allocate the memory dynamically.

Dynamic we will allocate the memory during runtime.

`malloc` → allocation of memory  
      "      "      "

`malloc`      "      "      "  
`calloc` →      "      "      "

~~calloc~~ → resizing the allocated memory  
~~realloc~~ → ~~allocating~~

~~seam~~  
free → deallocation of memory

~~p = malloc (n \* sizeof (int))~~  
give this address

```
for (i=0 ; i<n ; i++)
 scanf ("%d", &p[i]);
 (p+i)
```

→  $p = (\text{int } *) \text{malloc}(n * \text{sizeof}(\text{int}))$   
↓ type casting

free(P);

**Page No.**

Date : 3/5/2

man malice

Inclued *for* *handout*  
file  
for these  
handouts

## stdlib.h

A diagram illustrating memory storage. A vertical line labeled "100" at its top represents an address. Below it, a horizontal line is divided into eight equal segments by vertical grid lines. The label "40 bytes" is written above the first four segments of the horizontal line.

## Heap area

Local variable are allocated in stack  
global variable " " " Data segment

Dynamic memory → heap segment

static ~~variable~~ → Data segment

void ~~is~~ malleable (vit);

\ generate pointer → which  
can be converted to  
any kind of pointer

any

→ destroys the memory

Callos :

void \* callce (no of element , size of each element)

int n;

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

int \*p;

`p = (int *) Calloc (n, sizeof (int))`

Mallo doesn't fill  
0 in its memory  
whereas callee does.

Teacher's Signature

realloc :

Page No.

Date :

### ASSIGNMENT - 28 (3/5/21)

- ① Create a dynamic memory allocation using malloc for certain numbers given by user and display all the numbers along with the maximum value.
- ② Create DMA using calloc to collect employee information (empid, empname, sal) of 3 employees. Further you need to increase 2 more employees information and find the average salary (use realloc).
- ③ Create DMA using calloc to collect employee info of 3 employees and pass this information to a function where you need to find out the total salary of these 3 employees. If the total salary is less than 10,000 then add two more employees. All the employees info need to be displayed in the main function.
- ④ Create 3 pointers and store 3 values using these 3 pointers and find out max, min and middle value among these 3 values, and display (Note: Do not use any normal variable).

student signature

Teacher's Signature