

Array

- Array is a collection of homogeneous elements/
data value
- Array is an object in java
- Array will store the element / data based on
the index value
- Array is considered as non-primitive datatype
- default value of array is null

Advantages

- It stores the data in continuous memory all at once
- Searching operation is easy
- Retrieval operation is easy bcs on index value we
can get the elements.

Disadvantages

- Array size is fixed
- we can't store heterogeneous elements
- Insertion and deletion operations are difficult
- Array doesn't have any pre-defined methods to
deal with elements.

Declaration of array

Array type/ Array name [];
datatype

int a[]; ✓

double b[]; ✓

int[] a; ✓

[] char c; ✗

Syntax for array Initialization

Array type / Arrayname[] = new Array type [size];
 Data type

Ex: int a[5] = new int[4];

Syntax for array element initialization

Array name [index] = data value; ^{at index}

a[0] = 10;

a[1] = 20;

a[2] = 30;

a[3] = 40;

a[4] = 50; // XTE (AIOBE)

a =	<table border="1"> <tr> <td>10</td><td>20</td><td>30</td><td>40</td></tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td></tr> </table>	10	20	30	40	0	1	2	3	Size - 1
10	20	30	40							
0	1	2	3							

Q1 WAP to find the sum of even elements in the array.

Q2 WAP to print sum of odd index element.

Q3 WAP to find the biggest element in the array.

Q4 WAP to print Smallest element in the array.

Q5 largest nos problem

class Test{
public:
int n;}

int a[] = new int[3];

a[0] = 15;

a[1] = 10;

a[2] = 12;

for (int i = 0; a[i] < a.length - 1; i++)

{ if (a[i] % 2 == 0)

sum += a[i]; } System.out.println(sum); }

② Class Test

PSUM CSCJA

{
int arr = new int[4];
arr[0] = 5;
arr[1] = 10;
arr[2] = 15;
arr[3] = 20;

Output :-

int sum = 0;

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

if (arr[i] != 0)

sum = sum + arr[i];

System.out.println(sum);

④ Class Test

PSUM CSCJA {

int arr = {10, 20, 30, 40, 50};

int small = arr[0];

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

if (arr[i] < small)

small = arr[i];

}

System.out.println(small);

}

③ Class Test

PSUM CSCJA {

int arr = {10, 20, 30, 40, 50};

int big = 0;

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

if (arr[i] > big)

if (arr[i] > big)

big = arr[i];

System.out.println(big);

⑥ WAPT print all the prime elements in the user entered array

⑦ WAPT Count how many palindrome elements are present in the user entered array.

⑧ WAPT print sum of all the armstrong no.s present in the user entered array.

Let me use it for

arr = {10}

10 = 10

10 = 10

10 = 10

10 = 10

Q6APT Take array input from the user and print the sum of all the array elements.

```

import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array length");
        int n = sc.nextInt();
        int a[] = new int[n];
        System.out.println("Enter the array Elements");
        for (int i = 0; i <= a.length - 1; i++) {
            a[i] = sc.nextInt();
        }
        int sum = 0;
        for (int i = 0; i <= a.length - 1; i++) {
            sum = sum + a[i];
        }
        System.out.println(sum);
    }
}

```

Q6APT: print all non prime elements with their references in array.

```

import java.util.Scanner;
public class A6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array length");
        int n = sc.nextInt();
        int a[] = new int[n];
        System.out.println("Enter the array Elements");
        for (int i = 0; i <= a.length - 1; i++) {
            a[i] = sc.nextInt();
        }
        for (int i = 0; i <= a.length - 1; i++) {
            if (isPrime(a[i])) {
                System.out.println(a[i]);
            }
        }
    }
}

public static boolean isPrime(int num) {
    if (num < 2) {
        return false;
    }
    for (int i = 2; i <= num / 2; i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}

```

```

    System.out.println("Enter the array size");
    int size = sc.nextInt();
    int a[] = new int[size];
    System.out.println("Enter the array Elements");
    for (int i=0; i<a.length; i++) {
        a[i] = sc.nextInt();
    }
    for (int i=0; i<a.length-1; i++) {
        if (isprime(a[i])) {
            System.out.println(a[i]);
        }
    }
    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;
    }

```

③ Write a program to count how many palindrome elements present in the user entered array.

```

import java.util.Scanner;
public class ATE {
    public static int sum (int a) {
        Scanner sc = new Scanner (System.in);

```

```

sopln("Enter the array size");
int size = sc.nextInt();
int a[] = new int[size];
sopln("Enter the array elements");
for(int i=0; i<a.length; i++)
{
    a[i] = sc.nextInt();
}
int count=0;
for(int i=0; i<a.length; i++)
{
    if(isPallindrome(a[i]))
    {
        count++;
    }
}
sopln(count);
}

static boolean isPallindrome(int n)
{
    int num=n;
    int reverse=0;
    while(n>0)
    {
        int rem = n%10;
        reverse = reverse*10+rem;
        n /= 10;
    }
    return num==reverse;
}

```

- ⑧ WAP to print all the armstrong numbers present in the user entered array.

```

import java.util.Scanner;
class A8
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int size=Integer.parseInt(sc.nextLine());
        int a[]=new int[size];
        System.out.println("Enter the array Element");
        for(int i=0; i<a.length; i++)
        {
            a[i]=sc.nextInt();
        }
        int sum=0;
        for(int i=0; i<a.length; i++)
        {
            if(isArmstrong(a[i]))
            {
                sum+=a[i];
            }
        }
        System.out.println(sum);
    }
}

static boolean isArmstrong(int n)
{
    int num=n;
    int sum=0;
    int c=CountDigit(n);
    while(c>0)
    {
        int rem=n%10;
        sum+=(int) Math.pow(rem,c);
        n/=10;
    }
    return sum==num;
}

```

if true {
 return sum==num;
}

```

Static int CountDigit (int n)
{
    int count = 0;
    while (n > 0)
    {
        int rem = n % 10;
        count++;
        n /= 10;
    }
    return count;
}

```

Q) WAPT Replace every element by their sum of each digit

```

import java.util.Scanner;
class A {
    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the size of an array");
        int size = sc.nextInt();
        int a [] = new int [size];
        System.out.println ("Enter the " + 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++)
        {
            int rec = SumOfDigit (a[i]);
            a[i] = rec;
        }
        System.out.println (Array.toString (a));
    }
}

```

```

static int sumofdigit(Cint n) {
    int sum = 0;
    while (n > 0) {
        int rem = n % 10;
        sum += rem;
        n /= 10;
    }
    return sum;
}

```

- ⑩ WAP to print occurrence frequency of each element in the user entered array.

```

import java.util.Scanner;
class A10 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the size of the array");
        int size=sc.nextInt();
        int a[] = new int[size];
        System.out.println("Enter the " + size + " array elements");
        for (int i=0; i<a.length-1; i++) {
            a[i]=sc.nextInt();
        }
        int count=0;
        for (int i=0; i<a.length-1; i++) {
            for (int j=0; j<a.length-1; j++) {
                if (a[i]==a[j])
                    count++;
            }
        }
    }
}

```

```

System.out.println(Arrays.toString(a));
boolean[] visit = new boolean[size];
for (int i=0; i<a.length; i++) {
    if (!visit[i]) {
        int count=1;
        for (int j=i+1; j<a.length; j++) {
            if (a[i]==a[j])
                count++;
            visit[j]=true;
        }
        System.out.println(a[i]+": "+count);
    }
}
sc.close();

```

- ⑩ WAP to print unique elements in the user entered array.

```

import java.util.Arrays;
import java.util.Scanner;
public class Unique {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the size of the array");
        int size=sc.nextInt();
        int a[]=new int[size];
        System.out.println("Enter "+size+" array elements");
        for (int i=0; i<a.length; i++) {
            a[i]=sc.nextInt();
        }
        System.out.println(Arrays.toString(a));
        boolean[] visited = new boolean[size];
    }
}

```

```

for (int i=0; i<a.length-1; i++) {
    if (C5.visited[i]) {
        int count = 1;
        for (int j=i+1; j<a.length; j++) {
            if (a[i] == a[j]) {
                count++;
                visited[j] = true;
            }
            if (count == 1) {
                System.out.println(a[i]);
            }
        }
        sc.close();
    }
}

```

(12) Write a program to print duplicate elements in a user entered array.

```

public class duplicates {
    public static void main (String args[]) {
        Scanner sc=new Scanner (System.in);
        System.out.println ("Enter the array size");
        int size=sc.nextInt();
        int a[] = new int [size];
        System.out.println ("Enter " + size + " array elements");
        for (int i=0; i<a.length-1; i++) {
            a[i]=sc.nextInt();
        }
        System.out.println (Arrays.toString (a));
        boolean [] visit=new boolean [size];

```

```

    for(int j=i+1; j<a.length; j++) {
        if(a[i] == a[j]) {
            count++;
            visit[j]=true;
        }
    }
    if(count > 1) {
        System.out.println(a[j]);
    }
    sc.close();
}
}

```

⑩ WAP to print missing elements in the user entered array.

```

import java.util.Arrays;
import java.util.Scanner;
class A13 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the size of an array");
        int n=sc.nextInt();
        int a=new int[n];
        System.out.println("Enter the array element");
        for(int i=0; i<n; i++) {
            a[i]=sc.nextInt();
        }
        int min=a[0];
        int max=a[0];
        for(int i=0; i<n; i++) {
            if(a[i]<min)
                min=a[i];
            if(a[i]>max)
                max=a[i];
        }
        for(int i=0; i<n; i++) {
            if(a[i]==min)
                if(a[i]==max)
                    System.out.println(i);
                else
                    System.out.println(min);
            else
                System.out.println(max);
        }
    }
}

```

```

        static boolean isMissingElement(int[] arr, int n, int m) {
            if (max < ac[i])
                max = ac[i];
            }
            Array c.sort (ca);
            int cnt=0;
            sopln (" Missing elements are "); return true;
            for (int i=min; i<=max; i++) {
                if (ac[i] != i)
                    sopln (i);
                else
                    cont++;
            }
            if (cont == m)
                return true;
            else
                sopln (" Missing elements are ");
        }
    }
}

```

(14) Write a class to define a method to merge two arrays entered array into single array.

```

class Merge {

```

```

    public int[] merge() {

```

```

        Scanner sc=new Scanner (System.in);
        int s1=sc.nextInt();

```

```

        int s2=sc.nextInt();
        int ar1=new int [s1];

```

```

        int ar2=new int [s2];
        for (int i=0; i<s1; i++)

```

```

            ar1[i]=sc.nextInt();
            ar2[i]=sc.nextInt();
        }
    }

```

```

        int s3=s1+s2;
        int ar3=new int [s3];
        for (int i=0; i<s1; i++)

```

```

            ar3[i]=ar1[i];
            ar3[i+s1]=ar2[i];
        }
    }

```

```

        for (int i=0; i<s3; i++)
            ar3[i]=sc.nextInt();
    }
}

```

```

        int s4=sc.nextInt();
        int ar4=new int [s4];
        for (int i=0; i<s3; i++)

```

```

            ar4[i]=ar3[i];
            ar4[i+s3]=sc.nextInt();
        }
    }
}

```

```

        for (int i=0; i<s4; i++)
            ar4[i]=sc.nextInt();
    }
}

```

```

        int s5=sc.nextInt();
        int ar5=new int [s5];
        for (int i=0; i<s4; i++)

```

```

            ar5[i]=ar4[i];
            ar5[i+s4]=sc.nextInt();
        }
    }
}

```

```

    int a2 = new int[s2];
    for(int i=0; i<a2.length; i++)
        a2[i] = sc.nextInt();
    int[] rep = sum(a1, a2, s1, s2);
    System.out.println("A naya array is");
    for(int i=0; i<rep.length; i++)
        System.out.print(rep[i] + " ");
}
public static int[] sum(int[] a1, int[] a2, int s1, int s2) {
{
    int[] rep = new int[s1+s2];
    for(int i=0; i<s1; i++)
        rep[i] = a1[i];
    for(int i=0; i<s2; i++)
        rep[s1+i] = a2[i];
    System.out.println("Sum of two arrays is");
    for(int i=0; i<rep.length; i++)
        System.out.print(rep[i] + " ");
}
}

```

④ WAP to swap user entered two numbers

```

class A15 {
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        int n2 = sc.nextInt();
        System.out.println("Enter the second element");
        int temp = n1;
        n1 = n2;
        n2 = temp;
        System.out.println("Swapping 2 no's");
        System.out.println(n1 + " " + n2);
    }
}

```

⑤ WAP to swap user entered two numbers

```

class A15 {
    public static void main() {
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        int n2 = sc.nextInt();
        System.out.println("Enter the second element");
        int temp = n1;
        n1 = n2;
        n2 = temp;
        System.out.println("Swapping 2 no's");
        System.out.println(n1 + " " + n2);
    }
}

```

⑯ WAPT swap the user entered 2 no.p without using temp

class A16E

p s v m (sc[ja]) {

Scanner sc=new Scanner (System.in);

sopln ("Enter 2no.p")

int n1= sc.nextInt();

int n2= sc.nextInt();

$$n1 = n1 + n2; \quad 10 + 20 = 30$$

$$n2 = n1 - n2; \quad 30 - 20 = 10$$

$$n1 = n1 - n2; \quad 30 - 10 = 20$$

sopln (n1 + " " + n2);

}

⑰ WAPT to sort the user entered array elements in asc order without using inbuilt function

class A17E

p s v m (sc[ja]) {

Scanner sc=new Scanner (System.in);

sopln ("Enter the size of the array");

int size= sc.nextInt();

int a[] = new int [size];

sopln ("Enter the array elements");

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

a[i]= sc.nextInt();

}

sopln ("Sorting array");

{}

```

for (int i=0; i<a.length; i++)
{
    for (int j=0; j<a.length; j++)
    {
        if (a[j] < a[i])
        {
            int temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    System.out.println("After sorting at this step");
    System.out.println(Arrays.toString(a));
}

```

Q) WAP to enter array element in desc

```

class A18
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size of the array");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.println("Enter the array element");
        for (int i=0; i<a.length; i++)
        {
            a[i] = sc.nextInt();
        }
        System.out.println("Sorting array");
        for (int i=0; i<a.length; i++)
        {
            for (int j=i+1; j<a.length; j++)
            {
                if (a[i] < a[j])
                {
                    int temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        System.out.println("After sorting at this step");
        System.out.println(Arrays.toString(a));
    }
}

```

```

if (a[i] < a[j])
{
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}

System.out.println(Arrays.toString(a));
}
}

```

19) What point the pair elements whose sum is equal to given number.

P 8 VM CSC(JA)

```

int arr = {1, 4, 0, 2, 5, 6, 3}; bookon size same
int n = 6;
boolean b[] = new boolean[a.length];
for (int i = 0; i < a.length; i++) {
    if (b[i] == false) {
        for (int j = i + 1; j <= a.length - 1; j++) {
            if (a[i] + a[j] == n) {
                System.out.println(a[i] + " " + a[j]);
                b[j] = true;
            }
        }
    }
}
    
```

(Q) WAP to print first unique element in the array.

for loop

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

Scanner sc = new Scanner (System.in);

sopln("Enter the size of the array");

int size = sc.nextInt();

int [] a = new int [size];

sopln("Enter the array element");

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

a[i] = sc.nextInt();

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

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

if (a[i] == a[j]) {

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

break;

if (a[i] < a[j]) {

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

break;

if (a[i] > a[j]) {

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

break;

if (a[i] == a[j]) {

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

break;

if (a[i] < a[j]) {

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

break;

if (a[i] > a[j]) {

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

break;

if (a[i] == a[j]) {

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

Q1 print the user entered 2 array in a single array in zigZag format

```

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the 1st array size");
        int size1 = sc.nextInt();
        int a[] = new int[size1];
        sc.nextLine();
        System.out.print("Enter the 2nd array size");
        int size2 = sc.nextInt();
        int b[] = new int[size2];
        sc.nextLine();
        for (int i = 0; i < a.length; i++)
            a[i] = sc.nextInt();
        for (int i = 0; i < b.length; i++)
            b[i] = sc.nextInt();
        int res[] = zigZag(a, b);
        System.out.println(Arrays.toString(res));
    }

    static int[] zigZag(int a[], int b[]) {
        int in = 0;
        int c[] = new int[a.length + b.length];
        for (int i = 0; i <= a.length - 1 / 2; i++) {
            if (i <= a.length - 1)
                c[in] = a[i];
            in++;
        }
        if (i <= b.length - 1)
            c[in] = b[i];
        return c;
    }
}
  
```

$c[i][j] = b[i][j]; \quad i = \text{int} \times 3 + \text{int}$

$\text{int}++;$

10	20	30
0	1	2

100	200	300	400
0	1	2	3

100	200	300	400
0	1	2	3

$\text{return } c;$

$i=0 \quad 0 \leq i < 3 \quad i=1 \quad 1 \leq i < 3 \quad i=2 \quad 2 \leq i < 3 \quad i=3 \quad 3 \leq i < 3 \quad i=4 \quad 4 \leq i < 3$
$0 \leq i < 2 \quad c[2] = a[1] \quad c[4] = a[2]$
$c[0] = a[0] \quad 0 \quad 10 \quad \text{int}++ \quad 0 \quad 30 \quad \text{int}++ \quad 0 \quad 300 \quad \text{int}++ \quad 0 \quad 400 \quad \text{int}++$
$0 \leq i < 3 \quad c[1] = b[0] \quad 0 \quad 100 \quad \text{int}++ \quad 0 \quad 200 \quad \text{int}++ \quad 0 \quad 300 \quad \text{int}++ \quad 0 \quad 400 \quad \text{int}++$
$c[5] = b[2] \quad 0 \quad 300 \quad \text{int}++ \quad 0 \quad 400 \quad \text{int}++ \quad 0 \quad 500 \quad \text{int}++ \quad 0 \quad 600 \quad \text{int}++$

$(i = 0, 0 \leq i < 3 - 0) \text{ if } \text{not } c[1] = b[0] \text{ then }$

class Automorphic {

public void auto() { int n = 76; if (isAuto(n));

System.out.println("Automorphic"); else sopIn("Not automorphic");

else

sopIn("Not automorphic");

static boolean isAuto(int n)

int sq = n * n;

while (n > 0)

int rem1 = n % 10;

int rem2 = sq % 10;

if (rem1 == rem2)

return true;

n / 10;

sq / 10; return true; } }

String

- ① WAP to reverse the user entered string.

Class S12

psvm (SCJ0)

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

sc.nextLine();

String str = sc.next();

String rev = "";

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

{

rev += str.charAt(i);

}

System.out.println("Reversed string : " + rev);

}

- ② WAP to count how many vowels and consonants are present in the string

Class S2

psvm (SCJ0)

{

Scanner sc=new Scanner (System.in);

sc.nextLine();

String str = sc.next();

int vc=0;

int cc=0;

String star=str.toUpperCase();

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

```

char ch = str1.charAt(i);
if (ch >= 'A' & & ch <= 'Z')
{
    if (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' ||
        ch == 'U')
        uc++;
    else
        cc++;
}
System.out.println("Number of vowels and consonants are present in the string ");

```

③ Write a program to count how many alphabets, digits and special characters are present in the given string.

```

class S3
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the String:");
        String Str = sc.nextLine();
        String str1 = Str.toUpperCase();
        int A = 0, D = 0, S = 0;
        for (int i = 0; i < str1.length() - 1; i++)
        {
            char ch = str1.charAt(i);
            if (ch >= 'A' & & ch <= 'Z')
                A++;
            else if (ch >= '0' & & ch <= '9')
                D++;
            else
                S++;
        }
        System.out.println("Number of uppercase letters are " + A);
        System.out.println("Number of digits are " + D);
        System.out.println("Number of special characters are " + S);
    }
}

```

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

```
D++;
```

```
else
```

```
& S++;
```

```
}
```

sopln(A+)"Alphabets are present in the given String");

sopln(D+)"Digits are present in the given String");

sopln(S+)"Special characters are present in the given

String");

- * String is a predefined final class present in java.lang package since JDK 1.0.

↳ String is considered as collection of characters.

↳ String is also index based.

- * String is immutable in nature because of Constant pool, Security & Synchronization purpose.

- * String has 15 overloaded Constructors & 60 methods.

- * We can create object of String in two ways

1. Literal / Constant way

2. new keyword

- * From Object class 3 methods are overridden in the String class.

1. toString()

2. equals()

3. hashCode()

```

String s1 = "ab"
String s2 = "ab";
sopln(s1 == s2); // false & true
sopln(s1.equals(s2)); // true
String s1 = 'abcd'
String s2 = "abc";
sop(s1 == s2); // false
sop(s1.equals(s2)); // false
    
```

```

String s1=new String("abc")
String s2=new String("abc");
sopln(s1==s2); // false
sopln(s1.equals(s2)); // true
    
```

```

SBuffer s1=new SBuffer("abc");
SBuffer s2=new SBuffer("abc");
sopln(s1==s2); // false
sopln(s1.equals(s2)); // false
    
```

String

- * It is class present since JDK 1.0V

- * It is immutable in nature

- * we can create object in 2 ways

- * multithreaded

- * methods are not synchronized

- * performance is slow because immutable in nature

- * It consumes more memory b/t of immutable

- * String class has overridden methods.

1. toString()

2. equals()

3. hashCode()

String Buffer

- * It is a class present since JDK 1.0V

- * It is immutable in nature

- * we can create object in 1 way

- * Single threaded

- * methods are synchronized

- * performance is slow because of single threaded

- * It consumes less memory b/t of mutable

- * String Buffer only one method is overridden

from object class

..toString

String Builder

- * It is class present since JDK 1.5 V

- * It is immutable in nature.

- * we can create object in 1 way.

- * Single threaded

- * methods are not synchronized

- * performance is fast because of multithreaded & mutable in nature.

- * it consumes less memory b/t of mutable

- * Thread safe
- * Thread safe
- * Not thread safe
- * "+" operator will not act as a concatenation o/p
- * "+" operator will not act as a concatenation o/p
- * "+" operator will not act as a concatenation o/p

- * If we want to perform concatenation we will use concat() method
- * If we want to concat 2 Buffer we will append
- * If we want to concat 2 String Builder we will use append

- * reverse() method is present
- * reverse() method is present
- * reverse() method is present

length

- * instance variable
- * It returns the no. of elements present in the array buffer
- * pre-defined method present in the String class
- * It returns the no. of characters present in the string.

Conversion of String to String Builder

```
String s = "abc";
StringBuffer sb = new StringBuffer(s);
```

```
sb.append("xyz");
```

```
sopln(sb);
```

Conversion of String Builder to String

```
StringBuffer sb = new StringBuffer("abc");
```

```
String s = sb.toString();
```

```
sopln(s);
```

```
class reversing {
    public static void main (String [] args) {
        String s = "abc";
        String rev = new StringBuffer (s).reverse().toString();
        System.out.println (rev);
    }
}
```

④ WAP to print sum of digit present in the string.

```
class sum {
    public static void main (String [] args) {
        String s = "abc@123";
        int sum = 0;
        for (int i = 0; i < s.length() - 1; i++) {
            char ch = s.charAt(i);
            if ('0' <= ch & ch <= '9')
                sum += Character.getNumericValue(ch);
        }
        System.out.println (sum);
    }
}
```

14/12/24

⑦ WAP/T convert uppercase characters into lowercase and lowercase into uppercase.

class Test { public static void main(String[] args) {

String s = "AbCd";

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);

} } } → System.out.println(ch);

⑧ WAP/T convert each word first char to uppercase and remaining character in lowercase

class S8 {

public static void main(String[] args) {

String s = "ram is good";

char ch[] = s.toCharArray();

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

{

if (i == 0 && ch[i] != ' ') || ch[i] != ' ' && ch[i - 1] == ' ')

{

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);

}

sopln(ch);

}

(Hindi-alphabets = a, e, i, o, u)

- Q) WAP to convert each word kept char to upper case and remaining character into lower case.

class Sqr{

public static void main(String args){

String s= "ram is good";

char ch[] = s.toCharArray();

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

{

if (ch[i] == ch.length-1 && ch[i] != 'D') {
ch[i] = ' ' + ch[i+1];

} else 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);

}

sopln(ch);

- D) WAP to print occurrence or frequency of each character present in the string.

class Demo{

public static void main(String args){

String s= "chand";

char ch[] = s.toCharArray();

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

```

for(int i=0; i<=ch.length-1; i++)
{
    if (bc[i] == false)
        int count = 1;
    {
        for(int j=i+1; j<=ch.length; j++)
        {
            if (ch[i] == ch[j])
                count++;
            bc[j] = true;
        }
        System.out.println(ac[i] + "=" + count);
    }
}

```

(ii) write print unique characters present in the given string.

Code Demo

```

public class UniqueChar
{
    public static void main(String[] args)
    {
        String s = "Chandini";
        char ch[] = s.toCharArray();
        boolean bc[] = new boolean [ch.length];
        for(int i=0; i<ch.length-1; i++)
        {
            if (bc[i] == false)
                int count = 1;
            for(int j=i+1; j<ch.length; j++)
            {
                if (ch[i] == ch[j])
                    count++;
                bc[j] = true;
            }
            System.out.println(ch[i] + "=" + count);
        }
    }
}

```

if (count == 1)
 sopln(ch[i] + " = " + count);
 break;
} } }
⑩ write print duplicated character present in the
given string.

class Demo

public class Demo {

String s = "Taehyung";

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[j] = true;

}

if (count > 1)

 sopln(ch[i] + " = " + count);

}

⑪

```

    public static int sum(String s) {
        String s = "ram is good"; int f=0;
        char ch[] = s.toCharArray();
        for(int i=0; i<ch.length-1; i++)
            if(i==0 && ch[i] != ' ') || (ch[i] != ' ' && ch[i+1] == ' ')
                f=i;
        else if(i==ch.length-1 && ch[i] != ' ') || (ch[i] != ' ' && ch[i+1] == ' ')
        {
            char temp = ch[f];
            ch[f] = ch[i];
            ch[i] = temp;
        }
    }
}

```

(14) ~~INPUT~~ define a method to check whether a string is a panagram or not
~~13/12/24~~ A panagram is any string that contains all letters of a given alphabet set at least once.

Ex: "abcdefghijklmnopqrstuvwxyz"

Output: true

```

    public static boolean isPanagram(String str) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the String");
        String str=sc.nextLine();
        if(isPanagram(str))
            System.out.println("panagram");
        else
            System.out.println("Not a panagram");
    }
}

```

```

static boolean isPanagram (String s)
{
    HashSet hp = new HashSet();
    for (int i=0; i<s.length()-1; i++)
    {
        char ch = s.charAt(i);
        if ((ch>='A' && ch<='Z') || (ch>='a' && ch<='z'))
            hp.add(ch);
    }
    if (hp.size() == 26)
        return true;
    else
        return false;
}

```

(15) ~~WAP~~ define a method to check two strings ~~are~~ Anagram ~~or~~ not

class S15{

```

public static void main (String [] args)
{
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter the first string");
    String s1 = sc.nextLine();
    System.out.println ("Enter the second string");
    String s2 = sc.nextLine();
    if (isAnagram (s1, s2))
        System.out.println ("Anagram");
    else
        System.out.println ("Not a anagram");
}

```

```

static boolean isAnagram (String s1, String s2)
{
    if (s1.length () != s2.length ())
        return false;
    char [] ch1 = s1.toCharArray();
    char [] ch2 = s2.toCharArray();
    Arrays.sort(ch1);
    Arrays.sort(ch2);
    for (int i=0; i<ch1.length; i++)
        if (ch1[i] != ch2[i])
            return false;
    return true;
}

```

```
Arrays.sort(ch1);
Arrays.sort(ch2);

String s3 = new String(ch1);
String s4 = new String(ch2);

if(s3.equals(s4))
    return true;
else
    return false;
}
```

Q6) find first unique character

class S16

```
{
```

```
String s = "chandini.R";
char ch[] = s.toCharArray();
boolean bc[] = new boolean[ch.length];
for(int i=0; i<ch.length-1; i++)
{
    if(bc[i] == false)
    {
        int count = 1;
        for(int j=i+1; j<ch.length-1; j++)
        {
            if(ch[i] == ch[j])
            {
                count++;
                bc[j] = true;
            }
        }
        if(count == 1)
        {
            System.out.println(ch[i] + "=" + count);
            break;
        }
    }
}
```

(17) WAP to find the occurrence of characters present in the string using hash map

```
class S17 {
    public static void main(String args) {
        String s = "chandini. & ";
        HashMap<Character, Integer> hm = new HashMap<>();
        for (int i = 0; i < s.length() - 1; i++) {
            char ch = s.charAt(i);
            if (hm.containsKey(ch))
                hm.put(ch, hm.get(ch) + 1);
            else
                hm.put(ch, 1);
        }
        System.out.println(hm);
    }
}
```

(18) WAP to define a method to check the given string is unique or not

```
class S18 {
    public static void main(String args) {
        String s = "Taehyung";
        if (isUnique(s))
            System.out.println("String is unique");
        else
            System.out.println("String is not unique");
    }
}
```

```

static boolean isUnique(String s)
{
    HashSet<Character> set = new HashSet();
    for (int i=0; i<s.length()-1; i++)
    {
        char ch = s.charAt(i);
        if (!set.add(ch))
            return false;
    }
    return true;
}

```

print all the substring of string

```

class S
{
    public static void main (String args[])
    {
        String s = "ababaca";
        for (int i=0; i<s.length()-1; i++)
        {
            for (int j=i+1; j<s.length(); j++)
            {
                String sub = s.substring (i, j);
                System.out.println (sub);
            }
        }
    }
}

```

⑨ Longest palindromic substring

```

class S
{
    public static void main (String args[])
    {
        String longest = "";
        int len = 0;
    }
}

```

```
String s = "ababaca";
```

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

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

```
        string sub = s.substring(i, j)
```

```
//sopln(sub); x
```

```
{  
    if (isPall(sub))
```

```
{  
    //sopln(sub);
```

```
if
```

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

```
{
```

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

```
longest = sub;
```

```
} } }
```

```
sopln(longest);
```

```
}
```

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

```
{
```

```
String rev = new StringBuffer (s).reverse().toString();
```

```
if (rev.equals(s))
```

```
return true;
```

```
else
```

```
return false;
```

```
}
```

* print prime no. from 10 to 20

write the code to replace the space with a

Ex: S-A-C-H-I-N

```
class S{  
    public static void main (String args[]){  
        for(int i=10; i<=20; i++)  
        {  
            if (isPrime (i))  
                System.out.println (i);  
        }  
    }  
    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;  
    }  
    String s="S A C H I N";  
    String r="";  
    for (int i=0; i<s.length()-1; i++)  
    {  
        if (s.charAt(i) == ' ')  
            r=r+"A";  
        else  
            r=r+s.charAt(i);  
    }  
    System.out.println (r);  
}
```

② Longest unique Substring

class S20{

public String(s)(a){

String s = "abcaadm";

String & uniq = "";

int len=0;

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

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

{ string sub = s.substring(i, j);

if(isunique(sub))

{

if(sub.length() > len)

{

len = sub.length();

uniq = sub;

}

}

sopln(uniq);

}

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))

{

 if(s.charAt(i) == s.charAt(j))

 isunique = false;

}

return true;

Q1 Reversing a string Ram is good
class S21{

ps um(CS21){

String s= "ram is good";

String arr[] = s.split(" ");

String rev= "";

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

rev= rev+arr[i]+ " ";

}

sopln (rev);

}

Q2 Ram is good

o/p: m a r e d o o g

class S22{

ps um(CS22){

String s= "ram is good";

String arr= s.split(" ");

String rev= "";

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

{ String rev= reverse(arr[i]);

rev= rev+arr[i];

}

sopln (rev);

}

static bo String reverse (String s){

{ String rev= new String(s).reverse().toString();

}

2D Array

→ 2D array means collection of 1D arrays called 2D array.

Syntax: Array type [] [] array. datatype int [] [] a;

datatype int [] [] b;

int [] [] c;

int [] [] d;

int a [] [] = { { 10, 20, 30 }, { 40, 50, 60 }, { 70, 80, 90 } };

sopln (a [1]. length) // 3

sopln (a [0] [1]) // 20

sopln (a [1] [2]) // 60

sopln (a [2] [0]) // 40

② WAP to print 2D array elements row wise and column wise

class A :

{ sum (sc [0])

int a [] [] = { { 10, 20, 30 }, { 30, 40, 50 }, { 60, 70, 80 } }

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

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

{ sop (a [i] [j]);

{ sopln ();

{ sopln ();

{ } } } } }

② WAPT print prime elements in the 2d array

```
class A2{  
    public static void main(String args){  
        ps.println("prime elements in the 2d array");  
        int a[][] = {{10, 20, 30}, {40, 50, 60}, {70, 80, 90}};  
        for(int i=0; i<a.length-1; i++)  
            for(int j=0; j<a[i].length-1; j++)  
                if(icprime(a[i][j]))  
                    System.out.println(a[i][j]);  
    }  
    static boolean icprime(int n)  
    {  
        if(n<=1)  
            return false;  
        for(int i=2; i<n/2; i++)  
            if(n % i == 0)  
                return false;  
        return true;  
    }  
}
```

③ WAPT sum of diagonal elements in 2d array

```
class A2{  
    public static void main(String args){  
        ps.println("sum of diagonal elements in 2d array");  
        int a[][] = {{10, 20, 30}, {40, 50, 60}, {70, 80, 90}};  
        int pg=0, ss=0;
```

```

for (int i=0; i<a.length-1; i++)
{
    for (int j=0; j<a.length-1; j++)
    {
        if (i==j)
            ps = ps + a[i][j];
        if (i+j == a.length-1)
            ss = ss + a[i][j];
    }
    sopln("primary diagonal element sum : " + ps);
    sopln("secondary diagonal element sum : " + ss);
}

```

(4)

```

class A4E
{
    public static void main (String args[])
    {
        int a[][] = {{10, 20, 30}, {40, 50, 60}, {70, 80, 90}};
        int b[][] = new int [a.length][a[0].length];
        for (int i=0; i<a.length-1; i++)
        {
            for (int j=0; j<a.length-1; j++)
            {
                b[i][j] = a[i][j];
            }
        }
        sopln("-----");
        for (int i=0; i<b.length-1; i++)
        {
            for (int j=0; j<b.length-1; j++)
            {
                sopln(b[i][j]);
            }
        }
    }
}
```

```
    }  
    sopIn();
```

```
}
```

Taking a user input for 2D array.

Code Demos

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

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

```
    System.out.println("Enter the no. of rows");
```

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

```
    System.out.println("Enter the no. of cols");
```

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

```
    int a[][] = new int[rows][cols];
```

```
    System.out.println("Enter " + (rows * cols) + " elements");
```

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

```
{
```

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

```
{
```

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

```
}
```

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

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

```
{
```

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

```
{
```

```
        System.out.print(a[i][j]);
```

```
}
```

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

```
}
```

```
}
```