

M	T	W	T	F	S	S
Page No.:						
Date:					YOUVA	

Operator - Operator are the symbol which are used to perform operation on Operand.

- Types of Operator in Java.

- 1) Arithmetic Operator

- 2) Assignment Operator

- 3) Relational Operator

- 4) Logical Operator

- 5) Bitwise Operator

- 6) Ternary Operator

- 7) Unary Operator

- 8) Shift Operator

- 9) Instance Operator.

+

|

↓

Addition of
Numeric value

Concatination

of String Value

ex - $10 + 20 \rightarrow 30$

ex - 'java' + 'developer'

↳ javadeveloper.

ex: S.O.P(10 + 20 + "is a num"); // 30 is a num

S.O.P("num is " + 10 + 20); // num is 10 20.

S.O.P("num is " + (30 + 20)); // Num is a 50.

* Java program execution process:

To top
bottom ↓ → Left to Right.

- In a Java Program any operation in Parenthesis () then the Operation will be run first.

* Assignment Operators - These Operators are Used to store right hand Values values into left hand side Variable.

= → Normal Assignment Operator

+ =

- =

* =

} Compound Assignment Operators.

! =

!=

==

!=

* Expression - Combination of Operators and Operand.

$$d += b; \quad // d = d + b$$

$$S.O.P.(d) // 30.$$

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

ex. int i=5;

int j=2;

int k=i*j; // i*j = j*i ✓

ex. int a=3; int b=4;

int c=a+b; X

Because in Unary Variable are in String the result that Variables Should be one of the Operand of that Operator

ex: int a=10; int b=20;

int z=z+a+b; // z+=a+b X

Compound arithmetic operator with an only exactly 2 operators.

* (Unary Operator): (Unary Operator Works on exactly One Operand)

1) Increment Operator (++)

2) Decrement Operator (--)

Increment Operator

Decrement Operator

1) If increments the value by exactly 1. Current Value by 1.

Value by exactly 1.

2) Post increment Operator 2) Post Decrement Syntax:

VariableName++;

VariableName--;

3. pre-increment Operator 2. pre-decrement
 ++ VariableName -- VariableName

$\text{int } a = 10$

$a++;$ $\text{int } a = 0$

$\text{S.O.P}(a);$ $\text{int } b = 0$

$\hookrightarrow \text{int } b = a + 1;$

b = a + 1; $\text{S.O.P}(a) \text{ } 111$

$\text{int } a = 0;$ $\text{S.O.P}(b) \text{ } 110$

$\text{int } a = 0 \text{ } || \text{ } \text{int } r = 0$

$\text{int } b = 0 \text{ } || \text{ } \text{int } y = 0$

$b = a + r;$ $x = y + z;$

$\text{S.O.P}(a) \text{ } 111$ $\text{S.O.P}(x) \text{ } 111$

$\text{S.O.P}(b) \text{ } 110$ $\text{S.O.P}(y) \text{ } 111$

$\text{int } r = 0;$ $\text{int } r = 0, \text{ int } y = 0$

$\text{int } y = 0;$ $y = r + + + x + + + y;$

$y = r + + + z + x;$ $3 \quad 0 \quad 1 \quad 2$

$\text{S.O.P}(r) \text{ } 111$ $\text{S.O.P}(r) \text{ } 112$

$\text{S.O.P}(y) \text{ } 111$ $\text{S.O.P}(y) \text{ } 113$



M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

$x++ \rightarrow$ first we are using current value
after that we are incrementing the x
value.

$++x \rightarrow$ first we are incrementing the current
value of x after that we are using
the x value.

int $x = 0, y = 0;$

$y = x++ + x + x + x + x + x$
S.O.P(x) // 2

S.O.P(y) // 3

int $x = 0, y = 0$

$y = x + x + x + x + x + x + x + x + x + x$

S.O.P(x) // 13
S.O.P(y) // 10

int $x = 0, y = 0$

$y = x + x + x + x + x + x + x + x + x + x$

S.O.P(x) // 5

S.O.P(y) // 14

int $x = 0, y = 0$

$y = x + x + x + x + x + x + x + x + x + x + x + x + x + x + x$

S.O.P(x) // 7

S.O.P(y) // 31

* Relational Operators:

Relational Operator are used to check the relation between Values/Variables.

- Relational Operators are used to Compare the Values and to take the decision.

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

Note: Relational Operators Works on numeric Values also but always the Result is boolean Value.

ex: S.O.P(10 > 20); // false

S.O.P(5 < -20); // true

S.O.P(10 != 20); // true

* Flow Control Statements / Conditional Statement

These are the statement which are used to control the flow of the execution of the program. Types: (i) Decision Making Statement
 (ii) Looping Statement

→ Jumping Statement in Conditional Statement: (i) Continue
 (ii) Break
 (iv) Return

M	T	W	T	F	S
Page No.:					YOUVA
Date:					

* Decision Making Statement

→ Based on the decision / Condition Some Set of Statement gets executed / Skipped.

(i) if Statement

(ii) if else Statement

(iii) if else Ladder

(iv) Nested if

(v) Switch Case Statement

Expression : $a + b$ is a Combination of Operators and Operands.

ex - $a + b$.

Condition - Condition is an expression

which result in boolean value.

ex: $10 > 20 \rightarrow \text{False}$

* if Statement : If the Condition true if body get execute otherwise the body will get skipped.

ex: if (condition) {
 // Statement
}

Ex - Class Test

PSVM C12

S.O.P("Started")

if ($a > 5$) {

S.O.P("Value is 10");

} S.O.P("End");

}

Q. Write a Java Program to find the biggest number among two numbers Using if Statement:

int a = 10, b = 20;

if ($a > b$) {

S.O.P("Value is a");

} if ($b > a$) {

S.O.P("Value is b");

}

Q. Find the biggest Number among three numbers Using if Statement.

int a = 10, b = 20, c = 30;

int big = a;

if ($b > \text{big}$)

 big = b;

if ($c > \text{big}$)

 big = c;

S.O.P("Biggest Number: " + big);

* if - else Statement:

Syntax :- if (Condition / Expression)

{ If Statement

}

else {

If Statement

}

Condition if body else body

1. true Executed Skipped

2. false Skipped Executed

- if we have exactly two Possibility we Should Use if - else Statements.

- Both if - else body never executed depends on the Condition either if body else body get executed.

x The given number is even or odd.

class Test {

 PSVMCL

 int r = 5;

 if (r % 2 == 0) {

 S.O.P("num is Even");

 } else {

 S.O.P("num is Odd");

}

v Check the given number is even or odd without Using Modulo Operator.

Class Test 2

PSVM C15

int n=8;

if (n/2 * 2 == n)

s.o.p('even');

else

s.o.p('odd');

v Check the given Year is Leap Year or not.

Class Test 2

PSVM C15

int Year=2019;

if (Year % 4 == 0 && Year % 100 != 0)
 {
 s.o.p("leap year");
 }

else if (Year % 400 == 0)
 {
 s.o.p("leap year");
 }

else

{
 s.o.p("not leap year");
 }

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

* if-else ladder

Syntax: if c condition1 {

} // statements of if body

else if (Condition2){

} // statements of else if body

else {

} // statements of else body

}

Cond1 Cond2 if body else if body else body

	true	false	true	false	Skipped	Skipped	Execute	Skipped	SKIP
1	true	false	true	false	Skipped	Skipped	Execute	Skipped	skip
2	false	true	false	true	Execute	Execute	Skipped	Execute	skip

Input n = 15:

if (n > 10) {

S.O.P ("above 10");

else if (n < 10) {

S.O.P ("below 10");

else {

S.O.P ("equal");

}

* Logical Operator : Logical Operator Works on boolean Expressions and the result also in boolean Value.

1) Logical AND (&&)

2) Logical OR (||)

3) Logical NOT (!)

◦ Logical AND - Syntax: Expr1 && Expr2

Logical AND (&&) Operator returns true iff all the expression becomes true otherwise it returns false.

Note: When all the condition has to met go for AND(&&) Operator.

exp1 exp2 Result

True False False

False True False

False False False

True True True

ex: S.O.P(10>5 && 10>20); // False

S.O.P(5>2 && 20>15); // True

S.O.P(10+20 && 20+10); // CTE

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

Logical OR (||) : Syntax: Expr1 || Expr2

Logical OR Operator returns true iff
any one of cond/expres becomes true
otherwise it return false.

Note: When any one of condition or
expres. has to be met then go for the
Logical OR (||) Operator.

expr1	expr2	Result
true	false	true
false	true	true
false	false	false
true	true	true

ex: S.o.p(10>5 || 10>20); //true

S.o.p(true || false); // true

S.o.p(10>20 || 20>30); //false.

- Logical NOT (!) : Syntax: ! expres.

- Logical NOT (!) return true iff the cond/expres become false.

- Logical NOT (!) return false iff the cond/expres become false.

→ if we want to make true result becomes false or if we want to make false result becomes true then we should use logical Not (!) operator.

expr1 Result

true false.

false true.

ex: `s.o.p(c!(false));` || true

`s.o.p(c!(true));` || false

`s.o.p(c !(10>5));` || false

Q. write a Java Program to print 'Sanju' if the given number is divisible by 3 or print 'Geeta' if the given number is divisible by 5 or print 'Sanju wede geeta' if the given number divisible by 3 and 5 otherwise print Breakup.

class testS

psvm1 S

int num = 6;

M	T	W	T	F	S	S
Page No.:						
Date:						

```

if (num%3 == 0)
    S.o.p("Saaju")
if (num%5 == 0)
    S.o.p("Geeta");
if (num%5 == 0 && num%3 == 0)
    S.o.p("Saaju wede Geeta")
else
    S.o.p("Breakup");

```

V

```

int n = 15;
if (n%3 == 0 && n%5 == 0)
    S.o.p("Swig");
else if (n%3 == 0)
    S.o.p("Saaju");
else if (n%5 == 0)
    S.o.p("Geeta");
else
    S.o.p("breakup");

```

Write a Java program to find a
Smallest Number among three
number. Using if else ladder.

Java program to find the biggest
number among two number Using
ternary Operator.

```
15 int a = 2, b = 4, c = 5;
if (a < b && a < c) {
    S.O.P(a);
} else if (b < a && b < c) {
    S.O.P(b);
} else {
    S.O.P(c);
}
```

```
16 int a = 10, b = 20;
int big = (a > b) ? a : b;
S.O.P(big);
```

* Ternary Operator

- It works on 3 operands that's why it's called Ternary Operator.
- It is almost same as if-else Statement / it is replacement of if-else Statement.

Syntax:

Resultant Value

true

Resultant = (condition)? true body : false body

Variable

false

Resultant Value

Note: Resultant Value type and Resultant Variable type Should be same.

Q) Find the Smallest Number among three Number Using ternary Operator.

```
int a = 10, b = 20, c = 5;
```

```
int x = (a < b) ? a : b;
```

```
int y = (x < c) ? x : c;
```

```
System.out.println(y);
```

M	T	W	T	F	S	S
Page No.:						
Date:					YOUVA	

→ int a = 10, b = 20, c = 30;

int small = ($a < b \Rightarrow a \& a < c$) ? a :
($b < c$) ? b : c;

System.out.println(small);

Q. Find biggest number among four
numbers Using ternary Operator.

close Text + {

PSVM() {

int a = 2, b = 4, c = 6, d = 8;

int big = ($a > b \& a > c \& a > d$) ? a :
($b > c \& b > d$) ? b : ($c > d$)
? c : d;

System.out.println(big);

} }

* Switch Case Statement

Switch Case Statement is similar to if-else ladder.

→ Switch Statement Works with int, Short, byte, long, String, char and with some wrapper classes like Integer, Byte, Short etc

→ Switch Case Statement won't work with double, float, boolean type value.

Syntax: Switch (Expression)
{

 Case value1:

 //Statement

 break;

 Case value2:

 //Statement

 break;

 default:

 //Statement

}

int day = 2;

switch(day)

{

case 1:

s.o.p("Monday");

break;

case 2:

s.o.p("Tuesday");

break;

default:

s.o.p("invalid output");

}

* Important Points:

- There can be one or N number of case values for a switch expression.
- The Case Value must be of switch expression type only. The case value must be literal or constant. It doesn't allow variables.
- The case values must be unique. In case of duplicate value, it renders compile-time error.

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

- The Java Switch expression must be of byte, short, int, long (with its wrapper type), enum and String.
- Each Case Statement can have a break Statement which is optional. When control reaches to the break Statement.
- It jump the control after the Switch expression. If a break Statement is not found, it executes the next Case.
- The Case value can have a default label which is optional.

Check the given Number is even or odd Using Switch Case Statement

```
int num = 3;
switch (num) {
    case (num % 2 == 0):
        S.o.p("even");
        break;
```

```
    default:
        S.o.p("odd");
        break;
```

```
int n = 10;
switch (n % 2) {
    f.
```

Case 0:

```
    S.o.p("even");
    break;
```

Case 1:

```
    S.o.p("odd");
```

}

A customer has a bank account with the balance of 10000. Write a Java program to withdraw the amount from the account. The withdraw should be successful if the balance is sufficient otherwise withdraw should be failed.

If the withdraw successful show the remain balance.

```
int balance = 10000;
```

```
int amount = 2000;
```

```
if(amount <= balance)
```

```
{
```

```
S.O.P("Transaction success");
```

```
balance = balance - amount
```

```
S.O.P("Remaining bal" + balance -  
amount);
```

```
}
```

```
else
```

```
S.O.P("Failed");
```

```
}
```

* Nested if statement.

Syntax: if (condition 1) {
 body statements
 if (condition 2) {
 body statements
 }

 }

 if (condition 3) {
 body statements
 }

 }

 else {
 body statements
 }

 }

 else {
 body statements
 }

 }

 main else body statements

 }

Main nested if nested if nested else Main else
condition Cond. Body Statement Body Statement Body Statement

if true false Skip Execute Skip

 true true execute Skip Skip

 false false Skip Skip execute.

Execution

if (Condition)
 Body Statement

class test {

 public static void main(String[] args) {

 int pin = 1234; int ePin = 1234;

 double balance = 20000.0;

 double amount = 5000.0;

 if (pin == ePin) {

 if (balance > amount)

 {

 double rem = balance - amount;

 System.out.println("Amount withdrawn");

 }

 } else {

 System.out.println("Inufficient balance");

 } else {

 System.out.println("Incorrect Pin");

 }

* Looping Statement:

If we want to execute some set of statement repeatedly again and again then we should go for looping statement.

(i) for loop

(ii) while loop

(iii) do-while loop

(iv) for-each loop

- Q1. Why should go for looping statement.
- Q2. Diff bit for & for-each.
 - Q3. Diff bit while & do-while.

* For Loop:-

for (initialization; condition; increment)

{

 // Statement

}

Note: we know the exact number of iteration then should use for loop.

* Print all the even no. present in 1 to 20.

→ int num = 20;

for (int i = 1; i <= n; i++) {

 if (i % 2 == 0) {

 S.O.P(i);

}

M	T	W	T	F	S	S
Page No.						YOUVA
Date						

- One complete life cycle of the loop body is called 1 iteration.
- One Complete execution of the loop body is called 1 iteration.

* Write a Java program print the factor number for given Number.

```
int n = 6
for (int i = 1; i <= n; i++)
{
    if (n % i == 0)
        System.out.println(i);
}
```

Q. find the factorial for given Number

```
int num = 5;
int factorial = 1;
for (int i = 1; i <= num; i++)
{
    factorial = factorial * i;
}
System.out.println(factorial);
```

Q. Write a java program to check the user enter number is perfect or not.

→ Some of all the factors excluding itself number should be equal to original number that is considered Perfect Number.

input = 28 O/P: $1+2+7+4=14$

```

int n = 28;
int sum = 0;
for (int i = 1; i < n / 2; i++) {
    if (n % i == 0) {
        sum = sum + i;
    }
}
if (sum == n)
    S.O.P("Perfect no")
else
    S.O.P("not")
  
```

M	T	H	F	S	S
Page No.	YONIYA				
Date					

W $\text{for}(\text{int } i = 1; i <= 5; i++) \quad \text{for}(\text{int } i = 1; i <= 5; i++)$
 { if ($i == 3$) } if ($i == 3$)

continue;

break;

s.o.p(i);

s.o.p(i);

}

}

$\rightarrow 1 \ 2 \ 4 \ 5$

$\rightarrow 1 \ 2$

W $\text{for}(\text{int } i = 1; i <= 5; i++) \quad \text{for}(\text{int } j = 1; j <= 5; j++)$
 { }

$\text{for}(\text{int } j = 1; j <= 4; j++)$

s.o.p(i);

{

if ($j == 2$)

for($j = 1; j <= 5; j++$)

{

continue;

if ($j == 2 \ || \ j == 4$)

}

s.o.p(j);

break;

s.o.p(i);

s.o.p(j);

}

o/p: $\begin{matrix} j \\ 1 & 2 & 1 & 3 & 1 & 4 & 1 & 5 \end{matrix}$

O/p. 1 1 2 1 3 1 4 1 5 1

1 2 3 4 5 1 3 4 1

1 3 4 2

1 3 4 3

1 3 4 4

1 3 4 5

- 1 Count how many digits are present in the User enter Number
- 2 Write a Java program to print the sum of each digit Present in the User enter Number
- 3 Print the product of each digit present in user enter Number.

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

Scanner

It is a pre-defined final class present in `java.util` package.

- Scanner Class is used to take input from user / console / keyboard.
- Scanner Class is provided some input methods to take the input from the user.

- * `nextInt()` - Used to take integer type I/P from User
- `nextByte()` → Byte
- `nextShort()` → short
- `nextLong()` → long
- `nextFloat()` → float
- `nextDouble()` → double
- `nextBoolean()` → Boolean
- `nextLine()` → String
- `next()` → String
- `next(): charAt` → Character type

- ↳ all these are Non-static Method Present in Scanner Class.

```

ex: Scanner sc = new Scanner(System.in);
    sc.nextLine();
    int n = sc.nextInt();
    sc.nextLine();
    double d = sc.nextDouble();
    sc.nextLine();
    sc.close();
  
```

* While loop:

While loop is also a type of looping Statement which is used to execute some set of statement repeatedly.

- If we don't know the number of iteration then we should use while loop.

Syntax: Utilization

while (condition)

{

// Some Stmt

Updation

}

(1)

initialization

true

↓

(2)

↓

while (condition)

false

S

(4)

if Statement

(5)

Updation

→ 7

write a java program print all the odd numbers present between 1 to 20 using while loop.

int i = 0; i :

int i = 1, n = 20;

while (i <= 20) while (i <= 20)

{ }

if (i % 2 == 0)

System.out.println("Even")

else

if (i % 2 != 0) System.out.println("Odd"); i++;

W Close Demo {

PSVM C } {

Scanner sc = new Scanner (System.in);

s.o.p ("Enter Integer Value");

int n = sc.nextInt();

int count = 0;

while (n > 0) {

int rem = n % 10;

count = count + rem;

n = n / 10;

}

s.o.p (count);

10) 1234 (123

$$\begin{array}{r} 1234 \\ \times 10 \\ \hline 1234 \end{array}$$

W int n = 765;

int sum = 0;

while (n > 0) {

int rem = n % 10;

sum = sum + rem;

n = n / 10;

}

s.o.p ("sum is " + sum);

```

1. int n = 123;
   int prod = 1;
   while (n > 0) {
       int rem = n % 10;
       prod = prod * rem;
       n = n / 10;
   }
   S.O.P (prod);

```

- Q. Write a program to check the user enters number is Spy Number or not.

Spy - Sum of each digit & product of each digit be same.

```

int n = 2147;
int sum = 0, prod = 1;
while (n > 0) {
    int rem = n % 10;
    sum = sum + rem;
    prod = prod * rem;
    n = n / 10;
}
if (sum == prod) {
    S.O.P ("num is SPY");
} else S.O.P ("num is not SPY");

```

- Q. Write a program to count how many even digits are present in the given number.
- (Logical)
- Q. write a program to sum of odd no. digit present in the given number.
- (Logical)
- Q. To reverse the given number.

```

W int n = 123456; // input of user
int Count = 0; // output of program
while (n > 0) {
    int rem = n % 10; // remainder
    if (rem % 2 == 0) { // condition
        Count++; // increment
    }
    n = n / 10; // update
}
S.O.P(Count); // output
  
```

```

W int n = 1235; // input of user
int Sum = 0; // output of program
while (n > 0) {
    int rem = n % 10; // remainder
    if (rem % 2 != 0) { // condition
        Sum = Sum + rem; // add
    }
    n = n / 10; // update
}
S.O.P(Sum); // output
  
```

```

int rem=0;
int rev=0;
int n=1234;
while(n>0){
    rem=n%10;
    rev=rev*10+rem;
    n=n/10;
}
s.o.p(rev);

```

Q. Write a Program to Check even digit
Sum is even or not.

int n=1234;

int sum=0;

while(n>0){

int rem=n%10;

if(rem%2==0){

{

sum=sum+rem;

}

} n=n/10;

if(csum%2==0){

s.o.p("even");

else

s.o.p("odd");

Q. Write a program to check the given number is Palindrome or not.

Q. Write a program to check the given number is Neon or not.

W Public class Palindrome {

 sum = 0;

 Scanner sc = new Scanner(System.in);

 int n = sc.nextInt();

 int temp = n; int reverse = 1;

 while (n > 0) {

 int rem = n % 10;

 int reverse = reverse * 10 + rem;

 n = n / 10;

}

 if (reverse == temp) {

 s

 s.out.println("Palindrome");

}

 else {

 s.out.println("not");

}

w Public class Neon {
 Ps sum (String [] args) {
 int num = 0; int sum = 0;
 Scanner sc = new Scanner (System.in);
 int num = sc.nextInt();
 while (num > 0) {
 int rem = num % 10;
 int sq = rem * rem; int sum =
 sum + sq;
 num = num / 10;
 }
 if (sum == num) {
 System.out.println ("Neon Number");
 } else {
 System.out.println ("not a Neon Number");
 }
 }
}

Q. Write a program to find α^p .

```
Scanner sc = new Scanner (System.in);
System.out.print ("Enter");
int n = sc.nextInt();
System.out.print ("Enter power");
int p = sc.nextInt();
```

13

```

int pow = 1;
for (int i = 1; i < p; i++) {
    $ swap
    Pow = Pow * i;
}
    
```

Pow = Pow * n;

S.o.p("Value" + Pow);

3. Prime number or not.

Prime number or not.

Input will be given

and output will be

x Prime Number or not.

int n = 11, count = 0;

For (int i = 1; i <= n; i++) {

if (n % i == 0)

{ count++; }

if (count == 2)

S.o.p("prime");

else

S.o.p("Not");

W

Methods / Functions

Methods are the member of class which are used to perform a particular task.

Syntax: <Access Specifier> <Non-access Modifiers>
 <Return type> <Method Name> <Parameters>

Return type Method Name (<Parameters>)

Methods has two Parts:

(i) Method declaration

(ii) Method definition / implementation

Access Specifiers

(i) Public

(ii) Protected

(iii) Package level
 (default)

(iv) ~~defa~~ Private

Non-access Specifiers

(i) Static

(ii) Final

(iii) Abstract

(iv) Synchronized

(v) Transient

(vi) Volatile

(vii)

(viii) Native

Q. Define a method to count how many factors are present for the given number.

Class demo :-

```
public static void Factor(int r) {
```

```
    int Count = 0;
```

```
    for (int i = 1; i <= r; i++) {
```

```
        if (r % i == 0) {
```

```
            Count++;
```

```
}
```

```
}
```

```
S.O.P(Count);
```

```
PSVM() {
```

```
    int n = 6;
```

```
    Factor(n);
```

```
}
```

```
}
```

Lecture 3

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

* Local variable - If the variable declare inside the function body then it is called Local variable.

↳ Local variables can be access with in the function body only where we have declare.

↳ Local Variables Must be Initialize before Using but ~~any~~ it in any Operation because Local Variables Will Not be having default Value.

* Argument - Arguments are nothing but Values/Variables Which are Passed to the function Parameter.

Q. Neon or not

int n = 123, s[] = { "Neon" }

* Strong Number: Sum of factorial of digits should be equal to original number.

class test {

 PSVNCL {

 Scanner sc = new Scanner(System.in);

 S.O.P("Enter num");

 int n = sc.nextInt();

 } isStrong(n);

 Public static void isStrong(int n)

 {

 int sum = 0, temp = n;

 while (n > 0) {

 int rem = n % 10;

 int fact = factorial(rem);

 sum = sum + fact;

 if (sum == temp)

 }

 n = n / 10; s.o.p("is a strong")

 else

 Static int factorial (int n)

 {

 int fact = 1;

 for (int i = n; i >= 1; i--)

 {

 fact = fact * i;

 return fact;

 }

- Q Print all the prime number between 1 to 100.
- Q. Print sum of all the prime numbers present between 1 to 15.

W Class Demo :

```
PSVM()
```

```
{
```

```
int n = 100;
```

```
for (int i = 2; i <= n; i++)
```

```
{
```

```
    boolean res = isPrime(i);
```

```
    if (res == true)
```

```
        System.out.print(i + " ");
```

```
i++
```

```
}
```

```
private static boolean isPrime(int n)
```

```
{
```

```
if (n <= 1)
```

```
    return false;
```

```
for (int i = 2; i < n / 2; i++)
```

```
{
```

```
    if (n % i == 0)
```

```
        return false;
```

```
    }
```

```
    return true;
```

```
{
```

W Class Demo 5

D SUM()

S

int n = 15; int sum = 0;

for (int i = 2; i < n; i++)

S

boolean res = prime(i);

if (res == true) {

sum = sum + i;

}

System.out.println("Sum = " + sum);

}

private static boolean prime(int n)

{

if (n < 1) return false;

for (int i = 2; i < n / 2; i++)

S

if (n % i == 0)

return false;

}

return true;

}

- Q. WAP to print even/odd using Method
as boolean return type.

```

int n = 14;
boolean res = OddEven(n);
if (res == true)
    S.O.P("even");
else
    S.O.P("odd");
}

public static boolean OddEven(int n)
{
    if (n % 2 == 0)
        return true;
    else
        return false;
}

```

- Q. WAP to Check prime no. Using Method
and Boolean as Return type.

```

public boolean Prime(String target)
{
    int n = 13;
    boolean res = isPrime(n);
    if (res == true)
        S.O.P("prime");
    else
        S.O.P("not prime");
}

```

else

s.o.p("not prime")

{

public static boolean isPrime(int n) {

if (n <= 1) return false;

for (int i = 2; i <= n/2; i++)

{ if (n % i == 0)

return false

{

return true;

{

Q. WAP to define a Method Check the given sum is Armstrong number or not

Scanner Sc = new Scanner (System.in);

s.o.println("Enter the number");

int n = Sc.nextInt();

isArmstrongNumber(n);

{

static void isArmstrongNumber (int n) {

int sum = 0; int cd = CountDigit(n);

int temp = n;

while (n > 0) {

int rem = n % 10;

int Pow = Power(rem, cd);

Sum = sum + pow;

for (i=1; i<n; i++)

{ n = n/10;

if (sum == temp) {

 S.o.p("Armstrong");

 clear();

 S.o.p("Not");

static int power(int n, int p){

 int pow=1; for (int i=1; i<p; i++)

 for (int i=1; i<n; i++)

 pow = pow * i;

 return pow;

static int CountDigit (int n)

{ int count=0; if (n==0) return 0;

 while(n>0)

 int rem = n % 10; if (rem > 0)

 count = count + 1;

 n = n / 10;

 return count;

}

main()

{ int n;

 S.o.p("Enter a number");

* Deceirium Number.

```
int n = 154;
```

```
isDeceiriumNum(n);
```

```
}
```

```
static void isDeceiriumNum(int n) {
```

```
int sum = 0; int cd = CountDigit(n);
```

```
int temp = n;
```

```
while (n > 0) {
```

```
int rem = n % 10;
```

```
int pow = Power(rem, cd);
```

```
sum = sum + pow;
```

```
cd = cd - 1;
```

```
n = n / 10;
```

```
if (sum == temp) { S.o.p("Deceirium"); }
```

```
else { S.o.p("Not"); }
```

```
static int Power(int n, int p) {
```

```
int Pow = 1;
```

```
for (int i = 1; i < p; i++) {
```

```
Pow = Pow * n;
```

```
} return Pow;
```

```
static int CountDigit(int n) {
```

```
int count = 0;
```

```
while (n > 0) {
```

```
int rem = n % 10;
```

```
Count = Count + 1;
```

```
n = n / 10;
```

```
} return Count;
```

```
}
```

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

W WAP to print first N prime numbers.

W Print the Nth Prime number.

Class test :-

```

P S V M ( T - S . . . . . )
int n=5; int n=4, i=2;
for (int i=2; count=0;
while (n>i) {
    while (true) {
        boolean re = isprime(i);
        if (re == true)
            {
                System.out.println(i);
                count++;
            }
        if (count == n)
            break;
        i++;
    }
}

```

Static boolean isprime(int n)

```

for (int i=2; i<n/2; i++)
{
    if (n%i == 0)
        return false;
}
return true;
}

```

class test {

 public static void main (String [] args)

 int n = 4, i = 2, count = 0;

 while (true) {

 boolean r = isprime(i);

 if (r == true)

 {

 count++;

 }

 if (count == n) {

 System.out.println(i);

 break;

 }

 i++;

 static boolean isprime (int n)

 for (int i = 2; i <= n / 2; i++)

 {

 if (n % i == 0)

 return false;

 }

 return true;

}

M	T	W	T	F	S	S
Page No.	YOUVA					
Date:						

WQ. WAP to print nearest prime number.

Class Test

PSVM()

```
int n = 9, p = n - 1, n1 = n + 1;
```

```
while (true) {
```

```
    boolean r = isprime(n);
```

```
    if (r == true) {
```

```
        S.O.P(n);
```

```
        break;
```

}

```
    boolean rs1 = isprime(p);
```

```
    if (rs1 == true) {
```

```
        S.O.P(p);
```

```
        break;
```

}

```
    boolean rs2 = isprime(n1);
```

```
    if (rs2 == true) {
```

```
        S.O.P(n1);
```

```
        break;
```

}

```
    p--; n1++;
```

J

for (int i = 2; i < n1; i++) {

```
    if (n % i == 0)
```

```
        return false;
```

```
static boolean isprime(int n) {
```

```
    for (int i = 2; i < n1; i++) {
```

}

return true;

3 3

3 4 5 6

2 3 4 5 6 7

- Q. WAP to print sum of each digit until getting single digit.
- Q. WAP to Check the given number is magic or not.

Q. $1234 \rightarrow 1+2+3+4 \rightarrow 10 \rightarrow 1+0 = 1$
last digit equal to 1.

- Q. WAP to Check the given number is happy or not.

W Class Demo {

PSUM() {

Scanner sc = new Scanner (System.in);
sc.nextInt();

int n = sc.nextInt();

SumofEachDigit(n);

};

Static void SumofEachDigit(int n){}

while(n>9){}

{

int sum=0;

while(n>0){}

int rem=n%10;

sum=sum+rem;

n=n/10;

}

n=sum;

s.o.p(sum);

}

Stop(n)

ix Class Demo {

psum() {

Scanner sc=new Scanner(System.in);

s.o.p("Enter");

int n=sc.nextInt();

sumofEachDigit(n);

}

static void SumofEachDigit(int n){}

while(n>9){}

int sum=0;

int rem = n % 10;

sum = sum + rem;

n = n / 10;

}

n = sum;

s.o.p(sum);

}

if (n == 3) {

s.o.p("Magic number");

}

else {

s.o.p("not");

}

}

X Class Demo S

psum(String args) {

Scanner sc = new Scanner(System.in);

s.o.p("Enter");

int n = sc.nextInt();

if (isHappy(n))

s.o.p("Happy...");

else {

s.o.p("not");

}

Public static boolean isHappy(int n)

{

 while(n > 9)

 {

 int sum = 0;

 while(n > 0)

 {

 int rem = n % 10;

 sum = sum + rem * rem;

 n = n / 10;

 }

 n = sum;

 }

 return n == 1 || n == 7 ? if(n == 1 || n == 7)

}

 S.O.PC("happy");
 else
 S.O.PC("not");

}

- Q. WAP to define a Method to Convert decimal value into binary.
- Q. WAP to define a Method to Convert decimal value into Octal.
- Q. WAP to define a Method to Convert decimal value into hexa decimal.
- Q. WAP to Swap two Number With Using temporary Variable.

Q. Write a program to swap two numbers without using temporary variable.

V. Swap two Numbers

Class Swap {

PSVM() {

int a = 10, b = 20;

int temp = a;

a = b;

b = temp;

SOP(a);

IF (a == 10 & b == 20) SOP(b);

if (a == 20 & b == 10)

if (a == 10 & b == 30)

V. Swap two Number Without Using temporary Variable

Class Swap {

PSVM() {

int a = 10, b = 20;

a = a + b;

b = a - b;

SOP(a);

3. SOP(b);

3

Diagram showing the state of variables a and b at different stages:

- Initial State: a = 10, b = 20
- After a = a + b: a = 30, b = 20
- After b = a - b: a = 30, b = 10
- Final State: a = 30, b = 10

Number System

- 1) Decimal $\rightarrow 10 \rightarrow 0 \text{ to } 9$
- 2) Binary $\rightarrow 2 \rightarrow 0 \text{ & } 1$
- 3) Octal $\rightarrow 8 \rightarrow 0 \text{ to } 7$
- 4) Hexadecimal $\rightarrow 16 \rightarrow 0 \text{ to } 9 \text{ & A to F}$

$$(13)_{10} \rightarrow (1101)_2 \quad (16)_{10} \rightarrow (10000)_2$$

$$\begin{array}{r} 13 \\ 2 | \overline{1} \quad 1 \\ 2 | \overline{6} \quad 0 \\ 2 | \overline{3} \quad 1 \\ 2 | \overline{2} \quad \uparrow \\ 1 \end{array}$$

$$\begin{array}{r} 16 \\ 2 | \overline{1} \quad 0 \\ 2 | \overline{8} \quad 0 \\ 2 | \overline{4} \quad 0 \\ 2 | \overline{2} \quad 0 \\ 1 \end{array}$$

W Convert decimal to Binary

Class Demo 5

```

PSVM: C:\> and ascii mapping (11)
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
decoBin(n);

}

static void decoBin(int n) {
    String bin = "";
    while(n>0) {
        int rem = n%2;
        n = n/2;
        bin = rem + bin;
    }
    System.out.println(bin);
}
    
```

```

W class Demo {
    public static void main(String args[]) {
        int n = 47;
        dectoOct(n);
    }
}

static void dectoOct(int n) {
    String oct = "";
    while (n > 0) {
        int rem = n % 8;
        oct = rem + oct;
        n = n / 8;
    }
    System.out.println(oct);
}

```

W Decimal into hexadecimal.

```

int dec = 31;
String hex = "";
while (dec > 0) {
    int rem =
}

class Demo {
    public static void main(String args[]) {
        int dec = 30;
        dectoHex(dec);
    }
}

```

```

Public static void dectohex(int dec)
{
    String hex = "";
    while (dec > 0)
    {
        int rem = dec % 16;
        if (rem > 9)
        {
            hex = (char)(rem + 55) + hex;
        }
        else
        {
            hex = rem + hex;
        }
        dec = dec / 16;
    }
    System.out.println(hex);
}

```

Q. WAP to convert binary value to decimal.

Class Demo {

```
public static void main(String[] args) {
```

```
    int bin = 1101;
```

```
    bintodec(bin);
```

```
}
```

```
public static int bintodec(int bin) {
```

```
    int dec = 0, p = 0, n = 2;
```

```
    while (bin > 0)
```

```
    {
```

```
        int rem = bin % 10;
```

```
        int pow = (int) Math.pow(2, p);
```

```
        dec = dec + rem * pow;
```

```
        bin = bin / 10;
```

```
        p++;
```

```
}
```

```
s.o.p(dec);
```

```
}
```

```

W. int bin = 110110;
    int dec = 0;
    int pow = 1;
    while (bin > 0) {
        int rem = bin % 10;
        dec = dec + rem * pow;
        bin = bin / 10;
        pow *= 2;
    }
    System.out.println(dec);
}
  
```

Q. WAP to print N fibonacci Series.
fibonacci series - 0 1 1 2 3 5

Class Demo {

```

    public static void main (String [] args) {
        int n = 5;
        fibo (n);
    }
  
```

}

P S V fibo (int n) { f

```
int num1=0, num2=1; num3=0;
for (int i=2; i<n; i++)
```

```
{ S.O.P (num1 + " " + num2 + " ");
  for (int i=1; i<=n-2; i++)
    {
```

$$f_3 = f_1 + f_2; \quad num3 = f_3$$

S.O.P (f_3 + " ");

$$f_1 = f_2;$$

$$f_2 = f_3; \quad num2 = f_2$$

}

}

W int f1=0, f2=1, f3=0;

S.O.P C

fibonacci is being done.

Step 1 = 0 + 1 = 1

Step 2 = 1 + 1 = 2

i = 3

Step 3 = 1 + 2 = 3

System: System is a Predefined Class which is present Java.lang Package.

out: Out is a Static Object Reference Variable of Print Stream Class which is in System Class.

Println() - } both are Predefined Method
Print() } PrintStream Class.

j-1 j-2 j-3 j-4 j-5
j-1 * * * * * i = represent the
j-2 * * * * * nth of row
j-3 * * * * * j = represent nth
j-4 * * * * * of column
j-5 * * * * *

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

```
{
```

```
    for (int j = 1; j <= 5; j++)
```

```
{
```

```
    S.o.print("*");
```

```
    System.out.println();
```

```
S.o.p();
```

```
}
```

```

int n = 1;
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++) {
        cout << "n";
    }
    cout << endl;
}
    
```

w

```

int n = 5;
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        cout << i + " ";
    }
    cout << endl;
}
    
```

w

```

int n = 5;
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        cout << j + " ";
    }
    cout << endl;
}
    
```

int n = 5;

1 0 1 0 1

for (int i = 1; i <= n; i++)

1 1 1 1 1

{

0 0 0 0 0

 for (int j = 1; j <= n; j++)

1 1 1 1 1

{

0 0 0 0 0

 if (i * 2 != 0)

1 1 1 1 1

 s.o.p("1");

 if (i * 2 == 0)

 s.o.p("0");

 }

 s.o.println();

}

 s.o.println();

 s.o.println("Program ended");

}

 for (int i = 1; i <= n; i++)

{

 for (int j = 1; j <= n; j++)

{

 s.o.p((i * 2) + " ");

}

 s.o.println();

}

W int n=5; 1 0 1 0 1

for (int i=1; i<n; i++) 1 0 1 0 1

{ 1 0 1 0 1

for (int j=i; j<n; j++) 1 0 1 0 1

{ 1 0 1 0 1

cout << s.o.p((j%2) + " ");

}

else

s.o.println();

}

j=1 j=2 3 4 5

W int n=5; 1 0 1 0 1

for (int i=0; i<n; i++) 1 0 1 0 0

{

1 0 1 0 1

for (int j=i; j<n; j++) 1 0 1 0 0

{

4 1 0 1 0 1

if (j%2==0)

s.o.p("1");

s.o.p((i+j)%2 + " ");

}

s.o.println();

}

M	T	W	T	F	S	S
Page No.	YOUVA					
Date:						

W int n = 5, r = 1;

for (int i = 1; i <= n; i++) {

 for (int j = 1; j <= n; j++) {

 S.o.printf("%d",

 r++;

 } 1 2 3 4 5

 } 6 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

W int n = 5, x = 1;

for (int i = 1; i <= n; i++) {

 } 1 3 5 7 9

 } for (int j = 1; j <= n; j++) {

 } 10 11 13 15 17 19

 } 20 21 23 25 27 29

 } 30 31 33 35 37 39

 } 40 41 43 45 47 49

 } 50 51 53 55 57 59

 } 60 61 63 65 67 69

 } 70 71 73 75 77 79

 } 80 81 83 85 87 89

 } 90 91 93 95 97 99

 } 100 101 103 105 107 109

 } 110 111 113 115 117 119

 } 120 121 123 125 127 129

 } 130 131 133 135 137 139

 } 140 141 143 145 147 149

 } 150 151 153 155 157 159

 } 160 161 163 165 167 169

 } 170 171 173 175 177 179

 } 180 181 183 185 187 189

 } 190 191 193 195 197 199

 } 200 201 203 205 207 209

 } 210 211 213 215 217 219

 } 220 221 223 225 227 229

 } 230 231 233 235 237 239

 } 240 241 243 245 247 249

 } 250 251 253 255 257 259

 } 260 261 263 265 267 269

 } 270 271 273 275 277 279

 } 280 281 283 285 287 289

 } 290 291 293 295 297 299

 } 300 301 303 305 307 309

 } 310 311 313 315 317 319

 } 320 321 323 325 327 329

 } 330 331 333 335 337 339

 } 340 341 343 345 347 349

 } 350 351 353 355 357 359

 } 360 361 363 365 367 369

 } 370 371 373 375 377 379

 } 380 381 383 385 387 389

 } 390 391 393 395 397 399

 } 400 401 403 405 407 409

 } 410 411 413 415 417 419

 } 420 421 423 425 427 429

 } 430 431 433 435 437 439

 } 440 441 443 445 447 449

 } 450 451 453 455 457 459

 } 460 461 463 465 467 469

 } 470 471 473 475 477 479

 } 480 481 483 485 487 489

 } 490 491 493 495 497 499

 } 500 501 503 505 507 509

 } 510 511 513 515 517 519

 } 520 521 523 525 527 529

 } 530 531 533 535 537 539

 } 540 541 543 545 547 549

 } 550 551 553 555 557 559

 } 560 561 563 565 567 569

 } 570 571 573 575 577 579

 } 580 581 583 585 587 589

 } 590 591 593 595 597 599

 } 600 601 603 605 607 609

 } 610 611 613 615 617 619

 } 620 621 623 625 627 629

 } 630 631 633 635 637 639

 } 640 641 643 645 647 649

 } 650 651 653 655 657 659

 } 660 661 663 665 667 669

 } 670 671 673 675 677 679

 } 680 681 683 685 687 689

 } 690 691 693 695 697 699

 } 700 701 703 705 707 709

 } 710 711 713 715 717 719

 } 720 721 723 725 727 729

 } 730 731 733 735 737 739

 } 740 741 743 745 747 749

 } 750 751 753 755 757 759

 } 760 761 763 765 767 769

 } 770 771 773 775 777 779

 } 780 781 783 785 787 789

 } 790 791 793 795 797 799

 } 800 801 803 805 807 809

 } 810 811 813 815 817 819

 } 820 821 823 825 827 829

 } 830 831 833 835 837 839

 } 840 841 843 845 847 849

 } 850 851 853 855 857 859

 } 860 861 863 865 867 869

 } 870 871 873 875 877 879

 } 880 881 883 885 887 889

 } 890 891 893 895 897 899

 } 900 901 903 905 907 909

 } 910 911 913 915 917 919

 } 920 921 923 925 927 929

 } 930 931 933 935 937 939

 } 940 941 943 945 947 949

 } 950 951 953 955 957 959

 } 960 961 963 965 967 969

 } 970 971 973 975 977 979

 } 980 981 983 985 987 989

 } 990 991 993 995 997 999

 } 1000 1001 1003 1005 1007 1009

 } 1010 1011 1013 1015 1017 1019

 } 1020 1021 1023 1025 1027 1029

 } 1030 1031 1033 1035 1037 1039

 } 1040 1041 1043 1045 1047 1049

 } 1050 1051 1053 1055 1057 1059

 } 1060 1061 1063 1065 1067 1069

 } 1070 1071 1073 1075 1077 1079

 } 1080 1081 1083 1085 1087 1089

 } 1090 1091 1093 1095 1097 1099

 } 1100 1101 1103 1105 1107 1109

 } 1110 1111 1113 1115 1117 1119

 } 1120 1121 1123 1125 1127 1129

 } 1130 1131 1133 1135 1137 1139

 } 1140 1141 1143 1145 1147 1149

 } 1150 1151 1153 1155 1157 1159

 } 1160 1161 1163 1165 1167 1169

 } 1170 1171 1173 1175 1177 1179

 } 1180 1181 1183 1185 1187 1189

 } 1190 1191 1193 1195 1197 1199

 } 1200 1201 1203 1205 1207 1209

 } 1210 1211 1213 1215 1217 1219

 } 1220 1221 1223 1225 1227 1229

 } 1230 1231 1233 1235 1237 1239

 } 1240 1241 1243 1245 1247 1249

 } 1250 1251 1253 1255 1257 1259

 } 1260 1261 1263 1265 1267 1269

 } 1270 1271 1273 1275 1277 1279

 } 1280 1281 1283 1285 1287 1289

 } 1290 1291 1293 1295 1297 1299

 } 1300 1301 1303 1305 1307 1309

 } 1310 1311 1313 1315 1317 1319

 } 1320 1321 1323 1325 1327 1329

 } 1330 1331 1333 1335 1337 1339

 } 1340 1341 1343 1345 1347 1349

 } 1350 1351 1353 1355 1357 1359

 } 1360 1361 1363 1365 1367 1369

 } 1370 1371 1373 1375 1377 1379

 } 1380 1381 1383 1385 1387 1389

 } 1390 1391 1393 1395 1397 1399

 } 1400 1401 1403 1405 1407 1409

 } 1410 1411 1413 1415 1417 1419

 } 1420 1421 1423 1425 1427 1429

 } 1430 1431 1433 1435 1437 1439

 } 1440 1441 1443 1445 1447 1449

 } 1450 1451 1453 1455 1457 1459

 } 1460 1461 1463 1465 1467 1469

 } 1470 1471 1473 1475 1477 1479

 } 1480 1481 1483 1485 1487 1489

 } 1490 1491 1493 1495 1497 1499

 } 1500 1501 1503 1505 1507 1509

 } 1510 1511 1513 1515 1517 1519

 } 1520 1521 1523 1525 1527 1529

 } 1530 1531 1533 1535 1537 1539

 } 1540 1541 1543 1545 1547 1549

 } 1550 1551 1553 1555 1557 1559

 } 1560 1561 1563 1565 1567 1569

W int n = 5;

5 4 3 2,

for (int i = 1; i < n; i++)

5 4 3 2,

{

5 4 3 2,

for (int j = n; j >= 1; j++)

5 4 3 2,

{

5 4 3 2,

s.o.p(j + " ");

}

s.o.println();

}

W int n = 5

1 2 3 4 5

for (int i = 1; i < n; i++)

2 3 4 5 6

{

3 4 5 6 7

int r = i;

4 5 6 7 8

for (int j = 1; j < n; j++)

5 6 7 8 9

{

s.o.print(r + " ");

r++;

{

s.o.println();

{

W int n = 5;

for (int i = 1; i < n; i++) {

{ for (int j = 1; j < i; j++) {

s.o.print("*");

}

s.o.println();

}

W int n = 5

for (int i = 1; i < n; i++) {

{

for (int j = 1; j < i; j++) {

{

s.o.p(j + " ");

}

s.o.println();

W int n = 5, int x = 1;

for (int i = 1; i < n; i++) {

{ for (int j = 1; j < i; j++) {

s.o.p("x%d", x);

x++;

{

s.o.println();

}

```
W int n = 5;
for (int i = 1; i <= n; i++)
{
    for (int j = i; j >= 1; j--)
        s.o.print(j + " ");
    s.o.println();
}
```

```
W int n = 5;
for (int i = n; i >= 1; i++)
{
    for (int j = 1; j <= i; j++)
        s.o.p(" * ");
    s.o.println();
}
```

```

W int n = 5; *
for (int i = 1; i < n; i++) *
{
    for (int j = 1; j < n - i; j++) ** **
        s.o.print(" " * 112 spaces ** ** ** *
    }
    for (int k = 1; k < i; k++) *
        s.o.print("*" * 111 spaces *
    }
    s.o.println();
}

```

```

W int n = 5; *
for (int i = 1; i < n; i++) {
    for (int j = 1; j < n - i; j++) ** **
        s.o.p(" " * 112 spaces ** ** ** *
    }
    for (int k = 1; k < i; k++) *
        s.o.p("*" * 111 spaces *
    }
    s.o.println();
}

```

M T W T F S
Page No.:
Date: YOUNG

```
* int n = 5
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n - i; j++) {
        System.out.print(" ");
    }
    for (int k = i; k <= i + i - 1; k++) {
        System.out.print(k + " ");
    }
    System.out.println();
```

```
* int n = 5
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n - i; j++) {
        System.out.print(j + " ");
    }
    for (int k = i; k >= 1; k--) {
        System.out.print(k + " ");
    }
    System.out.println();
```

int n = 5; ~~for (int i = 1; i < n; i++)~~ A

for (int i = 1; i < n; i++) A B C D

{

E S T

A B C D

for (int j = 1; j < i; j++) A B C D

{ F D A + E C I A B C D E

PS. O. P((cchar)(j + 64) + "e"));

}

S. O. printen();

}

for (int i = 1; i < n; i++)

{ for (int j = 1; j < i; j++)

*

int n = 5;

if (" ") n is equal to *

for (int i = 1; i < n; i++)

* * * * *

{ for (int j = 1; j < n - i; j++)

* * * * *

{

if (" ") n is equal to

S. O. P(" ");

{

if (" ") n is equal to

for (int k = 1; k <= 2 * i - 1; k++)

f

{ S. O. P(" * ");

{

S. O. printen();

{

i=1 i=2 i=3 i=4 i=5 i=6 i=7 i=8 i=9

i=1

i=2

i=3

i=4

i=5

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1 2 3 4 5 6

1 2 3 4 5 6 7

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8 9

int n=5;

for(int i=1; i<=n; i++) {

 for(int j=1; j<=n-i; j++)

}

 System.out.print(" ");

 for(int k=1; k<=2*i-1; k++)

 System.out.print(k + " ");

 System.out.println();

}

System.out.println();

7

```

int n = 5;
for (int i = n; i >= 1; i--) {
    for (int j = 1; j <= n - i; j++) {
        if (j < i) cout << " ";
        else cout << "*";
    }
    cout << endl;
}

```

```

int n = 5
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
        if (n - i > j) cout << " ";
        else cout << "*";
    }
    cout << endl;
}

for (int k = 1; k <= 2 * i - 1; k++) {
    cout << (char)(k + 64) + " ";
}
cout << endl;

```

```

* int n = 5;
* for (int i = n; i > -1; i--) {
*   for (int j = 1; j < n - i; j++) {
*     s.o.p(" * ");
*   }
*   for (int k = 2 * i - 1; k >= 1; k--) {
*     s.o.print(" * ");
*   }
*   s.o.println();
* }
* s.o.open();
* s.o.a
* int n = 5;
* for (int i = 1; i < n; i++) {
*   for (int j = 1; j < n - i; j++) {
*     s.o.p(" * ");
*   }
*   for (int k = 1; k <= 2 * i - 1; k++) {
*     s.o.print((char)(k + 96) + " ");
*   }
*   s.o.println();
* }
* s.o.close();

```

M	T	W	T	F	S	S
Page No.	YOUVA					
Date:						

```

int n=5;
for(int i=1; i<n; i++)
{
    for(int j=1; j<n-i; j++)
        System.out.print(" ");
    System.out.print("10101");
}
System.out.println();
    
```

```

for(int k=j; k<=2*i-1; k++)
    System.out.print(k%2+"");
}
System.out.println();
    
```

```

System.out.println();
    
```

W i=1 - - - *

i=2 - - * * *

i=3 - * * * * *

i=4 * * * * * *

i=5 - * * * * *

i=6 - - * * * *

i=7 - - - * * * *

```

int n=7, Sp=n/2; St=1;
for(int i=1; i<=n; i++)
    
```

```

        for(int j=1; j<=Sp; j++)
            System.out.print(" ");
    }
}
    
```

```

for (int k = 1; k <= St; k++) {
    if (n % 2 == 0) {
        if (n / 2 == 0) {
            Sp += n;
            St = St + 2;
        } else {
            Sp += n;
        }
    } else {
        Sp += n;
        St = St - 2;
    }
}

```

S.print();

7

W

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

(int n = 7), Sp = n/2, St = 1;

for (int i=0; i<n; i++)

{

 for (int j=1; j<=sp; j++)

{

 s.o.print (" ");

 for (int k=1; k<=st; k++)

{

 s.o.p(k+ " ");

}

 if (i<-n/2)

{

 sp--;

 st=st+2;

}

 else

 sp+=i; else -i);

 st=st-2; -n/2

}

 s.o.print();

}

12

1

2 2 2

3 3 3 3 3 int n = 7;

4 4 4 4 4 4 4 int sp = n / 2;

5 5 5 5 5 int st = 1;

6 6 6

7

for (int i = 1; i < n; i++) {

for (int j = 1; j <= sp; j++)

{

S.o.p(" ");

for (int k = 1; k <= st; k++)

{

S.o.p(i + " ");

}

if (i <= n / 2) {

sp -= i - n / 2 - 1;

st = st + 2;

{

} else {

sp += i;

st = st - 2;

{

S.o.pn();

{

W	F	T	P	S	B
000000	000000	000000	000000	000000	000000
3000	3000	3000	3000	3000	3000

```

W           F           T           P           S           B
int n=7;
int sp=n/2;
int rt=rt-1;
for(int i=1;i<=sp;i++)
{
    s.o.p(" ");
}
for(int j=1;j<=sp;j++)
{
    s.o.p("  ");
}
if(i<=n/2)
{
    sp--;
    rt=rt+2;
}
else
{
    sp++;
    rt=rt-2;
}
s.o.println();
}

```

W int n = 7;

1

int &p = n/2;

E 1 0 1

int st = 7;

1 0 1 0 1

for (int i = 1; i <= n; i++)

1 0 1 0 1 0 1

{

for (int j = 1; j <= sp; j++)

1

{

s. o. p(" ");

(K.Y.2)

}

for (int k = 1; k <= st; k++)

{

s. o. p(k % 2 + " ");

{

if (i < n/2)

{

sp -= 3;

st = st + 2;

{

else {

sp += i;

st = st - 2;

{

- s. o. p(eul);

{

— 10 —

A

20

B B B

Fig. 1. A schematic diagram of the experimental setup.

c c c c c

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

D D D D D D D

19. 10. 1986

E E E E E

F F F

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

(i + 64)

卷之三

— 10 —

1 2 3 4 5 6 7 8

— 1 —

— 1 —

Figure 1. The effect of the number of nodes on the error.

— 10 —

int n = 7;

A

int sp = n / 2;

A B C

int rt = 1;

A B C D E

for (int i = 1; i < n / 2; i++)

A B C D E F G

{
 for (int j = 1; j < sp; j++)

A B C D E

A B C

 for (int k = 1; k < rt; k++)

A

 {
 if (i == j)

(K + 64)

 S.o.p(" ");

7

 for (int k = 1; k < st; k++)

 S.o.p((char)(K + 64) + " ");

7

 if (i < n / 2)

{

 sp--;

 rt = rt + 2;

7

 else

 sp++;

 rt = rt - 2;

7

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

```

int n = 5;
for (int i=1; i<=n; i++) {
    for (int j=1; j<=n; j++) {
        if (i == n/2 + 1 || j == n/2 + 1)
            cout << "* ";
        else
            cout << " ";
    }
    cout << endl;
}

```

```

int n = 5;
for (int i=1; i<=n; i++) {
    for (int j=1; j<=n; j++) {
        if (i == j || i+j == n+1)
            cout << "* ";
        else
            cout << " ";
    }
    cout << endl;
}
cout << endl;

```

```
int n=5;
```

```
for(int i=1; i<=n; i++) { * * * * }
```

```
for(int j=1; j<=n; j++) { * * }
```

```
if(j==1 || i==j || j==n)
```

```
s.o.print("*");
```

```
else { " " } ;
```

```
s.o.print(" ");
```

```
s.o.println();
```

```
int n=5;
```

```
for(int i=1; i<n; i++) { * * }
```

```
{ for(int j=1; j<n; j++) { * * }
```

```
{ }
```

```
if(i==1 || j==1 || i==n || j==n ||
```

```
i==j || i+j==n+1) { }
```

```
s.o.p(" *");
```

```
else
```

```
s.o.p(" ");
```

```
s.o.p();
```

```
}
```

j: 1 2 3 4 5

```

int n=5;
for(int i=1; i<=n; i++) {
    for(int j=1; j<=n; j++) {
        if(j==1 || j==n || j==i || j<=n/2+1
           || i+j==n+1 || i<=n/2+1)
            cout<"*";
        else
            cout<" ";
    }
    cout<endl;
}

```

```

int n=5;
for(int i=1; i<=n; i++) {
    for(int j=1; j<=n; j++) {
        if(i==1 || j==1 || i==n
           || j==n || i>n/2+1 || i<=n/2+1
           || j>n/2+1)
            cout<"*";
        else
            cout<" ";
    }
    cout<endl;
}

```

```

int n=5
for(int i=1; i<=n; i++)
{
    for(int j=1; j<=n; j++)
    {
        if(j==1 && i<=n/2+1 || i==n && j<=n/2
           || j==n/2+1 || i==n/2+1 ||
           i==1 && j>=n/2+1 || j==n &&
           j>=n/2+1)
            S.o.p("*");
        else
            S.o.p(" ");
    }
}

```

```

S.o.p();
j: 1 2 3 4 5 6 7
int n=7;
int Sp=1; i=1; j=1;
int St=n/2;
for(i=1; i<=n; i++)
{
    for(j=1; j<=Sp; j++)
        S.o.p(" ");
    S.o.p("\n");
    Sp+=2;
}

```

M	T	W	T	F	S	S
Page No.	YOUVA					
Date						

```

for (int k=1; k<=st; k++)
{
    if (k==1 || k==st) {
        for (int i=1; i<=sp; i++)
            cout << "*";
    }
    else {
        for (int i=1; i<=sp-2; i++)
            cout << " ";
        cout << "*";
        for (int i=1; i<=sp-2; i++)
            cout << " ";
    }
}
cout << endl;

```

```
int n=5; int i,j,k,l,m;
int st=1;
for(i=1;i<n;i++) { 1 2 3
    for(j=i+1;j<st;j++) { 1 2 3 4 5
        if(i==j) { 1 2 3
            cout << j << endl; 2 3
        }
        st++; 4
    }
    cout << endl; 5
}
```

M	T	W	T	F	S	S
Page No.:						
Date:						YOUVA

Array

Array is any collection of homogeneous data value / element.

- ↳ Array is an Object.
 - ↳ Array is index based that means data will get initiated based on the index Value.
 - ↳ ~~Array~~ always array Size is fixed.
 - ↳ Array doesn't have any predefined to deal with data / element.
- Continuous Memory allocation.
 - Searching / fetching the element is easy because it is index based.
 - Always Array index Start from 0.
 - Last index of an array is Size - 1.

Syntax for declaration of an array:

Array type / Array []:
data type Name

int x[]; // declaration

int *x = new int [size] // Initialization

int y[] = new int [5];

Syntax for initialization elements:

Any index i : data / element.

Name

int x[] = new int [4];

x[0] = 10;

x[1] = 20 x = [10, 20, 30, 40]

x[2] = 30

x[3] = 40 for (int i = 0; i < x.length(); i++)

System.out.println(x[i]);

M	T	W	T	F	S	S
Page No.:						
Date:						YOUVA

1. WAP to print even index elements in the array.

$r = \$\{10, 20, 30, 40, 50\}$

```
for (int i = 0; i < r.length - 1; i++) {
    if (i % 2 == 0) {
        System.out.println(r[i]);
    }
}
```

2. WAP to print sum of odd elements in the array.

$\text{int } r = \$\{10, 20, 30, 40, 50\};$

$\text{int sum} = 0;$

```
for (int i = 0; i < r.length - 1; i++) {
    if (r[i] % 2 == 1) {
        sum = sum + r[i];
    }
}
```

$\text{System.out.println(sum);}$

3

3. Write a program to print all the prime numbers present in the array.

```
int x[] = {1, 2, 3, 4, 5, 6, 7};
```

```
for (int i=0; i<x.length-1; i++) {
    if (isPrime(x[i])) {
        System.out.println(x[i]);
    }
}
```

```
private static Boolean isPrime(int n) {
    for (int i=2; i<n/2; i++) {
        if (n % i == 0)
            return false;
    }
    return true;
}
```

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

4. WAP to take array input from the user and print the array element.

```
import java.util.Scanner;
```

```
import java.util.*;
```

```
Scanner sc = new Scanner(System.in);
```

```
s.o.println("Enter the size of an array");
```

```
int size = sc.nextInt();
```

```
int a[] = new int[size];
```

```
s.o.p("Enter " + size + " array element");
```

```
for (int i=0; i<a.length-1; i++) {
```

```
    a[i] = sc.nextInt();
```

```
}
```

```
s.o.p("array elements are");
```

```
for (int i=0; i<a.length-1; i++) {
```

```
    s.o.p(a[i] + " ");
```

```
}
```

```
// s.o.println(Arrays.toString(a));
```

↳ print array element without using loop.

5 WAP to replace every element by the sum of each digit.

```
Scanner sc = new Scanner(System.in);
```

```
s.o.println("enter the size");
```

```
int size = sc.nextInt();
```

```
int x[] = new int[size];
```

```
s.o.println("Enter " + size + " array elements");
```

```
for (int i = 0; i <= a.length - 1; i++) {
```

```
a[i] = sc.nextInt();
```

```
}
```

```
for (int i = 0; i <= a.length - 1; i++) {
```

```
a[i] = SumofDigit(a[i]);
```

```
}
```

```
s.o.println("after replacing the element");
```

```
s.o.println(Arrays.toString(a));
```

```
}
```

```
private static int SumofDigit(int n) {
```

```
int sum = 0;
```

```
while (n > 0) {
```

```
int rem = n % 10;
```

```
sum += rem;
```

```
n /= 10;
```

M	T	W	T	F	S	S
Page No.:						
Date:						YOUVA

return sum;

7

- Q. WAP to print Smallest element in the User input array.
- Q. WAP to print Biggest element in the User input array.
- Q. WAP to print Count how Many Strong are present are p in the User input array.
- Q. WAP to Search User enter number in the user enter array if it is found Print the location or index of the number otherwise print Number Not found.

1. Scanner sc = new Scanner (System.in);
sc.nextLine(); // Enter the size;
int size = sc.nextInt();

int r[] = new int [size];
sc.nextLine(); // Enter + size + " array element";
for (int i = 0; i < r.length - 1; i++)
{}

r[i] = sc.nextInt();

}

int big = r[0];

for (int i = 1; i < r.length - 1; i++)
{ if (r[i] > big)

big = r[i];

}

sc.nextLine();

2. Scanner sc = new Scanner (System.in);
sc.nextLine(); // Enter the size;
int size = sc.nextInt();

int r[] = new int [size];

sc.nextLine(); // Enter + size + " array element";

M	T	W	T	F	S	S
Page No.:						
Date:						YOUVA

for (int i = 0; i < x.length(); i++)

{

y[i] = Sc.nextInt();

}

int small = x[0];

for (int i = 1; i < x.length - 1; i++)

{

if (x[i] < small)

{

small = x[i];

Sc.out.println(small);

Q WAP to Check Smallest and Biggest element Sum is Prime Number or not.

Scanner Sc = new Scanner (sy)

int a [] = {1, 2, 4, 9, 77};

int small = a[0];

int big = a[0];

for (int i = 1; i < a.length - 1; i++)

if (a[i] < small) {

small = a[i];

if (a[i] > big) {

big = a[i];

} }

```

int sum = Small + big;
int count = 0;

for (int i = 1; i <= sum; i++)
    if (sum % i == 0)
        count++;

if (count == 2)
    S.o.p("Sum is Prime");
else
    S.o.p("Sum is Not prime");

```

W Scanner : Sc = new Scanner(System.in);

```

Scanner Sc = new Scanner(System.in);
Sc.p("enter the size");
int size = Sc.nextInt();
Sc.p("enter " + Size + " array elements");
for (int i = 0; i < r.length - 1; i++)
    r[i] = Sc.nextInt();
int small = r[0], big = r[0];

```

W	T	W	F	S
Payee:				YUVRAJ
Date:				

function calculateSum(n) {
let sum = 0;

if (n < 0) return -1;

if (n == 0) return 0;

if (n > 0) {
sum += n;

for (let i = 1; i < n; i++) {
sum += i;

if (i * i == n) {
sum -= i;

return sum + Small + big;

else if (n < 0) {
return sum - Small - big;

else if (n == 0) {
return sum;

else if (n > 0) {
sum += n;

if (isPrime(sum)) {
return true;

else {
return false;

else {
return false;

else if (sum < 0) {
return false;

else if (sum == 0) {
return false;

else if (sum > 0) {
return true;

Public static boolean isPrime(int n) {

if (n <= 1) {
return false;

else return true;

for (int i = 2; i < n / 2; i++) {
if (n % i == 0) {
return false;

else if (n % i == 0) {
return false;

else return true;

}
return true;

4. Scanner sc = new Scanner (System.in);
sc.nextLine("enter the size");
int size = sc.nextInt();

int x[] = new int [size];
sc.nextLine("enter " + size + " array elements");
for (int i = 0; i < x.length - 1; i++)
{
 x[i] = sc.nextInt();
}

sc.nextLine("enter the number to search");
int num = sc.nextInt();

int res = linearSearch (x, num);
if (res == -1)
 sc.nextLine("element not found");
else
 sc.nextLine("element found in " + res +
 "Position");

Start a int binarySearch (int x[], int n)

{

 for (int i = 0; i < length - 1; i++)

 if (x[i] == n)

 return i;

}

 return -1;

}

Q1. WAP to Sort the array element
in ascending Order

Q2. WAP for sort the array element

in descending Order.

* Count Strong number in an array.

```

int r = [140, 145, 250, 27];
int count = 0;
for (int i = 0; i < r.length - 1; i++) {
    if (isStrong(r[i])) {
        count++;
    }
}
System.out.println(count);

```

```

P.S. boolean isStrong (int n) {
    int sum = 0, temp = n;
    while (n > 0) {
        int rem = n % 10;
        int fact = factorial(rem);
        sum = sum + fact;
        n = n / 10;
    }
    return sum == temp;
}

```

```

P.S. int factorial (int n)
{
    int fact = 1;
    for (int i = n; i > 1; i--) {
        fact = fact * i;
    }
    return fact;
}

```

* Sort the array element in ascending order.

```
#include <iostream>
int x[] = {2, 5, 7, 1, 4};
```

```
for (int i = 0; i < x.length - 1; i++) {
    for (int j = i + 1; j < x.length - 1; j++)
```

```
        if (x[i] > x[j])
```

```
            {
```

```
                int temp = x[i];
```

```
                x[i] = x[j];
```

```
                x[j] = temp;
```

```
s.out<operator><operator>::toString(x[i]);
```

* int x[] = {1, 2, 1, 7, 5};

```
for (int i = 0; i < n; i++) {
```

```
    for (int j = i + 1; j < x.length - 1; j++) {
        if (x[i] < x[j]) {
```

```
            int temp = x[i];
```

```
            x[i] = x[j];
```

```
            x[j] = temp;
```

```
        }
```

```
s.out<operator><operator>::toString(x[i]);
```

a) Write a program to merge two arrays into a single array.

```
Scanner sc = new Scanner(System.in);
int x[] = {1, 2, 3, 4, 5};
```

```
int y[] = {5, 7, 9, 11};
```

```
int n = x.length;
```

```
int m = y.length;
```

```
int arr = new int[n+m];
```

```
for (int i = 0; i < n; i++) {
```

```
arr[i] = x[i];
```

```
Scanner sc = new Scanner(System.in);
```

```
s.nextLine() "Enter the 1st array size";
```

```
int size1 = sc.nextInt();
```

```
int x[] = new int[size1];
```

```
s.nextLine() "Enter " + size1 + " array elements";
```

```
for (int i = 0; i <= size1 - 1; i++) {
```

```
x[i] = sc.nextInt();
```

```
} // 1st array completed
```

```
s.nextLine() "Enter the 2nd array size";
```

```
int size2 = sc.nextInt();
```

```
int y[] = new int[size2];
```

```
s.nextLine() "Enter " + size2 + " array elements";
```

```

for (int i=0; i<y.length-1; i++) {
    y[i] = s.nextInt();
}
    
```

```

int arr[] = merge(r, y);
    
```

```

System.out.println(Arrays.toString(arr));
    
```

```

static int[] merge(int[] r, int[] y)
{
    
```

```

        int z[] = new int[r.length + y.length];
    
```

```

        for (int i=0; i<r.length-1; i++) {
    
```

```

            z[i] = r[i];
    
```

```

            for (int j=0; j<y.length-1; j++) {
    
```

```

                z[r.length+j] = y[j];
    
```

```

            }
    
```

```

        return z;
    
```

```

}
    
```

Q.1 WAP to print Occurrence or frequency each element in the user entered array.

Q.2 WAP to print Unique elements in the user entered array

Q.3 WAP to print duplicate element in the array.

```
int x[] = {10, 20, 10, 20, 30};
```

```
for (int i = 0; i < x.length - 1; i++) {
```

```
    int Count = 1;
```

```
    for (int j = i + 1; j < x.length; j++) {
```

```
        if (x[i] == x[j]) {
```

```
            Count++;
```

```
x[j] = -1;
```

```
}
```

```
if (x[i] != -1)
```

```
System.out.println(x[i] + " - " + Count);
```

```
}
```

```

1. int r = {10, 20, 30, -2, 31}; // r is an array
boolean b[] = new boolean[r.length];
for (int i = 0; i < r.length - 1; i++) {
    if (b[i] == false) {
        int count = 1;
        for (int j = i + 1; j < r.length - 1; j++) {
            if (r[i] == r[j])
                count++;
            b[j] = true;
        }
        System.out.println(r[i] + " - " + count);
    }
}

```

2.

```

int r[] = { 10, 20, 20, 10, 20 };
boolean b[] = new boolean[x.length];
for (int i = 0; i < x.length - 1; i++) {
    if (b[i] == false) {
        int Count = 1;
        for (int j = i + 1; j < x.length - 1; j++) {
            if (x[i] == x[j]) {
                Count++;
            }
            x[j] = true;
        }
        if (Count == 1)
            System.out.println(x[i] + " - " + Count);
    }
}

```

3.

```

int x = { 10, 20, 30, 20, 10 };
boolean b[] = new boolean[x.length];
for (int i = 0; i < x.length - 1; i++) {
    if (b[i] == false) {
        int Count = 1;

```

```
for (int j = i + 1; j <= x.length - 1; j++)
```

```
if (x[i] == x[j])
```

```
{
```

```
Count++;
```

```
if (x[i] > x[j] == true):
```

```
{
```

```
}
```

```
if (CountAt > 1)
```

```
S.o.println(x[i] + " " + Count);
```

```
}
```

```
* int x = { 10, 20, 30, 10 };
```

```
boolean b[] = new boolean[x.length];
```

```
for (int i = 0; i < x.length - 1; i++) {
```

```
if (b[i] == false):
```

```
{
```

```
int Count = 1;
```

```
for (int j = i + 1; j <= x.length - 1; j++) {
```

```
if (x[i] == x[j])
```

```
{
```

```
Count++;
```

```
x[j] = true;
```

```
{ }
```

```
S.o.println(x[i]);
```

```
}
```

Q. WAP to print paired Elements whose Sum is equal to given number.

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

```
int num = 6;
```

```
boolean b[] = new boolean[r.length];
```

```
Boolean B[] = new Boolean[R.length];
```

```
for (int i = 0; i < x.length - 1; i++)
```

```
{
```

~~Partial Output~~

```
if (b[i] == false)
```

```
{
```

```
for (int j = i + 1; j < r.length; j++)
```

```
{
```

```
if (r[i] + x[j] == num)
```

```
{
```

```
S.o.p(r[i] + " " + r[j]);
```

```
b[j] = true;
```

```
}
```

1. WAP to insert User entered numbers in the user entered Specified index.
2. WAP to delete User entered Specified index element.
3. Give WAP to print missing elements in the array.

```

1. Scanner sc = new Scanner (System.in);
sc.open ("Enter the size");
int size = sc.nextInt();
int x[] = new int [size];
sc.open ("enter " + size + " elements");
for (int i = 0; i < x.length; i++) {
    x[i] = sc.nextInt();
}
sc.open ("enter number to insert");
int num = sc.nextInt();
sc.open ("enter index to insert");
int in = sc.nextInt();
sc.open ("after inserting");

```

Page No.:
Date:

```
int a[] = insert(x, num, in);
if (a == null)
    System.out("invalid index");
else
    System.out(Arrays.toString(a));
}

public static int[] insert(int[] x,
    int num, int in)
{
    if (in < 0 || in > x.length)
        return null;
    int y[] = new int[x.length + 1];
    y[in] = num;
    for (int i = 0; i < x.length - 1; i++)
        if (i < in)
            y[i] = x[i];
        else if (y[i] > y[i + 1])
            y[i + 1] = x[i];
    return y;
}
```

- Q1. WAP to merge two integer array into single array in zigzag format.

Q2. WAP to check the array elements are in ascending order or not.

Q3. Scanner Sc = new Scanner (System.in);
Sc·op (" enter Size ");
int size1 = Sc.nextInt();
int x [] = new int [size1];

Sc·op (" enter " + size1 + " element ");
for (int i = 0; i < size1.length - 1; i++)
{
 x[i] = Sc.nextInt();
}

Sc·op (" enter the 2nd array Size ");
int size2 = Sc.nextInt();
int y [] = new int [size2];

Sc·op (" enter " + size2 + " element ");
for (int i = 0; i < size2.length - 1; i++)
{
 y[i] = Sc.nextInt();
}

int arr = zigzag(x, y);

String Array to string(avr);

private static int[] zigzag(int[] x,
int[] y) {

int[] zigzag = new int[x.length + y.length];

int k = 0;

for (int i = 0; i <= x.length - 1 || i <
y.length - 1; i++) {

if (i < x.length - 1)

z[k] = x[i];

k++;

if (i < y.length - 1)

{

z[k] = y[i];

k++;

return z;

}

2.

```

int r[] = {1, 4, 2, 6, 5};
boolean res = ascending(r);

if (res == true)
    S.o.p("Ascending");
else
    S.o.p("not");

```

```

public static boolean ascending(int r) {
    for (int i = 0; i < r.length - 1; i++)
        if (r[i] > r[i + 1])
            return false;
    return true;
}

```

Q1. WAP to print how many String element are present in object array.

Object o[] = {15.08, "ab", 45, 'A', 20, 100.98, false};

```

int count = 0;
for (int i = 0; i < o.length - 1; i++) {
    if (o[i] instanceof String) {
        count++;
    }
}
System.out.println(count);
    
```

Q2. WAP to Print double type element Present in object array.

Object o[] = {15.08, "ab", 45, 'A', 20, 100.98, false};

```

for (int i = 0; i < o.length - 1; i++) {
    if (o[i] instanceof Double) {
        System.out.println(o[i]);
    }
}
    
```

* Print Missing Element in the Array.

```

int r[] = {1, 2, 5, 7, 9};
int small = r[0], big = r[0];
for (int i = 0; i < r.length - 1; i++) {
    if (r[i] < small)
        small = r[i];
    if (r[i] > big)
        big = r[i];
}

```

```

for (int i = small; i < big; i++)
{
    boolean res = isMissing(i, r);
    if (res == true)
        System.out.println(i + " ");
}

```

```

private static boolean isMissing(int n, int r[])
{

```

```

    for (int i = 0; i < r.length - 1; i++) {
        if (r[i] == n)
            return false;
    }

```

```

    return true;
}

```

S	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

Q. WAP to find nth Smallest element
in the given array.

Public Class Test {

 PSVN () {

 int r [] = { 15, 4, 2, 10, 8, 10 };

 int n = 3;

 int y [] :

* WAP to delete user enter specified index element.

```

int x[] = {10, 20, 30, 40};
int id = 2;
int a[] = deleting(x, id);
if (a == null)
    System.out.println("Invalid Index");
else
    System.out.println(Arrays.toString(a));
}

public static int[] deleting(int[] x, int id) {
    if (id < 0 || id > x.length)
        return null;
    int y[] = new int[x.length - 1];
    int j = 0;
    for (int i = 0; i < x.length; i++) {
        if (i == id)
            continue;
        y[j++] = x[i];
    }
    return y;
}

```

- String is a predefined final class present in java.lang package from JDK 1.0 version.
- String is also known as Collection of Characters / Sequence of Characters.
- Internally String characters are stored based on the index value.
- String is immutable in nature / String is not modifiable.
- String is immutable because of String Constant pool, Synchronization and Security.
- For String Class can create object two ways: i) Reversing / Constant by using new key word.

String s1 = "abc";

s1 = s1.concat("xyz");

System.out.println(s1);

Q. WAP to reverse String user enter

String str;

String str = "xyzw";

Scanner sc = new Scanner(System.in);

sc.nextLine("enter the String Value");

String s1 = sc.nextLine();

String rev = "";

for (int i = s1.length() - 1; i >= 0; i--) {

{

char c = s1.charAt(i);

rev = rev + ch;

System.out.println(rev);

Length : Length()

instance Variable • Method of String Class

It will be used in • It will be used in

array to return New String • String It return the
no of character in the
of element.

```

W Scanner sc = new Scanner(System.in)
sc.nextLine("Enter String Value");
String s1 = sc.nextLine();
sc.nextLine(new StringBuilder(s1).reverse()
());

```

- Q.1. WAP to reverse the array elements.
- Q.2. WAP to define a Method to
Check the "User" entered String is
Palindrome or not.
- Q.3. WAP to Count how Many Alphabets,
which digits and Special Characters
are Present in the Given String.

```
1 int r = s[1, 2, 3, 4, 7];
```

```
r = reverse(r);
```

```
s.nextLine(Arrays.toString(r));
```

```
Static int[] reverse(int x[]){
```

```
int i = 0, j = x.length - 1;
```

```
while(i < j) {
```

```
int temp = x[i];
```

```
x[i] = x[j];
```

```
x[j] = temp;
```

i = i++, j = -i; and i = 0, j = s.length();
 } and -1

return r;

} in driver class print result.

2. String s1 = "gadag";

if (ispalindrome(s1))

s.o.p("palindrome"); return 1;

else

s.o.p("not");

}

public static boolean ispalindrome (String s1)

{ int i = 0, j = s1.length() - 1;

while (i < j) {

if (s1.charAt(i) != s1.charAt(j))

return false;

i = i + 1; j = j - 1; and i = 0, j = s.length();

}

return true; i = 0, j = s.length();

}

4

String s1 = "gadag";

String rev = new StringBuffer(s1).reverse()

.toString();

if (s1.equals(rev))

s.o.p("Palindrome");

else s.o.p("not");

1. WAP to print the sum of digits in sum.
 2. WAP to count how many consonants and vowels are present in the given String.
- * Count alphabet, number and Special Characters.

String S1 = "abc@12";

```
for (int i=0; i<S1.length(); i++) {
```

```
    char ch = S1.charAt(i);
```

```
    if (ch >= 'A' && ch <= 'Z' || ch >= 'a'
```

```
&& ch <= 'z')
```

```
        ac++;
```

```
    else if (ch >= '0' && ch <= '9')
```

```
        dc++;
```

```
    sc++;
```

```
S.out.println("alpha=" + ac);
```

```
S.out.println("digit=" + dc);
```

```
S.out.println("space=" + sc);
```

1. String S₁ = "abc@123";
 int sum = 0;
 for (int i = 0; i < S₁.length() - 1; i++)
 {

Char ch = S₁.charAt(i);

if (ch >= '0' && ch <= '9')

sum = sum + ch - 48;

}

S₁.openStream();

2. String S₁ = "abAEzfc";

int Vc = 0, Cc = 0;

for (int i = 0; i < S₁.length() - 1; i++)

{

If cur ch = S₁.charAt(i);

if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z')

if (ch == 'a' || ch == 'e' || ch == 'i' ||

ch == 'o' || ch == 'u' || ch == 'A'

|| ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')

{

Vc++;

}

C++;

}

S. o. plu (Vc) : 11 - 3000

S. o. pen(Cc): incorrect

- 1.7.1.1 Impostos e impostas

1. WAP to convert Upper Case Characters into lower case and lower Case Convert into upper.

2. WAP to Count how many Number of words are present in the Given String "77-3446" - 13 BananA

3. WAP to Convert Each Word first Character into Capital Case and Remaining Characters Should be Lower Case.

M	T	W	T	F	S	S
Page No.						
Date:						

1. Scanner sc = new Scanner(System.in);
 sc.nextLine();
 String s = sc.nextLine();

char ch[] = s.toCharArray();

for (int i = 0; i < ch.length - 1; i++)

{

if (ch[i] >= 'A' && ch[i] <= 'Z')

 ch[i] = (char) (ch[i] + 32);

else if (ch[i] >= 'a' && ch[i] <= 'z')

 ch[i] = (char) (ch[i] - 32);

}

s.nextLine(); // Converting ...

s.nextLine(); // Converting ...

2. Scanner sc = new Scanner();

sc.nextLine();

String s = sc.nextLine();

char ch[] = s.toCharArray();

int count = 0;

for (int i = 0; i < ch.length - 1; i++) {

if (i == 0 && ch[i] != ' ' || ch[i] != ' ')

&& ch[i + 1] == ' ')

 count++;

s.nextLine();

Ques: WAP. to convert each word first character into Uppercase and remaining character Should be Lower Case.

String s1 = "rAm iS goOd".

char ch = s1.toCharArray();

for (int i = 0; i < s1.length() - 1; i++)

if (ch[i] >= 'a' & & ch[i] <= 'z')

ch[i] = (char) (ch[i] - 32);

else

ch[i] = (char) (ch[i])

if (ch[i] >= 'A' & & ch[i] <= 'Z')

ch[i] = (char) (ch[i] + 32);

if (ch[i] == ' ' & & count == 0)

ch[i] = (char) (ch[i] - 32);

count++;

ch[i] = (char) (ch[i] - 32);

System.out.println(ch[i]);

System.out.println("Count is " + count);

System.out.println("Total length is " + ch.length());

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

String S = "Am i good or not ?"

char cur[] = S.toCharArray();

```
for (int i=0; i<s.length(); i++)
```

```
if (i==0 && cur[i] == ' ' || cur[i] == ',')
```

```
i = cur[i+1];
```

```
if (cur[i] >='a' && cur[i] <='z')
```

```
cur[i] = (char)(cur[i]+32);
```

```
}
```

```
else if (cur[i] >='A' && cur[i] <='Z')
```

```
cur[i] = (char)(cur[i]+32);
```

```
}
```

```
System.out.println(cur);
```

Q 1. WAP to convert each word odd character in upper & Remaining character will lower

Q 2. WAP to swap each ^{word} odd first character and last character.

1. String str = ram gOD;

char cur[] = s.toCharArray();

: for (int i=0; i<ch.length(); i++)

{

cond. → if (i == ch.length() - 1) ss[cur[i]] =

for last letter { ss[cur[i]] = ch[i]; } ss[cur[i+1]] =

{

if (cur[i] > 'a' && cur[i] < 'z')

: (cur[i] - (char)(cur[i]+32))

{

cond. for → else if (cur[i] > 'A' && cur[i] < 'Z')

remaining letter. cur[i] = (char)(cur[i]+32);

{

System.out.println(cur);

M	T	W	T	F	S	S
Page No.:						
Date:					YOUVA	

2. String S1 = Sam is good.
 char CNE1 = S1.toCharArray();

int f = 0;

for (int i = i < CNE1.length - 1; i++) {

if (i == 0 && CNE1[i] != ' ') {

CNE1[i] = 'S' && CNE1[i + 1] == ' ') {

f = i;

else if (i == ch.length - 1 &&

ch[i] != ' ' && ch[i] != ' ' && CNE1[i + 1] == ' ')

else

CNE1[i] = CNE1[i];

CNE1[i] = ch[i];

CNE1[i + 1] = temp[i];

}

System.out.println(CNE1);

- Q1. WAP to Sort the String character in ascending order.
- Q2. WAP to find the Occurrence or frequency of each character present in the string.
- Q3. WAP to print duplicate element.

```

1. String s = "rameen"
char ch[] = s.toCharArray();
for (int i = 0; i < ch.length - 1; i++) {
    for (int j = i + 1; j < ch.length - i; j++) {
        if (ch[i] > ch[j]) {
            char temp = ch[i];
            ch[i] = ch[j];
            ch[j] = temp;
        }
    }
}

```

S::open ("After Sorting");

String str = new String(ch);

S::open (str);

2. String s = "gadag";

Char [] ch = s.toCharArray();

boolean b [] = new boolean [ch.length];

for (int i = 0; i < ch.length - 1; i++)

if (b[i] == false)

int Count = 1;

for (int j = i + 1; j < ch.length - 1; j++)

if (ch[i] == ch[j])

b[j] = true;

Count++;

}

s.append(ch[i] + " " + Count);

}

3.

String s = "gadag";

Char [] ch = s.toCharArray();

boolean b [] = new boolean [ch.length];

for (int i = 0; i < ch.length - 1; i++)

if (b[i] == false)

int Count = 1;

for (int j = i + 1; j < ch.length - 1; j++)

if (ch[i] == ch[j])

Count++;

b[i] = true;

```

if(count > 1) {
    S.o.p(cu[i]);
}

```

* WAP to Print Unique Characters Present
in a String.

```

String s = "qadag";
char cu[] = s.toCharArray();
boolean br[] = new boolean [ch.length];

for(int i=0; i < ch.length - 1; i++) {
    if(br[i] == false) {
        int Count = 1;
        for(int j=i+1; j < ch.length - 1; j++) {
            if(cu[i] == cu[j]) {
                Count++;
                br[j] = true;
            }
        }
        if(Count == 1) {
            S.o.p(cu[i]);
        }
    }
}

```

Anagram

Public Class Anagram

P SVM (1) 5

Scanner sc = new Scanner("C:\yeten.txt");

S-Open! "Enter two strings!"

String S1 = Scanner.nextLine();

String s2 = sc.nextLine();

boolean $r_2 = \text{isAnagram}(S_1, S_2)$;

if (re == true)

~~Supplementary ("Anagram . . ."):~~

elec

S.o·pen("Not..."):

3

```
public static boolean isAnagram(String s1,  
                               String s2) {
```

char [] ch1 = S1::CstrtoCharArray();

```
char CH1[CH2] = S2.toCharArray();
```

Arrays & sort (C#):

Average Sort (Ch2):

String sa = new String(cng);

~~String s4 = new String(ch2);~~

if(s3.equals(s4))

~~return true;~~

clue

~~return false;~~

* Pangram:

```
Scanner sc = new Scanner(System.in);
```

```
s.open("Enter String");
```

```
String s = sc.nextLine();
```

```
if (isPangram(s))
```

```
s.open("Pangram");
```

```
else
```

```
s.open("not");
```

```
}
```

```
public static boolean isPangram(String s)
```

```
{
```

```
s = s.toLowerCase();
```

```
hashset hs = new hashset();
```

```
for if (s.length() < 26)
```

```
return false;
```

```
for (int i = 0; i < s.length() - 1; i++) {
```

```
char ch = s.charAt(i);
```

```
if (ch >='a' && ch <='z')
```

```
hs.add(ch);
```

```
}
```

```
return hs.size() == 26;
```

* Reverse the Sentence in String
q/p: ram is Good o/p: Good is ram

```

Scanner sc = new Scanner(System.in)
sc.nextLine("Enter the String");
String s = sc.nextLine();

String[] x = s.split(" ");
String rev = "";
for (int i = x.length - 1; i >= 0; i--) {
    rev = rev + x[i] + " ";
}
System.out.println(rev);

```

* WAP to reverse the each Word in the String.

String s: "ram is good";

String m = "";

```

String[] r = s.split(" ");
for (int i = 0; i < r.length - 1; i++) {
    String t = r[i];
    r[i] = r[r.length - 1];
    r[r.length - 1] = t;
}

```

M	T	W	T	F	S	S
Page No.:						
Date:						

String rev = reverse(t);

M = M + rev + " ";

}

s.o.println(m);

}

public static String reverse(String s)

String rev = "";

for (int i = s.length() - 1; i >= 0; i--) {

{

char ch = s.charAt(i);

rev = rev + ch;

}

return rev;

}

W Q. WAP to Print Longest Palindrome Substring in the given String

```
String s = "ababaklkkamda";
```

```
String longest = "";
```

```
int len = 0;
```

```
for (int i = 0; i < s.length() - 1; i++)
```

```
s
```

```
for (int j = i + 1; j < s.length(); j++)
```

```
{
```

```
String sub = s.substring(i, j);
```

```
if (ispalindrome(sub)) {
```

```
if (sub.length() > len)
```

```
{
```

```
longest = sub;
```

```
len = sub.length();
```

```
}
```

```
}
```

```
s.println(longest);
```

```
}
```

```
public static boolean ispalindrome(String s)
```

```
int i = 0, j = s.length() - 1;
```

```
while (i < j)
{
```

```
    if (s.CharAt(i) != s.CharAt(j))
        return false;
```

```
    i++; j--;
```

```
}
```

```
return true;
```

Q. WAP to define a Method to check
the user enter String ie Unique or
not.

```
Scanner Sc = new Scanner(System.in);
Sc.nextLine();
```

```
if (isUnique(s))
```

```
    System.out.println("Unique");
```

```
else
```

```
    System.out.println("not");
```

```
}
```

```
public static boolean isUnique(String s)
```

```
{
```

```
for (int i=0; i<s.length()-1; i++)
{
```

```
    for (int j=i+1; j<s.length(); j++)
    {
```

```
        if (s.charAt(i) == s.charAt(j))
            return false;
```

```
}
```

```
}
```

```
return true;
```

```
}
```

W *

longest WAP to Using Method Length
(Unique Substring:

```
String s = "abcaabmbnbd";
```

```
String Unique = "";
```

```
int len = 0;
```

```
for (int i=0; i<s.length()-1; i++)
{
```

```
    for (int j=i+1; j<s.length(); j++)
    {
```

```
        String sub = s.substring(i, j);
        if (isUnique(sub))
```

```
{
```

```
        if (sub.length() > len)
```

```
{
```

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

len = Sub.length();

Unique = Sub.charAt(0);

}
}

}

s.o.p (Unique);

}
}

Public static boolean isUnique (String s)

{

for (int i = 0; i < s.length() - 1; i++)

{

for (int j = i + 1; j < s.length() - 1; j++)

{

if (s.charAt(i) == s.charAt(j))

}
return false;

}

return true;

}

2-Dimensional Array

Collection of 1-D Array is called 2D Array.

Syntax for 2D Array:

arraytype / Data type ArrayName [] [];

ex: int arr [] []

double arr [] []

```
int arr [ ] [ ] = { { 1, 2, 3 }, { 4, 5, 6 },
                     { 7, 8, 9 } }
```

S. o. p(arr.length) // 3 S. o. p(arr[1].length)

S. o. p(arr[0][1]) // 2

S. o. p(arr[2][2]) // 9

- Q. WAP to Print 2D array elements in Row and Column wise.

```
int a = { { 10, 20, 30 }, { 30, 40, 50 },
          { 60, 70, 80 } }
```

U T N T E S S	
Page No.	YEAR

```

for (int i=0; i<a.length-1; i++) {
    for (int j=0; j<a[i].length; j++) {
        s.o.p(a[i][j] + " ");
    }
    s.o.p();
}
    
```

No. of rows: 2D array length.

No. of Columns: 1D array length.

Q. WAP to Print Prime elements in 2D array.

```
int a[7][7] = { {5, 6, 87}, {9, 11, 15, 77} }
```

```

for (int i=0; i<a.length-1; i++) {
    for (int j=0; j<a[i].length; j++) {
        if (isprime(a[i][j])) {
            s.o.p(a[i][j] + " ");
        }
    }
}
    
```

Output: 5 11 77
{ 15 is prime number}

```

public static boolean isprime( int arr[])
{
    if( a <= 1) return false;
    for( int i = 2 ; i <= a/2 ; i++)
    {
        if( arr[i] == 0)
            return false;
        return true;
    }
}

```

Q. WAP to print sum of diagonal elements

int arr[] = { 10, 20, 30, 40, 50, 60, 70, 80, 90 }

```

int pd = 0
int sde = 0;

for( int i = 0 ; i <= a.length - 1 ; i++)
{
    for( int j = 0 ; j <= a[i].length - 1 ; a[i][j] + a[i][j])
    {
        if( i == j)
            pd = pd + a[i][j];
        if( i + j == a.length - 1 )
            sde = sde + a[i][j];
    }
}

System.out.println(pd);
System.out.println(sde);

```

Do while Loop

q+ is a looping statement which used to execute some set of statement repeatedly.

Syntax: do {

 list
}

 updation

} while (condition);

WAP to print all even no present bit 1 to 20. using do while loop.

int i = 1; n = 20;

do {

 if (i % 2 == 0)

 S.o.p(i);

 i++;

} while (i <= n);

while

do-while

(i) Every Control Loop (ii) Exits Control Loop

(iii) Minimum Iteration (iv) Minimum Iteration
is '0' is '1'

(v) Doesn't allow iteration at the end.

(vi) q+ allows iteration at the end.

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

(iv) First it will check the condition. If the condition is true then the loop will execute once. Then it will check if the condition is true then it will execute.

* For Each Loop.

It is a looping statement which is used to execute set of statements repeatedly.

For each loop is the advanced loop for "for loop".

Using for each loop we can iterate in only forward direction.

Using for each loop we cannot iterate in backward direction.

For each loop is used to iterate in array and also Collection.

Syntax: `for< type varName : ArrayName >`

`s`

`|| Stmt`

`f`

ex: `int arr = {10, 20, 30};`

`for (int i = 0; i < 3; i++)`

`s.o.p(arr[i]);`

7

* Bitwise Operators:

Bitwise Operators is a type of Operators which works bit by bit.

Type of:

1) Bitwise AND (&)

2) Bitwise OR (|)

3) Bitwise XOR (^)

* Bitwise AND :: it return true iff both arguments are true Otherwise it return false

* Bitwise OR (|) :: it return true iff only one of the argument is true Otherwise it return false.

* Bitwise XOR (^) :: it returns true iff both the arguments are different
Otherwise it return false.

ex: `s.o.p(4 & 5);` 1100

`s.o.p(4 | 5);` 1101

`s.o.p(4 ^ 5);` 1011

`s.o.p(7 ~ 4);` 1100

`s.o.p(~4);` 1111

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

* Shift Operators:

1> Left Shift Op - ($<<$)

2> Right Shift Op - ($>>$)

s.o.p(2 $<<$ 2); 110000000010

110000000010

Left Shift: The Left Shift Operator moves all bits by a given number of bits to the left.

Right Shift: The Right Shift Operator moves all bits by a given number of bits to the right.

s.o.p(14 $<<$ 2) 1156

s.o.p(16 $>>$ 2) 114

Q. WAP to define a method to find the factorial for the given number using Recursion.

Q. WAP to define a method to Count how many digits are present in the given number Using Recursion.

Class Test 5

Program 5

int n = 4;

int f = fact(n);

cout << f << endl;

public static int fact (int n)

{

if (n == 0)

return 1;

else

return (n * fact(n - 1));

}

}