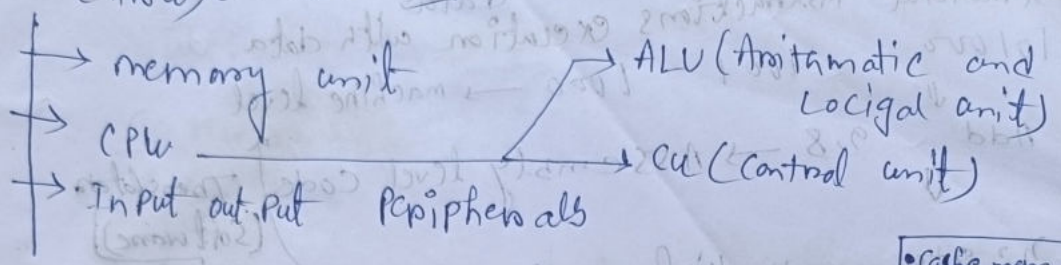
17/3/22

Computer (throw) [hardware]



- cache memory
- Cabinet

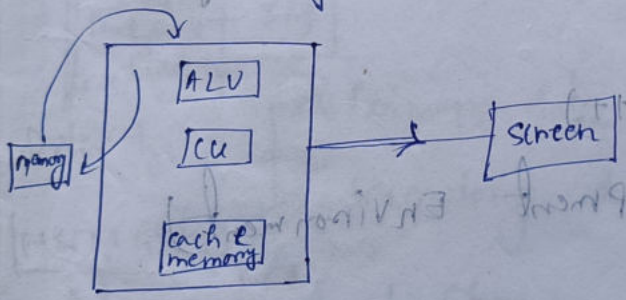
Optimal Solution

ASCII values

① Find out the day from the date.

2nd march, 1990

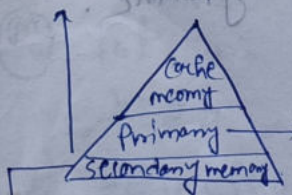
◉ the Processer for working memory is called cache memory, that is register actually.



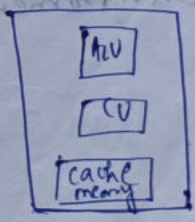
CPU

1GB/mB → Fast (Speed increasing)

volatile



RAM and Rom (main) (axilary)



→ HDD (Hard drive, operating System install হয় এখানে)
→ SDD

◉ ~~~~ multithreading don't need multitasking
multitasking need multithreading must

Programme = execution of sequencial instructions
Sequencial instructions execution with datas

10101010 1001 1000 → machine level

Add 9,8 → Assembly level code → Translator (Software)

x = 9+8 → High level

(or) x = 9, y = 8
z = x+8

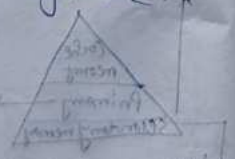
Compilers [turbo C++ / dev C++] Interpriters

Software → System Software
→ Application Software

- Compiler is a system software, that can transform HL things to machine level things.

- IDE (~~Compiler~~ turbo C++)
  ↳ (Integrated Development Environment)

  i) Compiler
  ii) Editor
  iii) Provides environment to program the programme.



C-Programing 29/3/22

Inputs - finite set of inputes

alphabets → a - z
A - Z

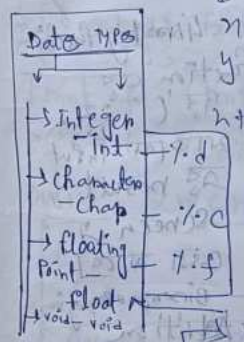
Numbers → 0 - 9

Symbols → >, <, =, ==

Variable
x, y, z, A, B, ... etc

constant
5, 3 ...

Keywords (32)
which one we cannot exchange



x = 5;
y = 3;
x+y;

Assingment operator
[5 is assinged to x]
[';' is the end of the statement]
[float → 42.15
int → 0, 9, 10, 11]

Data types → Primary data types
→ Secondary date types / Derived data types ↓ (function, Array, string, pointer, structure, enum)

int x = 5;
int y = 3;
z = x+y;

int z;

char x = 'A';
float a = 3.506;

(or) int x;
x = 5;

```
#include<stdio.h>
void main()
{
  int x,y,z;
  x=5;
  y=3;
  z=x+y;
```

Print f ("Result is:%d", z);
getch ( ):
}

[Access specifier]
[Print f = library function / system defined function]

[function
→ users defined
→ system defined]

```
#include<stdio.h>
void main ( )
{
int x, y, z;
x = 5;
y = 3;
z = x+y;
Print f ("Result is: %d", z);
getch ( ):
}
```

Library function

[function এর ( ) মধ্যে input এর result screen এ দেখা যাবে]

Binary operation
[Addition operation = +
Substraction = -
multiplication = *
Dividation = /]

```
#include <stdio.h>
void main ( )
{
int x, y; float z;
x = 5;
y = 3;
z = x/y;
Print f ("Result is: %f", z);
getch ( ):
}
```

Output screen

Result is 1.0

[∵ The dividation of two intigers will be an intiger, it will not be a float]

```
#include<stdio.h>
void main ( )
{ int x, y; float z;
z = x+y;
Print f ("Enter two integer values \n");
Scan f ("%d%d", &x, &y);
```

Output screen

Enter two integer values 4 13 6
Result is 419

[\t = Back slash t]
[\n = new line]
[& = Address]

1/3/22

```
int z
int x = 4, y = 5;
Print f ("%d %d", x, y);
```

Data type – double = %lf [10.3678903127]
↓ format specifier

[_ = underscore]
[float – After point 6 letters, double – After point 14 letters]

modifiers

signed unsigned short long

operators

(1) Arithmetic : +, -, *, /, %

(2) Logical operators:
||, && (Logical OR, Logical AND)

(3) Relational ==, operators : (comparision)
<, >, <=, >=

4) Pre-incriment ++x;
Post-incriment x++;

5) Conditional operator

6) Bitwise operators

5% -2 = -2.5
-5%2 = -2.5
-5% -2 = 2.6

logical OR

a	b	a || b
0	0	0
0	1	1
1	0	1
1	1	1

```
int x = 5;
Printf("%d %d %d", x);
```

6 → Result

(Incriment will be done by '1' and dicrimenent will be done by '1')

```
int x=5, y=4;
z = x++ + ++x + y++ + ++y;
Printf("%d %d %d", x/6, y/5)
```

(= 10)

Header file

```
#include<stdio.h>   → Pre Processor directive
#include<conio.h>
void main()          → name of the function
{
  Printf("Hellow world");
  getch();
}
```

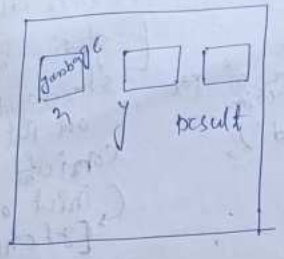
[stdio.h — standard input output Conio of console (input output) [getch();]]

[void, return type function]

[before no return type - function coming]

```
#include<stdio.h>
#include<conio.h>
int main()
{
  Printf("Hellow world");
  return 0;
}
```

```
#include<stdio.h>
int main()
{ int x, y, result;
  Printf("enter 2 int values");
  scanf("%d%d", &x, &y);
  result = x+y
  Printf("Addition :- %d", result);
  return 0;
}
```



[the value we don't know is garbage value]

[int x; = declaration
int x=5; = declaration and initialization]

4.4.22

CW

① Write a C programme for finding out whether the number is even or odd.

```
# include < stdio.h>
 int main ()
{
 int a;
 Printf ("Enter an integer value\n");
 Scanf ("%d", &a);
 if (a %.2 == 0);
  Printf ("Even no");
  else
   Printf ("odd value");
   return 0;
  }
```

③ There are three integer values, which one is maximum.

```
# include < stdio.h>
  int main ( )
 {
  int a, b, c;
  Printf ("Enter three integer values\n");
  Scanf ("%d %d %d", &a, &b, &c);
   if a > b
   if a > c
        max a;
  else
        max = c;
   else
    if (b > c)
        max = b
    else
        max = ( );
  Printf ("maximum is: %d", mg);
   return 0;
```

Conditional operator

max = (a>b) ? (a>c); a:e): (b>c); b:c);

```
int x = 5;
if (x == 5);
    Printf ("A");
else
    Printf ("B")
```

(A)

```
float x = 6.3
if (x == 6.3);
    Printf ("A")
else
    Printf ("B")
```

(B)

Loop

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

WAP to fix sum of n ~~natural~~ number

Addition of first 'n' int numbers

```
#include <stdio.h>
int main()
{
    int x;
    Printf("Enter the value\n");
    Scanf("%d", &x);
    for (i=1; i<=x; i++)
    { sum : sum +i};
    Printf ("Result [S:%d", sum);
```

8/4/22

CM

① ~~Write~~ a C Programme whether any number is Prime or not. (for a Particular no.)

```
#include <stdio.h>
int main ()
{ int a;
  Printf (" Enter an integer value\n");
  Scanf ("%d", &a);
  if (a%a==0 && a%1==a);
    Printf (" Prime number ");
  else
    Printf (" not a Prime number");
  return 0;
}
```

for 'n' no.

```
{int n, i; flag=0;
 for (i= 2; i<= n/2; i++);
   {if (n%i==0)  { flag=1
     { Printf ("not Prime");
       break;
     }
   else
   }
Printf (" Prime no")
```

(keyword) break (break the loop)

```
for(i=1; i<5; i++)
  for(j=1; j<5; j++)
  {if (i == j)
      break;
   else
   Printf("%d%d\n", i, j);
  }
```

output →

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

i=1 j=1 break
i=2 j=1 Print
i=2 j=2 break
i=3 j=1 Print
i=4 j=3 Print
i=4 j=4

['j' will continue till it don't complete its execution]

If break occurs then we won't have any control in loop.

(key word) continue → (Skip the condition)

output →

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

i=1 j=1 continue
i=1 j=2 Print
i=1 j=3 Print
i=1 j=4 Print
i=2 j=1 Print
i=2 j=2 continue
i=2 j=3 Print
i=2 j=4 Print
i=4 j=3 Print
i=4 j=4 Continue

```
for (i=1; i<5; i++)
  for (j=1; j<5; j++)
  { if (i==j)
      continue;
    else
      Printf("%d %d\n", i, j);
  }
```

```
for(i=i; i<5; i++)
for(j=1; j<5; j++)
  { if (i==j)
      continue;
    else
      printf("%d%d\n",
        i,j);
  }
```

Previous sum

```
int n, i, flag = 0,
  for (i=2; i<= n/2; i++)
  { if (n%i == 0)
    { flag = 1
      break
    }
  }
if (flag == 0)
  Printf("Prime no")
```

Perfect number → 6 ← 1+2+3=6 (divisors)

```
int main ( ).
{ int n, i , sum=0
   for (i = 1 ; i++);
      if (n % i = = 0)
         sum = sum + i;
   }
```

~~print f ("perfect number")~~
~~int x, y, z, m~~
~~m = x+y+z~~
~~if (n == m);~~
~~Print f ("Perfect number");~~
}

```
if (sum = = n)
Print f ("perfect no");
```

Palindrone

121 (The no. and it's reverse is same)
19991

```
int main ( )
{ int n,i
```

NOTE =

```
int n = 5;
Print f ("%d %d %d", n++, n, ++n);
           6   6   7
```

Imp things in 'c'
{ Precedence and associativity }

[Precedence is for operators]

[int x1, n2, ... int 1x, 2x...]

(char) (char/num/-)

[i++; → i = i+1;] = n+=1;

while (keyword)

```
while (  a > b  )   condition
{
   ___
   ___        ] body portion
   increment / decrement variable
}
```

```
int i=1;
for ( ; (i < 5; ) )
{ - -
  - -
  i++;
}
```

Same

```
for (i=1; i<5; i++;)
{
}
```

Infinite loop -

```
while (1)
{
  ...
}
```

dowhile

```
do
{
  ...
}
while( );
   Condition
```

```
i = 1;
do
{
  ...
  i++;
}while(i<5);
      Condition
```

```
#include <stdio.h>
int main()
{
  int n, flag=0, i, sum=0;
  Printf("Enter the integer no:");
  Scanf("%d", &n);
  for(i=1; i<=n; i++);
  {
    if(n%i==0)
      sum+=i;
  }
```

```
if (sum==n)
Printf("perfect number.");
else
Printf("not a perfect number");
  return 0;
}
```

Palindrome —

```
rev=0;
n=n;
while(n>0)
{
  d=n%10;
  rev=rev*10+d;
  d=n/10;
}
```

d = 3
d = 2
d = 1

n = 123

rev = 0
rev = 3
rev = 32
rev = 321

```
if (rev==n)
```

12/4/22

HW Assignment

① How to check Armstrong no. [153 $1^3+5^3+3^3=153$ Armstrong no.]

Hints

```
sum = 0;
mul = 1;
while (n, >0)
{ d = n,%. d
   for (i=1, i<=n; i++)
          mul = mul * d;
   sum = sum + mul;
   n = n/10;
}
```

Pattern

For loop, Printing : simple

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

if n = 3

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

① #include <stdio.h>

```
int main()
{ int i, n;
  Printf("Enter the no. of lines \n");
  Scanf("%d", &n);
for( i=0; i<n; i++)
  {for (j=0; j<=i; j++)
     Printf("*");
   Printf("\n");
  }
```

②

```
{ int i, n;
int c = 1
  for (i=0; i<n; i++)
   {for (j=0; j<n-i; j++)
     { Printf("%d", c);
       c++;
     }
   Printf("\n")
  }
```

```
1 1 1
2 2
3
```

n=4

i=0			*
i=1		*	*
i=2	*	*	*
i=3	*	*	*

n=4

i=0

j>0; 1, 2 ✗

i=1

j>0, 1, 2 ✗

```
int main()
{ int i, n;
for(i=0; i<n; i++)        → how many lines
  { for (j=0; j<n-i-1; j++)   → line for space
      Printf(" ");
    for(k=0; k<=i; k++)
      Printf("*");
```

printf("\n");

Assignment
② Pattern Printing — (Print this 2)

i)

ii)

iii) → stars Printing

- Switch — case
(keyword) (keyword)

choice = ch

```
#include <stdio.h>
int main()
{
    int x, y, ch, a, b, c, d
    Printf("Enter two integer values");
      Scanf("%d%d", &x, &y);

    Printf("1. Addition \n
            2. Substraction \n
            3. multiplication \n
            4. Division");
    scanf("%d", &ch);
    switch(ch)
    {
```

```
case 1: a = x + y
  Printf("Result: %d", a)
case 2: b = x - y
  Printf("result: %d", b)
case 3: c = x * y
  Printf("result: %d", c)
case 4: d = x / y
  Printf("result: %d", d)
default;
  Printf("Invalid choice")
}
return 0;
}
```

15 5
3 Result of mul = 7.5
0 Result of div = 3
2 5 Invalid choice

(3)

[After the no. 3 all choices ~~are~~ executed]

for only your choice, (you want to execute only your choice)

```
case 1:
    Print
    break;
case 2: Print
    break;
case 3: print
    break;
```

```
case 4:
    Print
    break;
```

② Assingment

Pattern Printing -

(i) /* Programe to Print pyramaid pattern in C */

```
#include <stdio.h>
int main()
{
    int i, j, k, t=0
    clrscr();
    for (i=1; i<=5; i++)
    {
        for (k=t; k<5; k++)
            { Printf(" ");
    }
        for (j=0; j<i; j++)
        {
            Printf("*");
            t=t+1;
        }
    Printf("\n");
}
```

```
}
```

Out Put

(ii) /* Program to Print pyramid pattern in C */

```
#include <stdio.h>
int main()
{
    int i,j;
    for Printf("Enter the no of lines\n");
    for (i=5; i>=1; i--)
    {
        for (j=1; j<=i; j++)
        {
            Printf("*");
        }
    Printf("\n");
    }
    return 0;
}
```

output –

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

(iii) /* Programe for Pattern printing */

```
#include<stdio.h>
int main()
{ int i,j;
  for(i=1; i<=1; i++)
  {
    for (j=5; j>=i; j--)
```

```
{ int i, j;
  for (i=5; i>=1; i++)
  { for (j=1; j<=i; j++)
    { Printf("*");
    }
    Print(" \n");
for (i=1; i<=1; i++)
  { for(j=5; j>=i; j--)
```

```
    {
      Printf("*");
    }
  Printf("\n");
}
return 0;
}
```

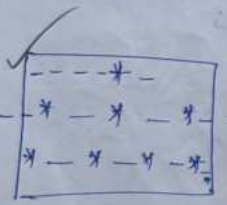
(2) /* to check armstrong no */

```
#include <stdio.h>
int main()
{ int n, K, sum=0, rev;
  K = n; enter the value of n\n");
  scanf("%d", &n);
  while (n!=0)
  {
    rem = n%10;
    sum = sum+(rem*rem*rem);
    n = n/10
  }
  if (K==sum)
    Printf(" it is an armstrong
  else
    Printf(" it is
```

return 0;
}

(1) 

```
for(i=0; i<3; i++)
{
    for(j=0; j<2-i; j++)
        Printf("  ");
    for(k=0; k<2*i+1; k++)
        Printf("*");
    Printf("\n");
}
```



```
for(i=0; i<3; i++)
{
    for(j=0; j<2*(2-i); j++)
        Printf(" ");
    for(k=0; k<2*i+1; k++)
        Printf("*");
    Printf("\n");
}
```

without Space

```
for(i=0; i<3; i++)
{
    for(j=0; j<2-i; j++)
        printf(" ");
    for(k=0; k<2*i+1; k++)
        printf("*");
    printf("\n");
}
for(i=2; i>0; i--)
{
    for(j=0; j<3-i; j++)
        Printf(" ");
    for(k=0; k<2*i+1; k++)
        Printf("*");
    Printf("*\n");
}
```

The return type of Printf (integer)

```
Printf("%d", printf("Alish"));
```

~~output~~ return Alish 5 [∵ A+l+i+s+h = 5]

• Function

- prototype declaration ~~return type~~
- definition
- calling

return type f-name (arguments);

int Prime (int);

more than 1 int
int Prime (int, int);

int ~~Prime~~ even (int x)
{
if (x % 2 == 0)
return 2;
else
return 1;
}

```
#include <stdio.h>
int even (int); → Prototype declaration
int main()
{
    int n, sum = 0, i;
    printf("enter the value of n");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    { if (i%2 == 0)
        printf(
```

```
{
    if (even(i) == 2)        (actual Parameter, calling)
        sum = sum + i;
}
return 0;
}
int even (int x)             (formal parameter)
{
if (x%2 == 0)
    return 2;                (definition)
else
    return 1;
}
```

Utility memory

Library function

① To where the definition of ~~hide~~ ~~header~~ ~~file~~ ~~with~~ stores? And which one relate it with the Programme.

② study about all library functions?

Ans Library functions are built-in functions that are grouped together and placed in a common location called library. Each function here performs a specific operation. We can use this library functions to get the pre-defined output.

```
//* armstrong no. using function *//
#include <stdio.h>
int length (int);
int power (int, int);
int main ( )
{
  int n, sum=0, n1, l
    Printf (" Enter a value");
    Scanf ("%d", &n);
    l = length (n);
    n1=n
  while (n!=0)
    { d = n%10
  sum = sum+Power(d, l);
  n = n/10;
  }
if (sum == n1)
    Printf ("armstrong");
  else
    Print ("not armstrong");
  return 0;
}
int length (int x)
{ int c = 0;
  while (x>0)
  {
    int c = 0;
  }
    c++;
    x = x/10;
  }
  return c;
}
int Power (int x, int y)
{
  int mul=1, i;
      for(i=1; i<=y; i++)
        mul = mul * x;
    return m;
}
```

• Library functions in Diff. header files –

C Header files	Description
<assert.h>	Program assertion functions
<ctype.h>	character type "
<locale.h>	Localization functions
<math.h>	mathematics "
<setjmp.h>	Jump "
<signal.h>	signal handling "
<stdio.h>	standard input/output
<stdlib.h>	standard utility "
<string.h>	string handling "
<stdarg.h>	variable arguments handling "
<time.h>	Date time "

Book: Thaleja [C-Programme]

19/4/22

next class:
- Scope of variable
- storage
- macro define

Which one can passes through an argument → All data types (~~Primary~~ Primary and secondary datatypes)

[A func which calling itself with different values] [remaining topic]

Recursion (calling same function again and again)

```
int main()
{
    inc(5);
}

int inc(int x)
{
    doc(10);
}

void doc(int y)
{
    Printf("5")
}
```

Recursion

① /* factorial */

```
#include <stdio.h>
int main()
{
  int i, n, m = 1;
  Printf("enter the value of n");
  Scanf("%d", &n);
  for(i=1; i<=n; i++)
      m=m*i
  Printf("%d", m);
}
return 0;
```

② /* factorial of any no. */

```
#include <stdio.h>
int fact(int);
int main()
{
  int n, p
  printf("enter a no");
  Scanf("%d", &n);
  p = fact(n);
  Printf("result: %d", p);
  return 0;
}
```

5*fact(4) → 4*fact(3) → 3*fact(2) → 2*fact(1)

recursion

```
int fact(int x)
{
  if (x==0 || x==1)
      return 1;
  else
      return x * fact(x-1);
}
```

~~for~~ addition (first n no.)

(definition)

```
int fact(int x)
{
  if (x==0 || x==1)
      return 1;
  else
      return x * fact(x-1)
}
```

memory — stack (LIFO structure) (last in first structure)

Programme

① Fibonacci series using recursion.

② All Programmes by using recursion (recursion)

① Ans The fibonacci numbers are the numbers in the following integer sequence.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ...

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation.

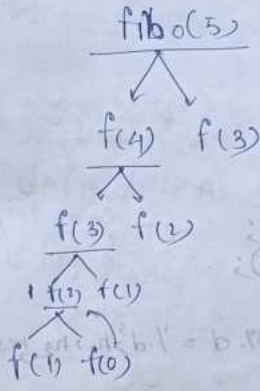
$F_n = F_{n-1} + F_{n-2}$

22/4/22

① Fibonacci series using recursion –

0 1 1 2 3 5 8 13

```
int fibo(n)
{
    if (n==0 || n==1)
        return n;
    else
        return (fibo(n-1)+fibo(n-2));
}
```



② How to calculate gcd using ~~fun~~ recursion and don't using recursion.

without using recursion

```
#include <stdio.h>
int main()
{
    int a, b, a1, b1, ~~lcm~~;
    printf("enter the value of a and b\n");
    scanf("%d %d", &a, &b);
    a1 = a;
    b1 = b;
    while (a != b)
    {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
    printf("the gcd is %d", a);
    ~~lcm~~ return 0;
}
```

Using recursion,

```
#include<stdio.h>
int gcd (int);
int main()
{ int n1, n2, result;
   Printf("enter two no");
   Scanf("%d %d, &n1, &n2");
   result = gcd(n1, n2);
   Printf("gcd of %d and %d = %d", n1, n2, result);
}
int gcd (int a, int b)
{ if (b!=0)
  return gcd (b, a%b) → else
                        return a;
}
```

- Normal recursion (factorial calculation)
- Tail recursion

Hint -

```
int main ()
{
   d = fact(n,1);
}
```

```
int fact (int x, int result);
{
   if(x==0||x==1)
      return result;
   else
      fact(x-1, x*result);
}
```

fact (5, 1)

↓

fact (4, 5*1)

↓

fact(3, 4*5)

↓ (4*5*3)

fact (2, 3*20)

↓ 4*5*3*2

fact (1, 2*60)

↓

← it will return here (tail recursion)

return result = (5*4*3*2)

- ~~Array~~ macro

```
# include < stdio.h >
# define MAX 50
int main ()
{ int x;
   if (x == MAX)
   {
   }
}
```

using function,

```
# include <stdio.h>
# define AREA(x) (x*x) (This is decleration, no need to Prototype decleration)
int main ()
{
   P = macro(5).  [compiler will see → P = 5*5;]
   (4)/(4*4) → another
}
```

(HOD) Prog.c (HLL) → Preprocessor expansion — directive

↓ Preprocessor → macro extension

Prog.I (extended to HLL)

↓ Compiler

Prog.asm

[asm code - assembly level code]

↓

(micro Processor)

↓ Assembler

Prog . obj

↓ Loder

Prog . exe

↓

Process

Linker → [Prog. obj + library]

(LLL) → Lower level language

[obj file — object file]

[Compiler → higher level → machine level]

[exe = executable file]

[1- bit
8-bit = 1 Byte]

[Bit = Binary digit]

char - 1 Byte
int - 2 Byte
float - 4 Byte
double - 8 Byte

[int x;
Print ("%d", size of (x));]

Range
[$0 \to (2^8 - 1)$]
[$0 \to (2^{16} - 1)$]
[$0 \to (2^{32} - 1)$]
[$0 \to (2^{64} - 1)$]

Size of data types in the Compilers we used.

25/4/22

① Short note for switch, Case

② What is the rule of break in switch case.

8 bit = 1 Byte

unsigned - $0 \to (2^8 - 1)$

signed $-2^7 + 0 + (2^7 - 1)$

int → signed, unsigned

unsigned [$0 \to (2^{32} - 1)$]

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

char → signed, unsigned

signed

[-2^7 to $+(2^7 - 1)$]

int → short, long

{
long signed int x;
}

• Storage class ***

(scope – (memory) store

auto

```
int main()
{
  auto int n=5;
  printf("%d", n);
}
```

[default value: garbage value]

```
int main()
{
auto int n=5;
  {
   auto int n=10;
   printf("%d", n);
  }
  printf("%d", n);
}
```

```
int main()
{ inc();
  inc();
  inc();
}
void inc()
{ auto int n=5
  printf("%d", n);
  n++
}
```

output

5
5
5

(register) store

register

```
int main()
{
register int n=6;
printf("%d", n);
```

static (stack)

Static variable take the value '0' by default

extern → Keywords (global)

```
extern int n=5
int main()
{ inc()
}
void inc()
{ printf("%d", n);
}
```

output

5

```
int main()
{
  inc();
  inc();
  inc();
}
void inc()
{
  static int n = 5;
  printf("%d", n);
  n++;
}
```

output

5
6
7

```
# include <stdio.h>
int a=50
int main()
# define a 50
```

```
extern int n=5;
• int main()
  { int n= 3;
  printf("%d", n);
  }
void inc()
{
  printf("%d", n);
}
```

output

~~3~~ 5

The difference between declaration of macro and global.

• Array → for array declaration we need continuous memory
[collections of similar data types]
[what type of data we want to store is the datatype of array]

function is derived data types / users define d data type

```
int m[10];
```

size of the array memory → 20 bytes
(1 int size = 4
5 int size = 20)

```
int m[5] = {1,5,9,3,2};
```
(initialization) [when we assign the values]

```
char c[10];
```
→ size of memory → 10 Bytes

• Contiguous memory Allocation

```
#include <stdio.h>
int main()
{
    int cm[50];i;sum=0;float avg;
    printf("enter the no of all students");
    scanf(
    for (i=0; i<50; i++)
    scanf("%d", &cm[i]);
    for (i=0; i<50; i++)
        sum = sum + m[i];
    avg = (float) sum/50;
    printf("%f", avg);
} return 0;
```

[The 1st index of array is 0]
[last index (n-1)]

• max value among 50 values

```
int main()
{
    int cm[50], i;
    printf("enter the no of all students\n");
    for (i=0; i<50; i++)
    scanf("%d", &cm[i]);
for (i=0; i<50; i++)
    max = cm[0];
    for(i=1; i<50; i++)
    {
        if (cm[i] > max)
            max = cm[i];
    }
```

• same logic for executing min value among 50 value.

```
{ int i, n, min;
    printf("enter the value of n\n");
    scanf("%d", &n);
    int a[n];
    printf("enter the values of array\n");
    for(i=0; i<n; i++)
    { scanf("%d", &a[i]);
    }
```

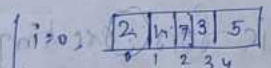
```
min = a[0];
for (i=0; i<n; i++)
{ a[i+1] < min;
    min = a[i];
}
printf ("the min value :%d", min);
return 0;
}
```

11/5/22

How to short an arry:

```
/* asendening array*/
# include <stdio.h>
int main
{ int i, n, j, tmp;

int a[n];
for (i=0; i<n; i++)
  for (j=i+1; j<n; j++)
  { if (a[i] > a[j])
    { temp = a[i];
      a[i] = a[j];
      a[j] = tmp;
    } } }
```

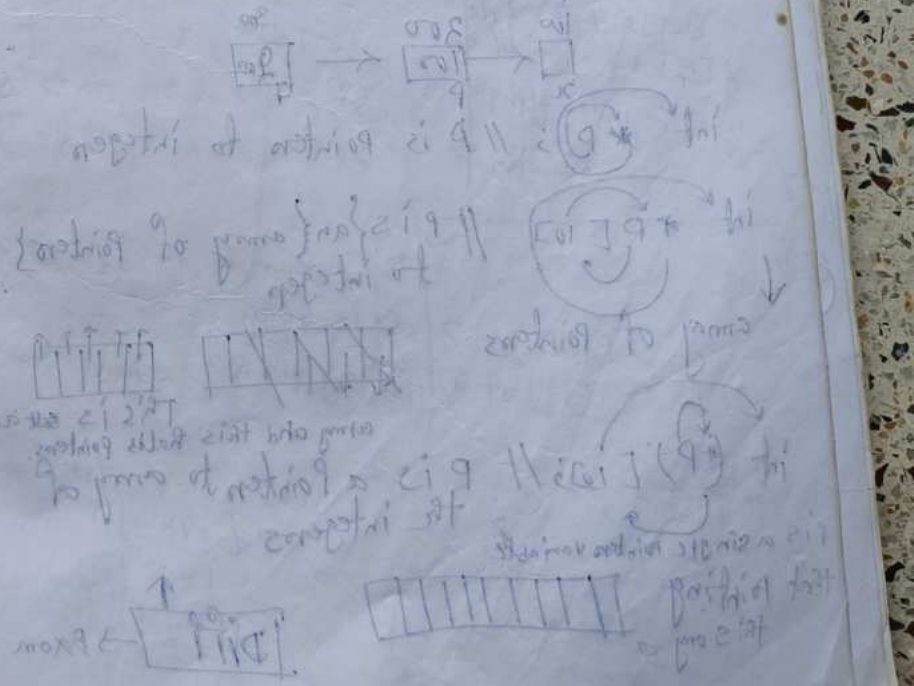



```
for(i=0; i<n; i++)
{printf("%d", a[i]);
}
```

How to do swaping of 2 variables —
(using 3rd variable)

```
x = 10;
y = 5
int tmp;
tmp = x;
y = x;
x = tmp;
```

How to convert a desimal to binary in array.



13/5/22

Pointers

int *p → value at this position

For Pointing address

p is a pointer variable it holds address, in which the int values exist

p is a pointer variable that pointing integer

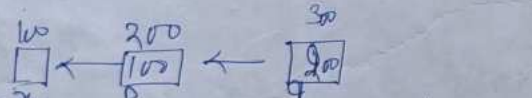
Printf("%u", &x);

```
int *p;
int x=5;
p=&x;
Printf("%d", *p);
```

```
int *p;
int x=5;
p=&x;
*p=2;
Printf("%d", *p);
```

```
int *p;
int x=5;
```

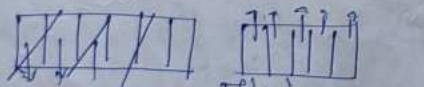
int **q; = Pointer to Pointer



int *p; // P is Pointer to integer

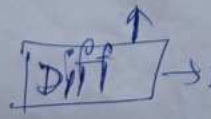
int *p[10] // P is {an} array of Pointers to integer

array of Pointers



int (*p)[10]; // P is a Pointer to array of the integers

P is a single pointer variable that Pointing this array



→ Exam

3m

• what is array pointer? Define with an ex:

```
int main()
{
    int x[] = {5, 4, 45, 7};
① int *p = &x[0];
② Printf("%u\n", p);
③ printf("%d\n", *p);
return 0;
}
```

```
int main
{
    int x=5;
    int *p=&x;
    int **q;
    q=&p;
    Printf("%u\n", p);
    Printf("%d\n", *p);
    Printf("%u\n", q);
```

```
int *p=&x[1];
int **q=&x[3];
Printf("%d", *q-q);
return 0;
```

④ *q=*q+2
⑤ Printf("%d", *q);

⑦ *q=*q*2
⑤ Printf("%d", *q);

Printf ("%d\n", *q);
Printf ("%d", q);
return 0;
}

int main()
{
int x = 5;
int *p = &x;
int y = 6;
int *q = &y;
p = &y;
*p = *q;
Printf ("%u", p);
Printf ("%d", *p);
Printf ("%u", q);
Printf ("%d", *q);
return 0;
}

*p = *q → remove that line

[&a[0] = a
&a[3], &a[0+3]]

#include <stdio.h>
int swap(int x, int y)
{ int temp;

x = y;
y = temp;
Printf ("In swap function x: %d y: %d");
}
int main()
{ int x = 10;
int y = 5;
swap (x, y);
Printf ("After swapping in main function x: %d y: %d", x, y);
return 0;
}

Previous one

t = *x;
*x = *y;
*y = t;

int swap(int *x, int *y);

swap (&x, &y);

diff between Call by value and call by reference.

original data can't be changed, Can get more than 1 return type

16/5/22

string

string - char array
↓
Nothing but a array of characters with a string terminating character

char x[30]; → string
float *b[30]; → string

string ending character → '\0'

```
#include <stdio.h>
int main
{
    char x[45];
    x = "Allah";
    Printf("%c%s", x[2]);
    Printf("%s", x);
    return 0;
}
```

{ char x[6] = "Allah" }
5 + 1 (\0)

output: i

0 1 2 3 4
A l l a h

(m) The ~~type~~ way of writing string (5 ke array)

char x[] = {'A', 'l', 'l', 'a', 'h', '\0'};

From user

```
/* string */
#include <stdio.h>
int main()
{
    char x[45];
    Printf("enter your name: ");
    Scanf("%s", &x);
    Printf("%c\n", x[2]);
    Printf("%s", x);
    return 0;
}
```

[x = Base address]

```
#include <stdio.h>
#include <string.h>
{
    char x[45], y[50];
    Puts/Printf("enter your name:");
    gets(x);
    Puts/Printf("enter your name:");
    ~~Scanf("%s",~~ get(y);
    Printf("%c\n", x[2]);
    Printf("%s", x);
    return 0;
}
```

gets → library function → header file - <stdio.h>

fn string
Scanf → gets
Printf → Puts
For Calculating the length (no. of char) strlen(x);
lo → (count ~~2~~ till \0)
↓ library func <string.h>

```
{
  char x[45], y[50];
  printf("enter your name:");
  gets(x);
  int l = strlen(x);  → [return int]
  printf("length: %d\n", l);
  return 0;
}
```

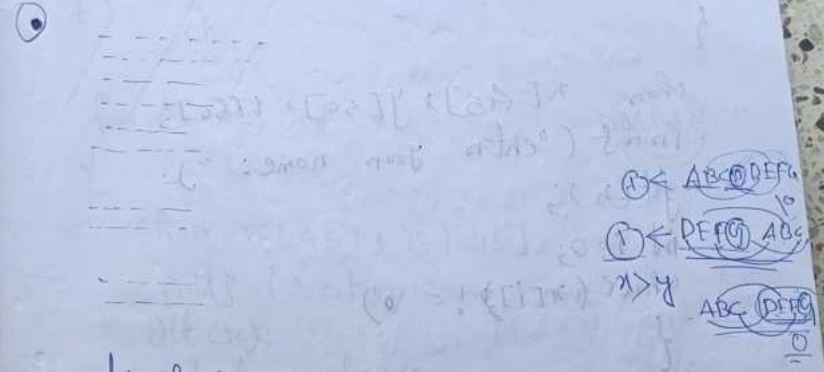
, st[5];

```
strcpy(t, x);  → [return string]
printf("t string is: %s", t);
return 0;
}
```

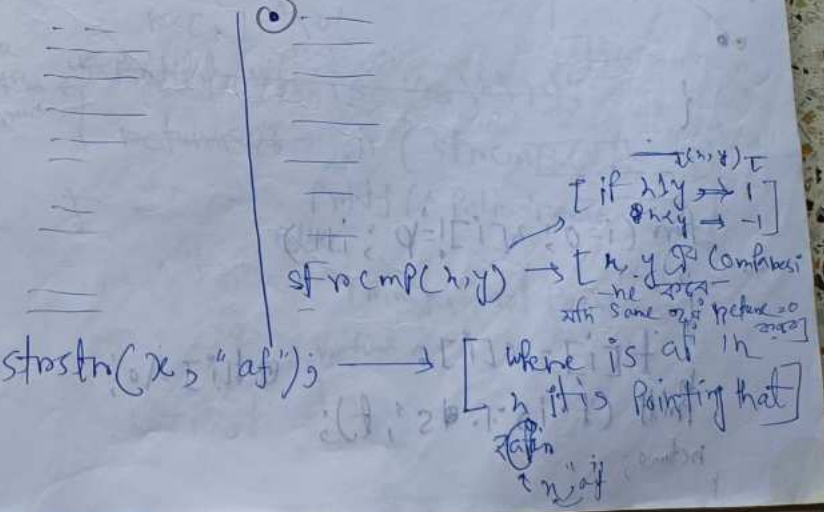
```
printf("enter your name:");
gets(y);
```

```
strcat(t, y);  → [return string]
printf("new t string is: %s", t);
```

```
return 0;
}
```




```
strchr(x, 'a')  → [return → pointer
                   [If is pointing
                    a, it is address
                    which is coming
                    a]
```



```c
#include <stdio.h>
int main()
{
    char x[45], y[50], t[50];
    printf("enter your name: ");
    get(x);
    int i=0;
    while(x[i]!='\0')
    {
        i++;
    }
    printf("length: %d\n", i);
    return 0;
}
```

```c
{
    .....
    .....
for(i=0; x[i]!=p; i++)
{
    t[i]=x[i];
printf("%s", t);
return 0;
}
```

palindrome not (string)

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char x[45], p[45];
    printf("enter your name:");
    get(x);
    int l = strlen(x);
    int i=0;
    while(l>0)
    {
        p[i++] = x[--l];
    }
    p[i]='\0'
    if(strcmp(x,p)==0)
        printf("Palindrome");
    else
        printf("not Palindrome");
    return 0;
}
```

MADAM

l=5
12345
i:0
i:1

MADAM

l---
p[i]=x[l];
i++;

Find vowel and consonant

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

int main()
{
    char x[45];
    printf("enter your name : ");
    gets(x);
    for (i=0; x[i]!='\0'; i++)
    {
        if (x[i]=='a'||x[i]=='e'||x[i]=='i'||
            x[i]=='o'||x[i]=='u')
            v++;
        else
            c++;
    }
    printf("vowels: %d\n", v);
    printf("cons : %d\n", c);
    return 0;
}
```

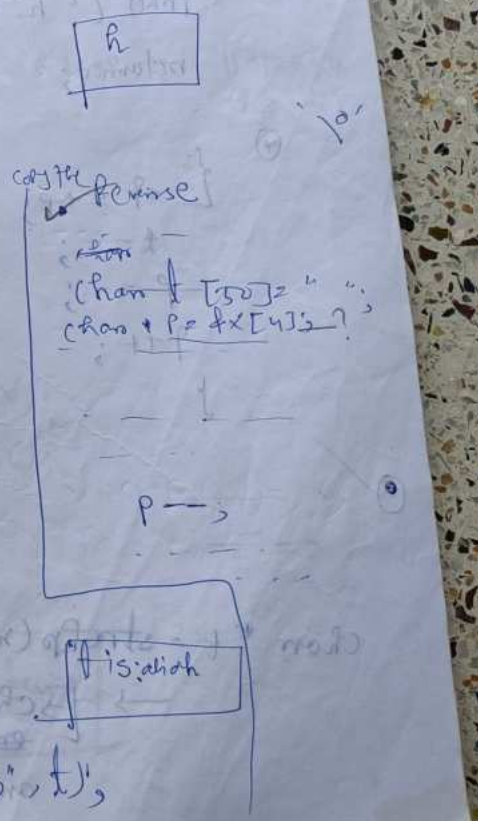
HW distinct no. of vowel and distinct no. of consonant [Rajasree]

String with Pointer

```
#include <stdio.h>
int main()
{
    char x[] = "Aliah";
    char t[50];
    char *p = &x[2];
    char *q = t;
    printf("%c", *(p+2));
    return 0;
}
```

copy the same

```
{
    char *p = x;
    while (*p != '\0')
    {
        *q = *p;
        p++;
        q++;
    }
    *q = '\0';
    printf("t is: %s", t);
```



• Palindrome string and Pointer

```
int l = strlen(s);
while (l >= 0)
{
printf("%c", *(s+l));
l--;
}
if (strcmp(s, t) == 0)
printf("P");
else
printf("NP");
return 0;
```

```
{
*q = *p;
l--;
p++;
q++;
}
```

char *t = strchr(s, 'a');

→ [→ char Pointer → Point the Pointer when Pointer and ...]

• Vowels using Pointer and string.

```
{
char s[45];
printf("enter your name");
gets(s);
char *p = s;
int v=0, c=0;
while (*p != '\0')
{
if (*p=='a' || *p=='e' || *p=='i' || *p=='o' || *p=='u')
v++;
else
c++;
}
printf("vowels: %d", v);
printf("Consonants: %d", c);
return;
}
```

Call by reference = Pointer

```
#include <stdio.h>
int a=15;
int main()
{
    if(a%2==0)
        Printf("even");
    else printf("odd");
    return a=0;20;
}
```

```
#include <stdio.h>
#define a 15
int main()
{
    return 9=20;
}
```

$1\frac{1}{2}$

$\frac{1}{2}$

10

5

20

10

4.4.22

If we will not add . hider file then there will be an errorr and the errorr is "Print sclope is not found"

Conditional statement / Branch :-

if(n;

if (n != 0)
↓
not

```
if (m<40)
   Print f ("fail");

else
  {if (m<50)
      Print f ("D");
   else
     {
      if (m<60)
       Print ("C")
      else
          Print ("A")
          }

     }
```

C

10/04/2022

ALGOL → B → C ✓

```
1  #include <stdio.h>
2  Void main ( )
3  {
4      int a, b;
5      a = 10;
6      b = 20;
7      c = a+b;
8      Printf (" Sum of a and b is", c);
9  }
```

standard input/output header file

stdio.h

Compile ✓

Interpreter ✓

Variable → a, b, c, Sum, ...

i) gabc ✗ → abc_g

ii) abc_cd ✗ ✓

abc_g

↓

Underscore

① Odd and even :—

```
#include <stdio.h>
int main()
{
    int n;
    Printf("Enter a valid no.");
    scanf("%d", &n);
    Printf("you input number %d", n);
    if (n%2==0)
        Printf("Number is even");
        break;
    else
        Printf("Number is odd");
    return 0;
}
```

= a=10
== a==10

n
20 n=5

⑪ For, while, do-while

① ②

```
for (i=1; i<=10; i++)
```

```
#include <stdio.h>
int main()
{
    int a, b, c, d, e;
    a=1;
    b=2;
    c=3;
    d=4;
    e=5;
    Printf("a");
    .
    .
    Printf(e);
}
```

```
#include<stdio.h>
int main()
{
    int i, n;
    Printf("Enter Value of n\n");
    scanf("%d", &n);          → n=5
    for(i=0; i<n; i++)
    {
        Printf("The value is %d", i);
    }
    return 0;
}
```

Dry Run

0	1	2	3	4

i = 0
i = 1
i = 2
i = 3
i = 4
i = 5

① Print all even no. between 1 to 20

```
#include<stdio.h>
int main()
{
    int i, n;
    Printf("Enter Value of n");
    scanf("%d", &n);
    for(i=1; i<=n; i++)          ++i
    {
        if(i%2==0)
            Printf("The number %d is even\n", i);
    }
    return 0;
}
```

\n → New line

output

i = 1
i = 2

2 4 6 8 10 12 14 16 18

2
4
6
8

② Print Sum of all even no From 1 to 20

```
#include <stdio.h>
int main()
{
    int i, n, Sum=0;
    Printf("Enter value of n \n");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        if(i%2==0)
        * Sum = Sum + i;
    }
    Printf("Sum is %d", Sum);
    return 0;
}
```

Sum = 0+2 = 2
Sum = 2+4 = 6
Sum = 6+6 = 12
Sum = 12+8 = 20

Prime no

4 → no 17 → yes
1, 2, 4 1, 17

(7) = (3) 5

~~1~~ 2 3 4 5 6 7
~~1~~ 2 3 5 7

```
#include <stdio.h>
int main()
{
    int i, n;
    Printf("Enter value of n \n");
    scanf("%d", &n);  → (17)
    for(i=2; i<=(n/2); i++)
    {
        if(n%i==0)
            Printf("not prime");
        else
            Printf("Prime");
    }
    return 0;
}
```

2, 3, 4, 5, 6, 7, 8

i = 2 (8)

6 → 1, 2, 3, 6

8 → 1, 2, 4, 8

```
#include <stdio.h>
int main()
{
    int i, n, Sum = 0;
    Printf("Enter a no \n");
    scanf("%d", &n);
    for (i = 1; i < n; i++)
    {
        if (n % i == 0)
            Sum = Sum + i;
    }
    if (n == Sum)
        Printf("Perfect no");
    else
        Printf("not perfect no");
```

1/10 d = 0 ✗

120 % 10

121/10 = 12 ✗

12/10 = 1 ✗

Sum = 0+1 = 1

Sum = 1+2 = 3

Sum = 3+3 = 6

```
    return 0;
}
```

3

Palindrome

```
#include <stdio.h>
int main()
{
    int n, rem, rev = 0, k;
    Printf("Enter Value of n \n");
    scanf("%d", &n);
    k = n;
    while (n != 0)
    {
        rem = n % 10;
        rev = rev * 10 + rem;
        n = n / 10;
    }
    if (k == rev)
        Printf("palindrom");
    else
        Printf("not palindeme");
    return 0;
}
```

121

rem	rev	n
	0	
1	0*10+1 = 1	12
2	1*10+2 = 12	1
1	12*10+1 = 121	0

!= → not equal to

→ 121 == 121

① Prime no using Flag Variable.

```
#include <stdio.h>
int main()
{
  int i, n, flag=0;
  Printf("Enter Value of n\n");
  Scanf("%d", &n);
  for(i=2; i<=(n/2); i++)
  {
    if(n%i==0)
      flag=0;
    else
      Flag=1;
  }
if(flag==0)
  Printf("not prime");
else Printf("Prime");
return 0;
}
```

* break
Continue

* Programming in ANSI C
E. balaguruSamy
Mc graw Hill Education.

```
int n;
```

float(n)

int → float

int → char ✗

```
i=i+1
```

```
i=10
```

i++ → 11 | 10

++i → 10 | 11

i=10

(i++) + (++i)

10 + 12 = 22

x=5, y=4

(x++) + (++y)

5 + 5

= 10

i=11

i++

i = i+1

(10)

x=6

Diff. between do-while loop and while loop.

```
do
{
   ...
}while(Condition);
```

```
while(Condition)
{
   ...
   increment
}
```

153 371

153 = $1^3+5^3+3^3$

= 1 + 125 + 27

= 1 + 152

= 153

153/10

153 n = 153

```
int Sum = 0; K; K = n;
while(n!=0)
{
   rem = n%10;
   Sum = Sum + (rem*rem*rem);
   n = n/10;
}
if(K==Sum)
   Printf("-----");
else
   Printf("-----");
```

1/10 = 0

n (153)	rem	Sum
	3	Sum = 0+27 = 27
15	5	Sum = 27+125 = 152
1	1	Sum = 152+1 = 153
0		

```
#include <stdio.h>
int main()
{
    int i, j, n;
    Printf("Enter a no");
    scanf("%d", &n);
    for(i=1; i<=n; i++)  → outer loop
    {
        printf("\n")
        for(j=1; j<=i; j++) → inner loop
        {
            Printf("*");
        }
        Printf("  ");
    }
    return 0;
}
```

→ *
→ * *
→ * * *
→ * * * *

n=4

i	j	output
1	1	*
2	1,2	* *
3	1,2,3	* * *
4	1,2,3,4	* * * *

Dry Run n=4

i	j	Print
0	0-4	1 1 1 1
1	0-3	2 2 2
2	0-2	3 3
3	0-1	4
4	×	×

c=1
c=2
c=3 c=4

int a=2;

int a;
a = 3/2*5
= 1*5
= 5

int 5a = 3; ✗

AB CD = 3 ✗
AB_CD = 3 ✓

A → 65
B → 66
C → 67

a → 97
b → 98
c → 99

3.142e8

3.142 × 10^8

return x * fact(x-1);

x=5

x=5?

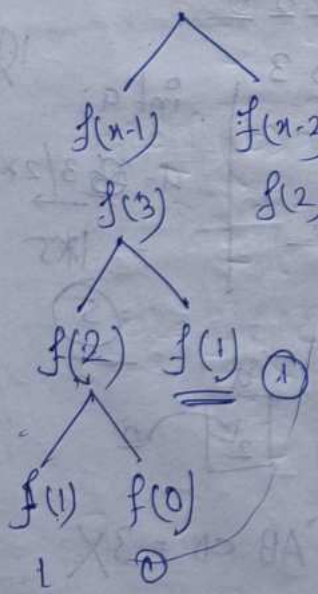
f(x-1) + f(x-2)

f(4) + f(3)

5x f(4)

5x24

120




```
int fact(int x)
{
    int fact = 1;
    ~~fact = fact *~~
    for(int i = 1; i<=x; i++)
    {
        fact = fact * i;
    }
    return fact;
}
```

5

i	fact
1	1
2	2
3	6
4	24
5	120

```
#include <stdio.h>
int main()
{
    int a, b, c;
    Printf("Enter Value of
            a, b, c\n");
    Scanf("%d %d %d", &a, &b, &c);
    if (a>b && a>c)
        Printf("Largest Value among a, b, c. is
                a");
    else if (b>a && b>c)
        Printf("Largest ---------");
    else if (c>a && c>b)
        Printf("---");
}
```

```
&&
10, 11, 12
 a   b   c
a>b && a>c
b>c && b>a
c>a && c>b
```

```
#include <stdio.h>
int main()
{
    int a, b, k;
    Printf("Enter Value of a and b\n");
    Scanf("%d %d", &a, &b);
    while(a!=b)
    if (a>b)
        k=a;
        a = a-b;
    else
        b = b-a;
}
Printf("Gcd of %d and %d is %d", k, b);
return 0;
}
```

(a) 10, (b) 12

a	b
10	2
8	2
6	2
4	2
(2) Ans	2

```c
#include <stdio.h>
int main()
{
    int a, b, a1, b1, Lcm;
    Printf("Enter value of a and b\n");
    Scanf("%d%d", &a, &b);
    a1 = a;
    b1 = b;
    while (a1 != b1)
    {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
    Printf("The gcd is %d", a);
    Lcm = (a1 * b1)/a;
    return 0;
}
```

Printf("....", lcm);

1st no X 2nd no = GCD X LCM

```c
#include <stdio.h>
int main()
{
    int n, k;
    Printf("Enter value of n\n");
    Scanf("%d", &n);
    k = even(n);
    if (k == 2)
        Printf("even");
    else if (k == 1)
        Printf("odd");
    return 0;
}
int even(int x)
{
    if (x % 2 == 0)
        return 2;
    else
        return 1;
}
```

```
#include <stdio.h>
int Armstrong(int x);
int main()
{
    int n,K;
    Printf(" Enter a number\n");
    Scanf("%d",&n);
    K = Armstrong(n);
    if(K==n)
      Printf(" Armstrong no");
    else Printf(" not Armstrong no");
}
int Armstrong(int x)
{
    int rem, Sum=0;
    while(x!=0)
    {
        rem = x%10;
        Sum = Sum+(rem*rem*rem);
        x = x/10;
    }
return Sum; }
```

```
#include <stdio.h>
int main()  ->  int Armstrong(int x);
                int Power(int y);
{
    int n,K;
    Printf(" Enter Value of n\n");
    Scanf("%d",&n);
    k = Armstrong(n);
    if(K==n)
      Printf(" Armstrong no");
    else
      Printf(" not Armstrong no");
}
int Armstrong(int x)
{
    int rem, Sum=0;
    while(x!=0)
    {
        rem = x%10;
        Sum = Sum+Power(rem);
        x = x/10;
    }
```

return sum;

}

int power(int y)

{ int c;

c = (y*y*y);

return c;

}

100 student Maths no store

Array → Similar data types

5, A, % → char, Sp. char

Number

```
int a[10] = { 10, 20, 30, 40, 50, ---, };
```

```
int a[];
int a[10];
```

int a[] = { ---- }

Index →	0	1	2	3	4	5	6	7	8	9
	10	20	30	40	50	60	70	80	90	100
	2000	2002	2004	2006						

a[0] = 10

a[8] = 90

```
#include <stdio.h>
int main()
{
    int i, n, sum = 0;
    int a[n];
    printf("Enter Value of n \n");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
```

```
for(i=0; i<n; i++)
  {
     sum = sum + a[i];
  }
Printf(sum);
}
```

Maximum value among 10 numbers.

```
#include <stdio.h>
int main()
{
  int i, n, max;
    Printf("Enter length of array\n");
    Scanf("%d", &n);
  int a[n];
  for(i=0; i<n; i++)
    {
      Scanf("%d", &a[i]);
    }
```

```
max = a[0];
for(i=0; i<n; i++)
  {
    if(max > a[i+1])
      max = a[i];
  }
Printf(max);
}
```

a[n]

0	1	2	3	4	5	6
10	20	5	6	30	70	1

10 > 20 ✗ 1 ✗

max = m[5]

```
if (b != 0)
    return gcd(b, a%b);
else
    return a;
```

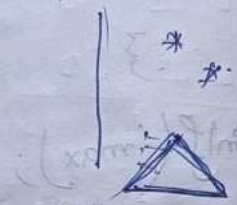
(2)

a, b
12 10
10 12

10 / 12

n/2

n=3



$2^n - 1$

* Linear Search using array

0	1	2	3	4
10	20	15	5	9

```
#include <stdio.h>
int main()
{
    int i, n, c;
    Printf("enter value of n\n");
    scanf("%d", &n);
    int a[n];
```

Search
- Linear Search
- Binary Search

Sorting
- * Radix Sort
- Bubble Sort
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort

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

(*)

```
for(i=0; i<n; i++)
{
    if(c == a[i])
        Printf("%d", i);
}
return 0;
}
```

0	1	2	3	4
10	20	30	40	50

c = 40

i = 0 ✗
i = 1 ✗
i = 2 ✗
i = 3 ✓

(3) Ans

(*) Printf("Enter Value for Search\n");
scanf("%d", &c);

Single line

//.....

multi line

/* .....
..... */

/** .....
..... */

```
#include <stdio.h>
int Palindrome(int x);
int main()
{
    int n, k;
    Printf("Enter Value of n\n");
    scanf("%d", &n);
    k = palindrome(n);
    if (k==n)
        Printf("Palindrome\n");
    else
        Printf("not palindrome\n");
} return 0;
```

```
int Palindrome(int x)
{
    int rem, rev=0;
    while(x!=0)
    {
        rem = x%10;
        rev = rev*10 + rem;
        x = x/10;
    }
    return rev;
}
```

a	10 2020
b	20 2022
c	30 2024

int *a

× clrscr()
× #include<conio.h>
× getch()

* Number of 0's within a number.

```
#include<stdio.h>
int main()
{
    int n, rem, Count=0;
    Printf("Enter a number\n");
    Scanf("%d", &n);
```

```
while (n!=0)
{
    rem = n%10;
    if (rem==0)
        Count = Count+1;
    n = n/10;
}
Printf ("%d", Count);
}
```

1020

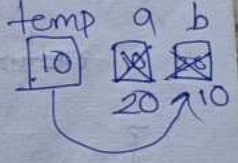
Output

	rem	n	Count
①	5	1020	0
②	5	102	0
③	0	102	1
④	2	10	1
⑤	0	1	2
⑥	1	—	②

Swap using temp Variable

```
#include <stdio.h>
int main()
{
    int a, b, temp;
    Printf("Enter Value of a and b\n");
    scanf("%d %d", &a, &b);
    temp = a;
    a = b;
    b = temp;
    Printf("%d %d", a, b);
    return 0;
}
```

a=10 b=20
b=10 a=20



a=10
b=20

output

temp	a	b
10		

Swap without third variable

```
#include <stdio.h>
int main()
{
    int a, b;
    Printf(" Enter value of a and b \n");
    Scanf("%d %d", &a, &b);
    a = a+b;
    b = a-b;
    a = a-b;
    Printf("... -");
    return 0;
}
```

a = 10 b = 20
a = 30
b = 30-20 = 10
a = 30-10 = 20

```
#include <stdio.h>
#define a 20
int main()
{
    int b, c;
    b = 10;
    c = a+b;
    a = b;
    Printf("%d%d", c, a);
}
```

```
#include <stdio.h>
int a = 20
int b = 10
int main()
{
}
```

x = g

g - 9x9/8

g - 81/8

g - 1 = 8 X

8 | 81 | 10 X
  8
  10

8 - 8x9/8

8 - 9 = -1 X

x - 8x9/8

g - g = 0 X

2 bit (0,1)

1 ~~byte = 8bits~~ 1 byte = 8 bits

1024 ~~bits~~ byte = 1 KB

1024 KB = 1 MB

1024 MB = 1 GB

1024 GB = 1 TB

(*)

i += 2	Prod *= i+7
i = i+2	
i *= 2	Prod = Prod * (i+7)
i = i * 2	Prod = 10 * (3+7)
	= 100

10 3

i += 2	→ i = i+2
i *= 2	→ i = i*2
i -= 2	→ i = i-2
i /= 2	→ i = i/2
i %= 2	→ i = i%2

a = 10 b = 20

while (a>b) ✓

{

}

do ✓

{

}

while (a>b);

a → 10

X 0

X 1

0	1	2	3	4
10	20	30	40	50

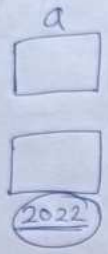
World

0	1	2	3	4	5	6
W	o	r	l	d	'\0'	

String Ending

for(i=0; i!='10'; i++)

int a;



*a = &i;

*a = *(2022) Pointer of Pointer

- value

```
#include <stdio.h>
int Swap(int *x, int *y);
int main()
{
    int a, b;
    Printf("Enter value of a and b \n");
    Scanf("%d%d", &a, &b);
    Swap(a, b);
    return 0;
}
```

```
int Swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
    Printf("%d%d", x, y);
}
```

menu driven

```
#include <stdio.h>
int main()
{
    int k;
    Printf("Enter your option\n");
    Printf("Press 1 for odd even check\n");
    Printf("Press 2 for Prime check\n");
    Printf("Press 3 for Swap\n");
    Scanf("%d", &k)
Switch(k)
{
    Case 1:
        int n;
    Printf("Enter value of n\n");
    Scanf("%d", &n);
    if (n%2 == 0)
        Printf("even");
    else Printf("odd");
```

```
Case 2:
    int n1, i, flag = 0;
Printf("Enter value of n\n");
Scanf("%d", &n1);
for(i = 2; i <= (n1/2); i++)
{
    if(n1 % i == 0)
        flag = 0;
    else
        flag 1;
}
if (flag == 1)
    Printf("Prime");
else
    Printf("not prime");
Case 3:
    int a, b, temp;
Printf("Enter value of a and b\n");
Scanf("%d %d", &a, &b);
```

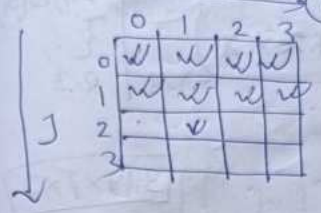
```
Printf("before swapping a=%d and b=%d\n", a,b);
    temp = a;
    a = b;
    b = temp;
Printf(" After swapping a=%d and b=%d\n",
                                    a,b);
 default:
 Printf("your option is wrong\n");
}
return 0;
}
```

* Ternary Operator

if ... else ... Substitute

```
if (a>b)
    Printf("a);
else
    Printf(b);
```

```
K = (a>b) : a ? b;
```



2D Array

1D Array

i = 1

J = 2

```
for(i=0; i<=n; i++)
{
    for(J=0; J<=n; J++)
    {
        scanf("%d", &a[i][J]);
    }
}
```

i	J	a[i][J]
0	0,1,2,3	a[0][0,1,2,3]
1	0,1,2,3	

Sum = Sum + a[i][J]

```c
#include <stdio.h>
#define Area(x) (3.14*x*x)
int main()
{
    int r1, r2, P, Q;
    r1 = 7;
    r2 = 8;
    P = Area(r1);
    Q = Area(r2);
    Printf("%d", P);
    Printf("%d", Q);
    return 0;
}
```

Area(x)
{
r=3

3.14×7×7

3.14×8×8

```c
#include <stdio.h>
int main()
{
    int a=0, b=1, n, c;
    Printf("Enter Value of n\n");
    Scanf("%d", &n);
    Printf("%d\n", a);
    Printf("%d\n", b);
    for(i=2; i<=n; i++)
    {
        c = a+b;
        Printf("%d", c);
        a = b;
        b = c;
    }
    return 0;
}
```

0 1 1 2 3

1

a=1
b=1

a=1
b=2

Strong No using & (1) (24) (120) (145)

$145 = 1! + 4! + 5!$

$= 145$

Sum

fact

```
#include <stdio.h>
int main()
{
    int rem, n, fact=1, Sum=0; Copy
    Printf("Enter value of n\n");
    Scanf("%d", &n); → Copy=n;
    while (n!=0)
    {
        int i=1;
        rem= n%10;
        while (i<=rem)
        {
            fact = fact*i;
            i++;
        }
        Sum= Sum+fact
        n=n/10;
    }
    if (Copy == Sum)
        Printf("Strong no");
    else
        Printf("not Strong no");
}
```

C → Structured Procedure programming Language

Java → object oriented programming Language

class ✓ Object ✓

CAR → Car class

class → 10 → Object

Colour, Size, Company → Object

MAN → Class
Object

G-Map → Class

Address → object.

Laptop → class

Object → Company, Ram, Rom, Colour...

Each program (1) class ✓