13/7/23

2) write a program to swap 2 no. by using 3rd variable or temeporary variable

```
int a = 20;
int b = 10;

int tamp = a;
    a = b;
    a = Temp;

sop (a);
sop (b);
```

→ int a = 20;
int b = 10;

a = a+b;
30 = 10+20

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

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

sop(a);
sop(b);

To find sum of numbers

```
int no = 1234;
int sum = 0;
while (no != 20)
{
int rem = no % 10;

sum = sum+rem;
```

4 = 0 + 4
7 = 4 + 3
9 = 7 + 2
10 = 9 + 1

```
no = no/10;
```

123 = 1234/10
12 = 123/10
1 = 12/10
0 = 1/10

```
sop (sum);
```

1234! = 0 ✓
123! = 0 ✓
12! = 0 ✓
1! = 0 ✓

10) 1234 (123
1230
4

10) 123 (12
120
3

10) 12 (1
10
2

10) 1 (0
0
1

```
-> package Jspider;
import java.util.scanner;
Public class scanner
                sumeven.
{
    Static int sum (int no)
    {
        int sum=0;
        int rem;
        while (no!=0)
        {
            rem= no%10;
            if (rem%2==0)
            {
                sum= sum+rem;
            }
            no= no/10;
        }
        return sum;
    }
    PSVM(-)
    {
        sop (" Enter number");
        scanner sc= new scanner (system.in);
        int no= sc.nextInt();
        sop (sum(no));
    }
}
```

2)

```
import java.util.scanner;
Public class sumodd
{
    Static int sum (int no)
    {
        int sum=0;
        int rem;
```

```
while (no!=0)
{
  rem = no%10;
  if (num%2!=0)
  {
    sum= sum+rem;
  }
  no= no/10;
}
return sum;
}
psvm (-)
{
  sop ("enter number:");
  Scanner sc= new scanner (System.in);
  int no = sc. nextInt ();
  sop (sum(no));
}
}
```

15/7/23

→ WAP TO print even index array ~~values~~ elements.

```
psvm ( )
{
  int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
  for (int i=0; i< arr.length; i++)
  {
    if (i%2==0).
    {
      sop (arr[i]);
    }
  }
}
}
```

→ WAP To print even array elements

```
for (int i=0; i<arr.length; i++).
{
    if (arr[i]%2==0)
    {
      sop(arr[i]);
    }
}
```

→ WAP To search given values is present in the unsorted

```
int[] arr={9,8,1,3,2,5,4,7,6};
int K=7;
for (int i=0; i<arr.length; i++)
{
    if (arr[i]==K)
    {
      sop("The value'+K+'is present in the index'
                                  of'+i);
    }
}
```

O/P

The value 7 is present in the index of 7.

→ WAP To print array elements in ascending order in un sorted array elements.

```
import java.util.Arrays;
class sample
{
  psvm ( ).
  {
    int[] arr={4,5,1,3,2};
    Arrays.sort(arr);
    for (int i=0; i<arr.length; i++)
    {
      sop (arr[i]);
    }
  }
}
```

(fnt F3)

Back end

```
final class Arrays
{
  static void sort(int[] a)
  {
    Dual pivote Quick.sort(a)
  }
}
```

Assignment

1) I/P: mava1 atte2 rama3 pavan4

O/P: mava9 atte9 rama9 pava9

```
class sample{
 psvm()
 {
  string s1= "mava1 atte2 rama3 pavan4";
  sop (s1.replaceAll("[0-9]","9"));
 }
}
```

2) I/P: mava67 6tte7 r6m67 pGv6n7

```
class sample{
 psvm (-)
 {
  string str= "mava1 atte2 rama3 pavan4";
  str= str.replaceAll("a","6");
  sop (str.replaceAll("[1-4]","7"));
 }
}
```

3) O/P: java atte raja ravana

```
class sample{
 psvm (-)
 {
  string str= "mava atte rama pavana";
  str= str.replaceAll("p","r");
  sop(str.replaceAll("m","j"));
 }
}
```

4) O/P: j@123v@123 @123tte r@123m@123 r@123v@123n@123

```
-> class sample4{
 psvm (-)
 {
  string s1= "java atte rama ravana";
  sop (s1.replaceAll("a","@123"));
 }
}
```

5) O/P: mava atte rama pavana

```
class sample5{
 psvm (-)
{
```

```
string s1 = "ma@va! a#tc$ey r%am*a p?av#an@
        a";
  sop (s1. replaceAll ("[@!#$% * ?]", " "));
 }
}
```

→ print input in Reverse order.

```
Publice class sample{
    static void check()
    {
      String arr[] = {" kedarnath", "badrinath", "amarnath",
                      "brundan", "dwarak", "kash"};

      for (int i=0; i<arr.length; i++)
      {
        string str = arr[i];
        String reverse="";
        for (int j = str. length()-1; j>=0; j--)
        {
          reverse= reverse + str. charAt(j);
        }
        sop (reverse+" ");
      }
    }
  }
  psvm ( - )
  {
    Check();
  }
}
```

→ print only Palindrome from input.

```
Class sample{
 Static void check (string[] s1)
 {
   for (int i=0; i<s1. length; i++)
   {
    string str= s1[i];
```

```
String reverse1 = "";
for (int j = str.length() - 1; j > 0; j--)
{
  reverse1 = reverse1 + str.charAt(j);
}
if (str.equals (reverse1))
{
  sop (str + " is palindrome");
}
else
{
  sop (str + " is not palindrome");
}
}
psvm (-)
{
  String[] str = {"pinya", "Kanak", "Nayan", "malayu",
    "Toyot", "saras", "manya", "amazon"};
  Check (str);
}
}
```

→ pramdu. ReturnType of factorial using DynamicInput.

```
import java.util.scanner;
Public class {
static void num (int n) {
  int fact = 1;
  for (int i = n; i >= 1; i--)
  {
    fact = fact * i;
  }
  sop (fact);
}
static int no ()
{
    int fact1 = 1;
    for (int i = 6; i >= 1; i--)
```

```
{
    fact1 = fact1 * i;
}
return(fact1);
}
psvm (-) {
    scanner sc= new scanner(system.in);
    sop ("enter num");
    int a = sc.nextInt();
    num(a);
    int z= no();
    sop(z);
}
}
```

1) sum of even num

I/P : 123456 O/P: 12

```
-> import java.util.scanner;
Public class sample1 {
  static int sum (int no){
    int sum=0;
    int rem=0;
    while (no!=0)
    {
      rem =no%10;
      if (rem%2==0)
      {
        sum = sum+rem;
      }
      no = no/10;
    }
    return sum;
  }
  Psvm (-) {
    sop("enter value");
    scanner sc= new scanner(system.in);
    int no = sc.nextInt();
    sop (sum(no));
}
}
```

2) sum of odd num

I/P : 123456 O/P: 9

```
import java.util.scanner;
Public class s2 {
  static int sum (int no){
    int sum=0;
    int rem=0;
    while (no!=0)
    {
      rem= no%10;
      if(rem%2!=0)
      {
        sum = sum+rem;
      }
      no=no/10;
    }
    return sum;
  }
  psvm (-) {
    sop("enter value");
    scanner sc= new scanner(system.in);
    int no = sc.nextInt();
    sop(sum(no));
}
}
```

3) number divided by 3 & it's sum

i/p: 13956 o/p: 18

```
import java.util.scanner;
public class S3 {
static int sum (int no){
    int sum=0;
    int rem =0;
    while (no!=0)
    {
      rem=no%10;
      if(rem%3==0)
      {
        sum= sum+rem;
      }
      no=no/10;
    }
    return sum;
  }
  psvm(-){
  sop("enter sum");
Scanner sc2 new scanner(system.in);
  int no = sc. nextInt();
  sop (sum (no));
  }
}
```

4) sum of num div by 4.

i/p: 82345 o/p: 12.

```
-> import java.util.scanner;
    class sample{
  static int sum(int no){
    int sum =0;
    int rem=0;
    while (no!=0){
      rem = no%10;
      if (rem%4=0)
      {
        sum= sum +rem;
      }
      no= no/10;
    return sum;
  }
  psvm(-)
  {
  sop("enter number");
  Scanner sc = new Scanner
      (system.in);
  int no = sc.nextInt();
  sop (sum(no));
  }
}
```

5) count even num.

i/p: 123456. o/p: 3

```
-> import java.util.scanner;
  Public class sample{
  static int sum(int no){
    int count=0;
    int rem=0;
    while (no!=0){
      rem= no%10;
      if(rem%2==0){
        count++;
      }
      no = no/10;
    }
    return count;
  }
  psvm (-){
  sop ("enter value");
Scanner sc2 new scanner (system.
    in);
  int no= sc.nextInt();
  sop (sum (no));
  }
}
```

6) Count even number

i/p: 123456 o/p: 3

```
import java.util.scanner;
```

```
public class S6{
static int sum (int no){
 int count =0;
 int rem =0;
 while (no!=0){
 rem= no%10;
 if (rem%2!=0)
 {
  count++;
 }
  no= no/10;
 }
 return count;
 }
 psvm(-){
sop("Enter number");
scanner sc= new scanner(system.in);
int no = sc.nextInt();
 sop (sum (no));
 }
}
```

7) multiplication of even no.

```
import java.util.Scanner;
public class S7 {
 static int mul( int no){
     int mul =1;
     int rem=0;
    while (no!=0){
     rem= no%10;
     if (rem%2==0){
      mul= mul*rem; }
       no=no/10;
    }
    return mul;
 }
 psvm()
 {
  sop("enter no");
  scanner sc= new scanner(system.in);
  int no= sc.nextInt();
  sop(mul(no));
 }
}
```

8) multiplication of numbers which is divisible by 3.

```
import java.util.Scanner;
public class S8 {
 static int sum(int no)
 {
   int mul= 1;
   int rem=0;
   while(no!=0)
   {
     rem= no%10;
     if(rem%3==0){
       mul=mul*rem;
     }
     no=no/10;
   }
   return mul;
 }
 psvm(-)
 {
  sop("enter no");
  scanner sc= new scanner
     (system.in);
  int no = sc.nextInt();
   sop(sum(no));
   }
 }
```

8) sum of 1st & Last number

i/p: 1234 o/p: 5

```
import java.util.Scanner;
Public class S8 {
 static void int sum(int no)
 {
   int sum=0;
   int rem=0;
```

```
while(no!=0)
 {
   rem = no%10;
   sum = sum+rem;
   no = no/1000;
 }
 return sum;
}
psvm(-) {
sop("enter no");
Scanner sc = new Scanner(system.in);
int no = sc.nextInt();
sop(sum(no));
}
}
```

9) multiplication of 1st and last number

I/P: 9735 O/P: 45

```
import java.util.scanner;
Public class sq {
 static int sum(int no)
  {
    int mul = 1;
    int rem = 0;
    while(no!=0){
      rem = no%10;
      mul = mul* rem;
      no = no/1000;
    }
    return mul;
  psvm(-) {
  sop("enter number")
  scanner sc = new scanner(system.in);
  int no = sc.nextInt();
  sop(sum(no));
  }
 }
```

10) mul no. div. by 4

i/P: 82345 O/P: 32

```
import java.util.scanner
Public class 10 {
 static int sum(int no)
 {
   int mul = 1;
   int rem = 0;
   while(no!=0){
   rem = no%10;
   if(rem%4==0){
   mul = mul* rem;
   }
   no = no/10;
 }
 return mul;
}
psvm(-) {
 sop("enter no");
Scanner sc = new scanner(
  system.in);
int no = sc.nextInt();
sop(sum(no));
}
}
```

Assignment on split method

1) i/p: hello java how are you.

o/p: hello java you are how.

```
psvm (-){
String str = "hello java how are you";
String[] s1 = str.split(" ");
for (int i=0; i< s1.length/2; i++)
{
    sop (s1[i]+" ");
}
for (int i=s1.length-1; i>2; i--)
{
    sop (s1[i]+" ");
}
```

2) i/p: Java is programming Lang.

o/p: Lang programming is Java

```
Publicclass C1{
String str = "Java is programming Lang";
String[] s1 = str.split(" ");
for (int i=s1.length-1; i>=0; i--)
{
    sop (s1[i]+" ");
}
}
```

3) i/p: hi hello java how are you

o/p: hi hello java you are how.

```
Publicclass C2{
psvm(-){
String str = "hi hello java how are you";
String[] s1 = str.split(" ");
for(int i=0; i<s1.length/2; i++)
{
    sop(s1[i]+" ");
}
for(int j=s1.length-1; j>=s1.length/2; j--)
{
    sop (s1[j]+" ");
}
}}}
```

4) o/p: java hello hi how are you

```
Public class C4{
psvm(-){
String str = "hi hello java how are you";
String[] s1 = str.split(" ");
for (int i=s1.length/2-1; i>=0; i--)
{
    sop(s1[i]+" ");
}
sop(" ");
for(j=s1.length/2; i<s1.length; i++)
{
    sop (s1[j]+" ");
}
}}}
```

5) o/p: hi java are (even index)

```
Public class C5{
psvm(-){
String str = "hi hello java how are you";
String[] s1 = str.split(" ");
for(int i=0; i<=s1.length-1; i++)
{
    if (i%2==0)
    {
        sop(s1[i]+" ");
}}}}
```

6) O/P hello how you (odd index)

```
Public class C6 {
 PSVM(-) {
String str = "hi hello java how
                are you";
String[] s1 = str.split(" ");
for (int i=0; i<=s1.length-1;i++)
  {
    if(i%2!=0)
    {
      sop (s1[i]+" ");
    }
  }
 }
}
```

7) O/P :- how are you java hello hi

```
Public class C7 {
  PSVM (-) {
String str = "hi hello java howare you";
String[] s1 = str.split(" ");
for (int i=s1.length/2; i<s1.length;
        i++)
{
   sop (s1[i]+" ");
}
for(int i=s1.length/2-1; i>=0; i--)
  {
    sop (s1[i]+ " ")
  }
}}}
```

8) Reverse even index

```
Public class C8 {
  PSVM(-) {
  String str = "hi hello java
        howare you";
String[] s1 = str.split(" ");
for(int i=0; i<=s1.length-1;
        i++)
{ if (i%2==0)
  {
    String str2 = s1[i];
    String reverse = "";
for(int j = str2.length()-1;
        j>=0; j--)
  {
    reverse = reverse + str2.
        charAt(j);
  }
  sop (reverse+" ");
}}}
```

9) O/P: olleh woh uoy (Revers odd Index)

```
Public class C9 {
 PSVM(-) {
String str = "hi hello java how
        are you";
String[] s1 = str.split(" ");
for (int i=0; i<s1.length; i++)
{
  if (i%2!=0)
  {
    String str2 = s1[i];
    String reverse = "";
for(int j=str2.length()-1;
        j>=0; j--)
  {
    reverse = reverse+str2.
        charAt(j);
  }
  sop (reverse);
}}}
```

→ write a pgm to find given no is perfect or not.

```
psvm (-){
int no = 28;
int sum = 0;
for (int i=1; i<= no/2 ; i++)
}
    if (no% i == 0)
    }
      sum = sum+i;
    }
}
if (no==sum)
}
    sop ("perfect no");
    else
    sop ("NOT a perfect no");
}
}
```

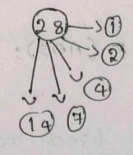
i<14; 2<14; 3<14;

28%1==0. 28%2==0 28%4==0

1 = 0 + 1
3 = 1 + 2
7 = 3 + 4
14 = 7 + 7
28 = 14 + 14

28 == 28

NOTE → 14+7+4+2+1 = 28



its perfect

→ write a pgm to write factors of given no.

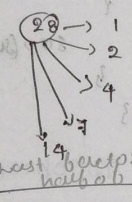
```
psvm ()
{
int no=28;
for (int i=1; i<= no/2 ; i++)
{
  if (no% i ==0)
  }
    sop(i);
  }
}
```

i<= no/2
1<= 14
2<= 14
3<= 14

28%1==0
28%2==0
28%3==0 ×

o/p = 1
o/p = 2

NOTE



last factor will be half of given no.

→ write a pgm to write print power of. n(n^p)

```
psvm ()
{
Scanner sc = new scanner (system.in);
sop ("enter the no");
int base = sc.nextInt();//2
int exp = sc.nextInt();//3
int Ans = 1;
while (exp != 0)
```

n = base
p = exponential

```
{
    Ans = Ans * base;
    Exp - - ;
}
sop (Ans);
}
```

O/P = 2^3
= $2 * 2 * 3$
= 8

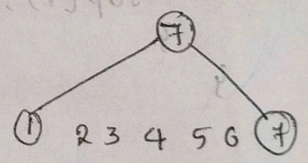
→ WAP To find range of perfect number 100.

```
psvm.
for (int no=1; no<=100; n++){
  int sum=0;
  for (int i=1; i<=no/2; i++)
  {
      if (no% i==0)
      {
          sum = sum + i;
      }
  }
  if (no == sum)
  {
      sop(sum);
  }
}
```

100 → 1, 2
1.

→ WAP To check given number is prime or not.

```
psvm (—)
{
  int no = 4;
  boolean flag = true; //false
  for (int i=2; i<no; i++)
  {
      if (no% i==0)
      {
          flag = false;
          break;
      }
  }
  if (flag == true)
  {
```



```
        sop("prime number");
        }
        else
        {
            sop("not a prime number");
        }
    }
```

-> WAP To check 7 is prime or not.

```
psvm (-)
{
  int no=7;
  boolean flag = true;
  for (int i=2; i<no; i++)
  {
      if (no% i==0)
      {
        flag=false;
        break;
      }
  }
  if (flag==true)
  {
    sop("prime number");
  }
  else
  {
    sop("NOT a prime number");
  }
}
```

-> WAP To find range of prime number from 3 to 100

```
psvm
{
  for(int j=3 j<=100; j++)
  {
      int no=j;
      boolean flag=true;
      for (int i=2; i<no; i++)
        { if (no%i==0)
```

```
            flag = false;
            break;
        }
    }
    if (flag == true)
    {
        sop (no+" is a prime no");
    }
    else
    {
        sop (no+" is not prime no");
    }
}
```

-> WAP To find range of prime no. from 3 To 100 using parameters.

```
class prime {
    void num (int a, int b)
    {
        for (int j=a; j<=b; j++)
        {
            int no = j;
            boolean flag = true;
            for ( int i = 2; i< no; i++)
            {
                if ( no % i == 0)
                {
                    flag = false;
                    break;
                }
            }
            if (flag == true)
            {
                sop (no + " is a prime no");
            }
            else
            {
                sop ( no + " is not a prime no");
            }
        }
```

→ wap to print next prime no of given number

```
psvm (→)
{
    int a=3;
    for(int j=a+1; j<=1000; j++)
    {
        int no=j;
        boolean flag=true;
        for (int i=4; i<no; i++)
        {
            if (no%i==0)
            {
                flag=false;
                break;
            }
        }
        if (flag==true)
        {
            sop (no+"is prime");
        }
    }
}
```

→ write a pgm to print sum of given no till getting single digit. [AACAP]

```
public class Q12 {
    static int numsum (int no)
    {
        int sum=0;
        while (no!=0)
        {
            int rem= no%10;
            sum= sum+rem;
            no= no/10;
        }
        return sum;
    }
    psvm (→) {
        int no=64;
```

```
while (no>9)
  {
    no= numsum(no);
  }
  sop(no);
}
}
```

19/7/23

* To convert string to Integer we use Integer.parseint.

→ WAP to find sum of given number, check wether its prime or not using parameter.

```
Public class{
  static void numsum(int no)
  {
    int sum=0;
    while(no!=0)
    {
      int rem = no%10;
      sum= sum+rem;
      no= no/10;
    }
    sop("sum is"=>"+" "+sum);
    Prime(sum);
  }

  static void prime(int no)
  {
    boolean flag = true;
    for (int i=2; i<no; i++)
    {
      if (no%i==0)
      {
        flag= false;
        break;
      }
    }
    if (flag==true)
    {
      sop(" prime number");
    }
```

```
else
{
  sop ("not a prime number");
 }
}
psvm (-) {
   int no = 3453;
   numsum (no);
 }
}
```

→ WAP To find the sum of factorial of each sum

i/p = 145 o/p = 5! + 4! + 1! = 145

```
Public class c2 {
   static void numsum (int no)
   {
     int copy=no;
     int sum=0;
     while (no!=0)
     {
      int rem = no % 10;
      sum = sum + factorial (rem);
       no = no/10;
     }
     if (copy == sum)
     {
       sop (sum+" "+ is a strong number");
     }
    }
    static int factorial (int no)
    {
      int fact=1;
      for (int i=no; i>=1; i--)
      {
          fact = fact * i;
      }
       return fact;
    }
```

```
psvm (-)
{
 int no = 145;
 numsum (no);
}
}
```

→ WAP To find The sum of given num with parameters return Type I/P : 64

```
Public class C4 {
 static int numsum (int no)
 {
   int sum=0;
   while (no!=0)
   {
   int rem = no%10;
   sum = sum+rem;
     no = no/10;
   }
   return sum;
 }
 psvm (-){
   int no=64;
   int y= numsum(no);
   sop(y);
 }
}
```

→ WAP To find the sum of given number Till getting single digit (happy number) if we get [147]

```
Public class C5 {
static int numsum (int no)
{
   int sum=0;
   while (no!=0)
   {
     int rem= no%10;
     sum= sum+rem*rem;
```

```
    no = no/10;
  }
  return sum;
}
psvm (-) {
int no=49;
while (no>9)
{
   no = numsum(no);
}
if(no==1 || no==7)
{
  sop(" Happy number");
}
else
{
   sop ("not a happy number");
}
}
```

19/7/23

```
-> public class C6{
   psvm (-) {
     int no=65;
     char ch =(char) no;
     sop (ch);
     int n=97;
     char c1= (char) (no+4);
     sop (c1);
     char ch1 = 'A';
     int num =(int) ch1;
      sop (num);
     char ch2= 'a';
     int num2= (int) (ch2+4);
     sop (num2);
```

I/P = 65
O/P = A

I/P = 97
O/P = E

I/P = 'A'
O/P = 65

I/P = a
O/P = 101

→ I/P string="hello", int no = 4 O/P Lipps using scanner i/p.

```
import java.util.Scanner;
public class Ascci3{
static void characterprinting (string a, int b)
 {
    for (int i=0; i<a.length(); i++)
    {
      char ch = a.charAt(i);
      sop ((char) (ch+4));
    }
 }
 psvm (-)
 {
    Scanner sc = new scanner (system.in);
    sop ("enter string");
    string s1 = sc.nextline();
    sop ("enter int");
    int x = sc.nextInt();
    characterprinting (s1, x);
  }
 }
```

→ WAP TO convert string TO Integer

```
-> public class Ascci6
  {
   static int add (string a)
   {
      int no = Integer.parseInt(a);
      return (12+10);
   }
   public static void main (string[] args)
   {
      string = "12";
      int x = add(str);
      sop(x);
   }
  }
```

21/7/23

→ convert to string array

I/P : hi java hello java how are you java

O/P : Java
(3 Times)

```
Public class stringtoarray
{
    Public static void main(String[] args)
    {
        String str= "hi java hello java how are you java";
        string[] s1= str. split(" ");
        String K= "java";
        int count =0;
        for (int i=0; i<s1.length; i++)
        {
            if (s1[i].equals(K))
            {
                Count++;
            }
            }
            sop(count);
        }
    }
```

Above question using dynamic input.

```
Package sample;
import java.util.Scanner;
Public class stoarray{
    static int area (string s1) { int count=0;
  String[] s2 = s1. split (" ");
  for (int i=0; i<s2.length ;i++)
  {
    if ( s2[i]. equals ("java"))
      {
        count++;
      }
  }
  return count;  }
```

```
public static void main (String[] args)
{
  scanner sc= new scanner(System.in);
  sop (" enter a string");
  String s1 = sc.nextInt line ();
   sop (area (s1));
 }
}
```

→ creating array with dynamic input
That is [initialisation & declaration].

```
package sample;
import java.util.scanner;
Public class s4 {
   Public static void main (String[] args)
   {
      Scanner sc= new scanner(System.in);
      sop (" Enter The size of the array");
      int size = sc.nextInt();
      int[] arr= new int[size];
       sop (" Enter The Elements");
       for( int i=0; i<arr.length; i++)
        {
          arr[i] = sc.nextInt();
        }
        sop (" The array values are");
        for (int i=0; i<arr.length; i++)
         {
            sop (arr[i]+" ");
         }
       }
     }
```

→ convert string To character without charAt method.

```
package sample;
Public class s7 {
```

```
psvm (- )
{
  String str = "java";
  char[] ch = str.toCharArray();
  for (int i = ch.length-1; i>=0; i--)
  {
    sop (ch[i]);
  }
}
}
```

22/7/23

1) WAP To reverse a given string.

```
class sample
  psvm (-)
{
  String str = "sandhya";
  String[] s1 = s1.split(" ");
  String rev = "";
  for(int i = str.length-1; i>=0; i--)
  {
    rev = rev + str.charAt[i];
  }
  sop (rev);
}
```

2) WAP To reverse a string without using inbuilt function

```
Class sample
{ psvm(-)
  String str = "java";
  String rev = "";
  char[] arr = str.toCharArray();
  for(int i = str.length-1; i>=0; i--)
  {
    rev = rev + arr[i];
  }
  sop(rev);
}
```

- To convert from string to Array

```
String s1 = "Java"
char[] arr = s1.toCharArray();
```

arr → | J | a | v | a |
0 1 2 3

3) WAP To reverse a number count How many time character a is repeated in the string.

```
class sample
{
  psvm ()
  {
    int count = 0;
    String s1 = "Java";
    String rev
    char[] arr = s1.toCharArray();
    for (int i = s1.leng
    for (int i=0; i<=s1.length; i++)
    {
        if (arr[i] == 'a')
        {
          count++;
        }
    }
```

4) WAP To find factorial of number.

5 * 3 * 2 1.

```
class sample
{
  psvm ()
  {
    int fact = 1;
    static int find fact (int no).
    {
      fact for(int i=no; no >= 1 ;  i--)
      {
        fact = fact * i;
      }
    return fact;
```

```
psvm (-)
{
    int no = 5;
    sop(binBcd(no));
}
}
```

P) WAP To reverse a given string without using for loop & inbuilt fun.

```
String s1 = "sandhya";
String rev = " ";
char[] arr = s1.toCharArray();
int i = arr.length-1;
while (i >= 0).
{
    rev = rev + arr[i]; (OR) rev = rev + arr[i--];
    i--;
}
sop(rev);
}
```

→ WAP To print the sentence in the reverse order one by one

```
String[] arr = {"Hi", "OK", "COOL"}
for (int i=0; i<arr.length; i++)
{
    sop(arr[i]);
}
String s1 = new String(arr);
    = " Hi OK COOL";
    sop(s1)
```

Convert An Array to String

```
static void reverseSentence ( )
{
 string s1= "Java mava sobaa firaa";
 char[] abb = s1.toCharArray();
 string[] arr = new string[ countingchar (s1, ' ')+1];
 string rev = " ";
 for (int i=0; i< abb.length; i++)
 {
   if (abb[i] != ' ')
   {
     rev = rev+abb[i];
   }
   else
   {
     for (int j=arr.length-1; j>=0; j--)
     {
       arr[j]= rev;
       rev = " ";
     }
   }
 }
 arr[0]= rev;
 String s3 = new string(arr); // converting array
                                  To string
 sop (s3);
}

static int countingchar (string s1, char search)
{
   int charcount =0;
   char[] arr = s1.toCharArray();
   for (int i=0; i< arr.length; i++)
   {
     if (search == arr[i])
     {
       charcount++;
     }
   }
   return charcount;
}
```

1) I/P. string s1= "cool. java mava Rodaa"

search a

O/P = 2

2) I/P string s1= " mom Ram sam dam col"

Search 'o'

o/P = mom, col

3) I/P string s1= " cool Rama sita wow mom"

Search which word has 4 letter and contain 'o'

o/P : 2

4) I/P string s1= " Java Rama sita wake morn"

<u>O/P</u> :- <u>'atis'</u> which contains 4 letter & it has only one 'a'.

<u>24/7/23</u>

```
class sample
{
  static void print (int[] arr)
  {
    for (int i=0; i<arr.length; i++)
    {
      if (arr[i] % 2 ==0){
          sop (arr[i]);
      }
      psvm ( )
      {
        int [] abb = {1,2,3,4,5};
        printArray (abb);
      }
    }
```

Area of circle method with parameter 24/7/23

```
class sample
{
  static void printarray (int[] arr)
  {
    for(int i=0; i<arr.length; i++)
    {
      area1 (arr [i])
    }

static void area1 (int r)
{
  final double Pi = 3.142;
  double res = pi*r*r;
    sop (res);
}

psvm(-)
{
 int[] abb = {1,2,3,4,5}
  printArray (abb); // actual arguments
}
```

Area of circle method with return type

```
class sample
{
  static void printArray (int[] arr)
  {
    for (int i=0; i<arr.length; i++)
    {                          if (arr[i] % 2 == 0) {
doublex=area1(arr[i]);
      sop (x);
    }
  }

static double area1 (int r)
  {
      final double pi= 3.142;
      double res = pi*r*r;
  }
   return (res);
  }
```

```
Public static void main (string[] args)
{
  int [] abb = {1,2,3,4,5};
  PrintArray (abb);
}
}
```

Siddhartha
25/7/23

1) WAP TO print an array in forward direction from mid

```
-> Class sample
{
  static void printArray ()
  {
    int [] arr = {1,2,3,4,5,6,7,20,30}
    for (int i = arr.length/2 ; i < 9 ;)
      {
        sop (arr[i++]);
      }
  }
```

2) WAP TO print an array in backward direction from mid

```
-> class sample
{
  static void PrintArray ()
  {
    int [] arr = {1,2,3,4,5,6,7,20,30}
    for (int i = arr.length/2 ; i >=0 ;)
      {
        sop (arr[i--]);
      }
  }
}
```

3) WAP TO print an array in forward & backward direction

```
Class sample
{
  static void PrintArray ()
  {
    int arr = {10,20,30,40,50,60,70,80,90}
    int mid = arr.length/2 ;
    for ( int i=0; i<arr.length; i++)
```

```
}
  if (mid>=0)
  {
    sop (arr[ mid--] );
  }
  else
  {
    sop (arr[mid+j]);
  }
}
```

I/P: 10 20 30 40 50 60 70 80 90

O/P: 50 60 70 80 90 10 20 30 40 50

I/P: 10

→	→	R R
→	←	R L
←	←	L L
←	→	L R

WAP TO sort an array in ascending order without using built in method

[Bubble sort]

j → 0	1	2	3	4	
5	4	1	3	2	
4	5	1	3	2	
4	1	5	3	2	
4	1	3	5	2	
4	1	3	2	5	→ ① Iteration
1	4	3	2	5	
1	3	4	2	5	
1	3	2	4	5	→ ② Iteration
1	3	2	4	5	
1	2	3	4	5	→ ③ Iteration
1	2	3	4	5	→ ④

⇒ Pgm

```
Public class Array1 {
Public static void main (String[] arg[])
{
  int[] arr = { 5, 4, 1, 3, 2 };
  bubble sort (arr);
}
static void bubble sort (int[] arr)
{
  for(int i=0; i<arr.length; i++)
                    //no.of iteration
  {
    for(int j=0; j<arr.length-1;
                  j++) //traversal
    {
      if (arr[j] > arr[j+1])
      {
        int temp = arr[j];
        arr[j] = arr[j+1];
        arr[j+1] = temp;
      }
    }
  }
  for (int i=0; i<arr.length;
                  i++)
  {
    sop (arr[i] + " ");
  }
}
```

```
Class Sample
{
 static int[] findMidPl (int[] acc)
 {
  int mid = acc.length/2;
   int [] akk = new int[mid+1]
   for (int i=0; i<acc.length; i++)
   {
    if (mid>=0)
    {
     akk[i]= acc [mid--]
    } }
     return akk;
   }
   PSVM (-)
   {
    int[] arr= {10, 20, 30, 40, 50, 60, 70, 80, 90};
   int[] up1 = findmidPl (arr);
    for (int i=0; i<up1.length; i++)
     {
      sop (up1 [i]);
     }
   int [] up2 = findmidPl (up1)
     for (int i=0; i<up2.length; i++)
     {
      sop (up2[i]);
     }
   int[] up3 = findmidpl (up2)
     for (int i=0; i<up3.length; i++)
     {
      sop (up3 [i]);
     }
   }
```

26/7/23

Int

```
Class sample
{
 static int[] dynamicInput Array ( )
 {Scanner sc = new scanner (system.in);
  sop ("enter the size of an Array");
  int size = sc.nextInt();
  int[] arr = new int [size];
  for (int i=0; i<arr.length; i++)
  {
   sop ("enter the " + i + " index value");
     arr[i]= sc.nextInt();
  }
  return arr;
 }
 static void printArray (int[] acc)
 {
  for (int i=0; i<acc.length; i++)
  {
    sop (acc[i]);
  }
 psvm (-)
 {
  int[] akk = dynamicInputArray();
   printArray (akk);
 }
}
```

String

```
Class sample
{
 static string[] dynamicInput Array()
 {
  scanner sc = new scanner (system.in);
  sop(" enter the size of an array");
  int size = sc.nextInt();
  string[] arr = new string [size];
  for (int i=0; i<arr.length; i++)
  {
    sop (" enter the" + i + " index value");
```

```
    arr[i] = sc.nextLine();
  }
  return arr;
}
static void printArray (String[] acc)
{
  for(int i=0; i<acc.length; i++)
    {
      sop (acc[i]);
    }
}
PSVM (-)
{
  String[] akk = dynamicInputArray ();
  printArray (akk);
}
}
```

→ char

```
class sample
{
  static char[] dynamicInputArray ()
  {
    Scanner sc = new Scanner (System.in)
    sop(" enter the size of an Array");
      int size = sc.nextInt();
    Char[] arr = new Char [size];
    for (int i=0; i<arr.length; i++)
      {
        sop (" enter the "+i+" index value");
          arr[i] = sc.next().charAt(0);
      }
    return arr;
  }
  static void printArray (Char[] acc)
    {
      for (int i=0; i<acc.length; i++)
      {
        sop (acc[i]);
      }
    }
  PSVM (-)
```

```
{
    char [] akk = dynamicInputArray ();
    printArray (akk);
  }
}
```

→ double

→ class sample

```
{
  static double[] dynamicInputArray ()
  {
    scanner sc = new scanner (System.in)
    sop ("enter the size of an array");
    int size = sc.nextInt();
    double [] arr = new double [size];
    for(int i=0; i<arr.length; i++)
    {
      sop (" enter the " + i + " index value");
      arr [i] = sc.nextdouble ();
    }
    return arr;
  }
  static void printArray (double[] acc)
  {
    for (int i=0; i<acc.length; i++)
    {                    if(acc [i]%2 == 0), for even
      sop (acc[i]);
    }
  }
  psvm (-)
  {
    string[]
    double[] akk = dynamicInputArray ();
      print Array (akk);
  }
}
```

Amstrong Number:

```
int no = 153;
int no1 = no; //count
int copy = no; //comparision.
int count = 0; //3
int sum = 0;
```

153 → count 3

↳ $3^3 = 27$

→ $5^3 = 125$

→ $1^3 = 1$

153

```
while (no1 != 0)
{
    no1 = no1/10;
    count++;
}
while (no1 != 0)
{
    int rem = no%10;
    sum += Math.pow(rem, count);
    no = no/10;
}
if (copy == sum)
{
SOPrint("~~Arm~~ Armstrong Number");
}
else
{
    SOP ("Not a Armstrong number");
}
```

->WAP To find The ~~Fibona~~ Fibonaccie

[print till Range 10]

```
-> class sample
{
    public class Array1 {
        PSVM(-) {
            int fib1 = 0;
            int fib2 = 1;
            int fib3 = 0;
            Sop (fib1 + " " + fib2 + "\n");
        for(int i=1; i<10; i++)
        {
            fib3 = fib1 + fib2;
            Sop (fib3 + " ");
            fib1 = fib2;
            fib2 = fib3;
        }
    }
}
}
```

O/P:

0 1
1 2 3 5 8
... 55

→ PSVM (-) {

[print only 10th digit Fibonaccie]

```
int fib1=0;
int fib2=1;
int fib3=0;
Sop (fib1 + " " + fib2 + "\n");
for (int i=1; i<10; i++)
{
    fib3 = fib1 + fib2;
    fib1 = fib2;
    fib2 = fib3;
}
Sop (fib3 + " ");
}
```

Output : 0 1
55

27/7/23

Convert from number to Array:-

```
PSVM (-)
{
 int no = 1238;
 int pos = count()-1;
 int[] arr = new int [count()];
  while (no!=0){
   int rem = no%10;
  arr [pos--] = rem; // fetch & store it in
                        an array
   no = no%10;
  }
}
 static int count ( )
  {
    int no = 1238;
    int count = 0;
    while (no!=0)
    {
      no = no/10;
      count ++;
    }
    return count;
  }
```

String to no.

```
String s1 = "" + 1
s1 = s1 + 2;
s1 = s1 + 3;
s1 = "123" + 4;
s1 = "1234";
int n = Integer.parseInt(s1);
sop(n);
```

parseInt will take string i/p (only nos) & convert to int format

Character to string

```
char[] arr = {'J', 'a', 'v', 'a'}
String s1 = "";
s1 = s1 + arr[0];
   "" + 'J'
sop(s1);
```

String To Array

```
String s1 = "Java";
Char[] arr = s1.toCharArray()
```

Array To number

```
int[] arr = {1, 2, 3, 4};
String s1 = "";
s1 = s1 + arr[0]
convert to no
```

Character to string

```
Char[] arr = {'J', 'a', 'v', 'a'}
String s1 = "";
s1 = s1 + arr[0];
   = "" + J
sop(s1);
```

27/7/23

Leap Year which is divisible by 4, not divisible by 100 & divis-ble by 400.

Code optimization

```
static boolean Leapyear (no)
{
    boolean divby4 = (no%4==0); //~~false~~ true
    boolean divby100 = (no%100==0); //false
    boolean divby400 = (no%400==0) //~~false~~ true

    return (divby4 && !divby100) || divby400;
}
PSVM ()
{
    int no = 2024;
    if (leapyear(no))
    {
        sop (no+" is a leapyear");
    }
    else
    {
        sop (no+" is not a Leapyear");
    }
}
```

OR

```
-> static boolean Leapyear (int no)
{
    boolean divby4 = false;
    boolean divby100 = false;
    boolean divby400 = false;

    if (no%4 == 0)
    {
        divby4 = true;
    }
    else if (no%400 == 0)
    {
        divby 100 = true;
    }
```

```
else if (no%400==0)
{
    divby400 = true;
}
else
{
    sop ("Invalid number");
}

if (((divby4 == true) && (divby100 != false)) ||
            (divby400 = true))
{
    return true;
}
else
{
    return false;
}

psvm (-) {
    int no = 2024;
    boolean result = leapyear (no);
    if ( result == true)
    {
        sop ("leap year");
    }
    else
    {
        sop ("not a leap year");
    }
}
```

→ Print the leap year from 2000 to 2100.

→
```
static boolean Leapyear(no)
{
    boolean divby4 = (no%4 == 0)
    boolean divby100 = (no%100 == 0)
    boolean divby400 = (no%400 == 0)

    return (divby4 && !divby100) || divby400;
}
psvm (-)
{
```

```
for(int no= 2001;   no<=2101;  no++)
{
  if (leapyear(no))
  {
    sop (no+" is a leap year");
  }
}
}
```

```
for(int no= 2001;   no<=2101 ; no++)
{
  if (Leapyear (no))
  {
    sop (no+" is a leap year");
  }
}
}
```

28/7/23

WAP To Rotate an array Towards Right One Time.

```
static void rotateRight (int[] arr. int times)
{
  for (int j=1 ; j<=times; j++)
  {
    int temp= arr [arr.length-1];
    for (int i= arr.length-2; i>=0; i--)
      {
        arr[i+1] = arr [i]
        1        0
      }
    }
    arr[0] = temp;
    } for(int i=0; i<arr.length; i++) { sop (arr[i]);
    }
class main class
{
PSVM ( )
{
int[] arr = {10, 20, 30, 40, 50};
    rotateRight (arr, 2);
}
```

-> WAP To print 1 to 10 from using while loop

```
int i=1
while (i<=10).
{
  sop (i);
  i++;
}
```

10 to 1

→ int i = 10

while (i >= 1)

{

sop (i--);

}

10
9
8
⋮

Switch case:

→ Switch case is used for pattern matching based on the input it will match the particular case way and gets executed.

Syntax:

```
switch (char seq)
{
 case charseq : statement;
                break;

 case charseq : statement;
                break;

 case charseq : statement;
                break;

 default : statement;
           break;
}
```

→ Using switch case printing 1 to 10, 10 to 1

→ Public class sample1 {

PSVM (-) {

int input = 1;

switch (input) {

case 1 : OneToTen ();
break;

case 2 : Ten to One ();
break;

}

}

```
static void onToten()
{
    int i = 1;
    while (i<=10)
    {
        sop(i++);
    }
}
static void Ten toone()
{
    int j=10;
    while (j>=1)
    {
        sop(j--);
    }
}
```

using char

```
psvm (-){
 char input = 'A';
 switch (input){
   case 'A' : sop ("FCA");
              break;
   case 'B' : sop ("FC");
              break;
   case 'C' : sop ("SC");
              break;
   default : sop ("Invalid");
              break;
   }
 }
```

-> int using scanner:

```
Package pgm;
import java.util.scanner;
Public class sample1{
```

```
static scanner sc= new scanner (system.in);
psvm (-){
    sop ("enter your choice");
    sop (" 1,  1 to 10");
    sop (" 2,  10 to 1");
    int input = sc.nextInt();
    switch (input) {
        case 1 : OneToTen(); //input 1 ->   O/P 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
                 break;
        case 2 : TenToOne(); //input 2 ->   O/P 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
                 break;
        default : sop ("Invalid input");
                 break;
    }
  }
}
```

-> Addition, sub, mul

```
class sample{
 psvm (-)
 {
    int input=1;
    switch (input){
      case 1 : add ();
               break;
      case 2 : sub();
               break;
      case 3 : mul();
               break;
      default : sop (" Invalid num");
               break;
    }
 }
 static void add()
 {  sop ("enter 1st no");
    int a = sc.nextInt();
```

```
sop ("enter 2nd no");
int b = sc.nextInt();
sop (a+b);
}
}
```

→ Develop a software for calculator To perform all the Arithmetic operation

```
calculator package;
import java.util.scanner;
public class calculator
{
  static scanner sc = new scanner (system.in);
  static void add (int[] arr)
  {
    int sum = 0;
    for (int i=0; i<arr.length; i++)
    {
      sum = sum+arr[i];
    }
    sop(sum);
  }
  static void sub (int[] arr)
  {
    int sub=0;
    for (int i=0; i<arr.length; i++)
    {
      sub = sub- arr[i];
    }
    sop(sub);
  }
  static void mul (int[] arr)
  {
    int mul= 1;
    for(int i=0; i<arr.length; i++)
    {
      mul = mul* arr[i];
    }
    sop (mul);
  }
```

```
static void div (int[] arr)
 {
    int div= 1;
    for (int i=0; i<arr.length; i++)
    {
        div= div / arr[i];
    }
     sop (div);
 }

psvm ( - ).
 {
    sop("enter How many digits");
    int[] arr = new int [sc.nextInt()];
    sop ("enter the values");
    for (int i=0; i<arr.length; i++)
    {
        arr [i] = sc.nextInt ();
    }

    sop ("enter 1. for addition");
    sop ("enter 2. for substraction");
    sop("enter 3. for multiplication");
    sop("enter 4. for division");
    int input = sc.nextInt();
    switch(input)
    {
        case 1 : add(arr);
                 break;
        case 2 : sub (arr);
                 break;
        case 3 : mul (arr);
                 break;
        case4 : div(arr);
                break;
        default : sop("Invalid input");
                  break;
    }
}
```

Assignment

1) I/P 10, 20, 30, 40, 50 O/P: 10, 20, 50, 30, 40

```
Public class Rotate {
    static void Rotate (int[] arr)
    {
        int temp = arr[arr.length - 1];
        for (int i = arr.length - 2; i >= arr.length/2; i--
        {
            arr[i+1] = arr[i];
        }
        arr[arr.length/2] = temp;
        for (int j=0; j<arr.length; j+1)
        {
            sop (" " + arr[j]);
        }
    }
    psvm(-)
    {
        int[] arr = {10, 20, 30, 40, 50};
        rotate(arr);
    }
}
```

O/P: 10 20 50 30 40

2) I/P : 10, 20, 30, 40, 50 O/P: 20 30 10 40 50

```
Public class Rotate 1 {
    Static void Rotate (int[] arr)
    {
        int temp = arr[0];
        for (int i=0; i<=arr.length/2; i++)
        {
            arr[i] = arr[i+1];
        }
        arr[arr.length/2] = temp;
        for (int j=0; j<arr.length; j++)
        {
            sop (" " + arr[j]);
        }
    }
    PSVM (-)
    {
        int[] arr = {10, 20, 30, 40, 50};
        rotate(arr);
    }
}
```

3) I/P: 10 20 30 40 50 O/P: 10 40 20 30 50

```
Public class Rotate2
{
 static void rotate (int[] arr)
 {
  int temp = arr [arr.length-2];
  for (int i=arr.length-2; i>=1; i--)
   {
    arr [i] = arr[i-1];
   }
   arr [1]= temp;
   for (int j = 0; j<arr.length; j++)
   {
    sop (" " + arr (j));
   }
 }
 PSVM (-)
 {
  int [] arr = {10, 20, 30, 40, 50};
  rotate(arr);
 }
```

4) I/P: 10 20 30 40 50 O/P: 10 30 40 20 50

```
Public class Rotate3 {
 static void rotate (int[] arr)
 {
  int temp=arr[1];
  for(int i=2; i<=arr.length-2; i++)
   {
    arr[i-1]=arr[i];
   }
   arr [arr.length-2]=temp;
   for (int i=0; i<arr.length; i++)
   {
    sop (" " + arr[i]);
  }
  PSVM (-)
  {
   int [] arr = {10, 20, 30, 40, 50};
    rotate (arr);
  }
 }
```

29/7/23.

Right shift -> size of Array + index - number of rotation

No. of Rotation -> 3

If condition is True

False

arr[size of

Index - no. of rotation

arr[index - no. of rotation

```
int [] arr = {1,2,3,4,5,6,7,8,9,10};
int K = 3
int n = arr.length;
for (i=0;  i<n; i++)
{
    if (i<K)
    {
      sop (arr[n+i-K]);
    }
    else
    {
      sop (arr[i-K]+" ");
    }
}
```

For. n. no. of Rotation

```
int [] arr = {1,2,3,4,5,6,7,8,9,10};
int K = 10;
int n = arr.length;
K = K % n;
for (int i=0; i<n; i++)
{
    if (i<K)
    {
      sop (arr [n+i-K]);
    }
  }
  else{
        sop (arr[i-K]+" ");
    }
}
```

No. of Rotation with dynamic Input with method with Param

```
class Rotation1
{
    static void rotation (int[] arr, int K)
    {
        int n = a.length;
        K = K%n;
        for (int i=0; i<n; i++)
        {
            if (i<K)
            {
                sop (arr [n+i-K]+" ");
            }
            else
            {
                sop (arr [i-K]+" ");
            }
        }
    }
    Public static void main (String[] args)
    {
        Scanner sc = new Scanner (System.in);
        sop ("enter how many values: ");
        int[] arr = new int (sc.nextInt());
        sop ("enter values: ");
        for (int i=0; i<arr.length; i++)
        {
            arr [i] = sc.nextInt ();
        }
        sop ("enter no. of rotation");
        int rotate = sc.nextInt ();
        rotation (arr, rotate);
        }
    }
}
```

31/7/23

→ counting vowels.

```
String str = "India";
int count =0;
for( int i=0; i<str.length(); i++)
 {
      char ch = str.charAt(i);
      if (ch=='a' || ch=='e'|| ch=='i' || ch='o' || ch='u')
       {
         count++;
       }
 } sop (count);
```

printing vowels & consonants.

```
String str= "india"
String vowels = " ";
String consonants = " ";
for (int i=0; i < str.length(); i++)
 {
   char ch = str.charAt(i);
   if (ch = str.charAt(i));
    if (ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
     {
       vowels = vowels+ch;
     }
    else
     {
       consonents = consonents+ch;
     }
   }
   sop (consonents+ " " + vowels);
```

→ Reverse number:-

```
int no = 123;
int rev = 0;
while (no != 0)
{
  int rem = no%10;
  rev = (rev*10) + rem;
  no = no/10;
}
sop(rev); //321
```

123 != 0
12 % 10
123 % 10
(10)123 (12
120
3
10)12 (1
10
2
123
3
0*10 + 3
12*10 + 2
12
1

→ number palindrome

```
int no = 121;
int copy = no;
int rev = 0;
while (no != 0)
{
  int rem = no%10;
  rev = (rev*10) + rem;
  no = no/10;
}
int (copy == rev)
{
  sop ("Number is palindrome");
}
```

→ Sum of number

```
psvm (-){
int no = 646;
sop (add(no));
}
static int add (int no)
{
  int sum = 0;
  while (no != 0)
  {
    int rem = no%10;
    sum = sum+rem;
    no = no/10;
  }
  return sum;
}
```

Till getting single digit

```
psvm (-)
{
int no = 646;
while (no > 9){
  no = add(no);
}
sop(no);
}
static int add (int no)
{
int sum = 0;
while(no != 0)
{
  int rem = no%10;
  sum = sum+rem;
  no = no/10;
}
return sum;
}
```

→ Happy number

```
static int add (int no)
{
  int sum = 0;
  while (no != 0)
  {
    int rem = no%10;
    sum = sum+rem* rem;
    no = no/10;
  }
  return sum;
}
psvm (-)
{
int no = 19;
while (no > 9)
{
  no = add (no);
}
if (no == 1 || no == 7)
{
  sop ("Happy Number");
}
}
```

→ strong number:-

```
int no = 145
int copy = no
int sum = 0;
while (no !=
{
  int rem =
  int fact =
  for (int
  {
    fact =
  }
  sum =
  no = no
}
if (copy
{
  sop ("
}
```

→ write a pro

→ Reverse number:-

```
int no = 123;
int rev = 0;
while (no != 0)
{
    int rem = no % 10;
    rev = (rev * 10) + rem;
    no = no / 10;
}
sop (rev); //321
```

→ number palindrome

```
int no = 121;
int copy = no;
int rev = 0;
while (no != 0)
{
    int rem = no % 10;
    rev = (rev * 10) + rem;
    no = no / 10;
}
int (copy == rev)
{
    sop ("number is palindrome");
}
```

→ Sum of number

```
psvm (-){
int no = 646;
sop (add (no));
}
static int add (int no)
{
    int sum = 0;
    while (no != 0)
    {
        int rem = no % 10;
        sum = sum + rem;
        no = no / 10;
    }
    return sum;
}
```

Till getting single digit

```
psvm (-)
{
int no = 646;
while (no > 9){
    no = add (no);
}
sop (no);
}
static int add (int no)
{
    int sum = 0;
    while (no != 0)
    {
        int rem = no % 10;
        sum = sum + rem;
        no = no / 10;
    }
    return sum;
}
```

→ Happy number

```
static int add (int no)
{
    int sum = 0;
    while (no != 0)
    {
        int rem = no % 10;
        sum = sum + rem * rem;
        no = no / 10;
    }
    return sum;
}
psvm (-)
{
    int no = 19;
    while (no > 9)
    {
        no = add (no);
    }
    if (no == 1 || no == 7)
    {
        sop ("Happy number");
    }
}
```

→ strong number:-

```
int no = 145;
int copy = no;
int sum=0;
while (no !=0)
{
  int rem = no%10;
  int fact = 1;
  for (int i= rem; i>=1; i--)
  {
    fact = fact*i;
  }
  sum = sum+fact;
  no = no/10;
}
if (copy==sum)
{
  sop ("strong number");
}
```

→ Frequency of the string.

```
Public class Mainclass{
  psvm(-){
  String str= "07 ms@hon:143@
                gmail.com";
int[] arr= new int[122];
for(int i=0; i<str.length(); i++)
{ char ch= str.charAt(i);
  arr[ch]++;
}
for(int i=0; i<arr.length; i++)
{ if (arr[i]!=0)
  { sop ((char) i+ "→"+ arr[i]
  }
}
```

→ write a program To find the frequency of the given string

```
String str= "abbababa"
int [] arr= new int [122];
for (int i=0; i<str.length; i++)
{
  char ch = str.charAt(i);
  arr [ch]++;
}
for (int i=0; i<arr.length; i++)
{
  if( arr [i]!=0)
  {
    sop((char) i+ "→" +arr [i]);
  }
}
```

O/P: a→4
b→4.

→ WAP TO check given string are anagram or not.

Psvm→

```
String s1 = "care";
String s2 = "race";
if ( s1.length() == s2.length())
{
  Char[] c1 = s1.toCharArray();
  Char[] c2 = s2.toCharArray();
   Arrays.sort(c1);
   Arrays.sort(c2);
   boolean result = Arrays.equals(c1,c2);
   if (result == true)
   {
     SOP ("Anagram");
   }
   else
   {
     sop ("not a Anagram");
   }
}
   else
   {
     sop("not a Anagram");
   }
}
```

care → [c|a|r|e] → [a|c|e|r]

race → [r|a|c|e] → [a|c|e|r]

Asc

1/8/23

WAP TO find 1st 2nd and 3rd Largest in an unsorted array

```
int [] arr = { 9, 7, 8, 10, 1, 3, 4, 5, 2, 6 }.
class sample
{
 psvm → {
   int[] arr = {9, 7, 8, 10, 1, 3, 4, 5, 2, 6},
arrays.sort (arr);
   for (int i = 1 ; i<=3 ; i++ )
   {
     sop ( arr [arr.length - i] );
   }
```

(OR)

```
sop (arr[arr.length-1] + " " + (arr[arr.length-2] + " " arr[arr.length-3]);
}
```

2) WAP to find the 2 largest in an unsorted array without using inbuilt method.

```
Psvm (-)
{
int[] arr = {9,7, 8,10, 1,3,4,5, 2,6};
   bubble sort (arr.
int[] arr = {9,7,8,10,1,3,4,5,2,6};
int FL=0, SL=0, TL=0;
for (int i=0; i<arr.length; i++)
{
  if (arr[i] > FL)
  {
     TL = SL;
     SL = FL;
     FL = arr[i];
  }
  else if (arr[i] > SL)
  {
     TL = SL;
     SL = arr[i];
  }
  else if (arr[i] > TL)
  {
     TL = arr[i];
  }
sop (FL+" "+SL+" "+TL+" ");
```

FL

SL.

```
FL = SL;
SL = FL;
FL = arr[i];
```

```
TL = SL;
SL = arr[i];
```

2) using without in-built

```
Public class Binarywithoutbuilt{
Psvm (-){
  int no=4;
  string n=" ";
  while (no!=0)
  {
    int rem=no%2;
     n=rem+n;
     no=no/2;
  }
  sop(n);
  }
}
```

3) ASCII Table:-

```
Public class ASCII{
  Psvm (-){
for(int i=0; i<=127; i++)
{
  int no=i;
  string sum=" ";
  while (no!=0)
  {
    int rem = no%2;
    sum= rem+sum;
     no= no/2;
  }
  sop(i+"->"+sum);
}
}
}
```

4/8/23

* Binary Number:-

1) using in-built:-

```
Public class Binary num{
  Psvm(-){
  int no=5;
   sop (Integer.toBinaryString(no));
  }
}
```

2) WAP To find 1st 2nd and 3rd Largest in an unsorted array without using inbuilt method.

```
psvm (-)
{
int[] arr = {9, 7, 8, 10, 1, 3, 4, 5, 2, 6};
   bubble sort (arr,
int[] arr = {9, 7, 8, 10, 1, 3, 4, 5, 2, 6};
int FL=0, SL=0, TL=0;
for (int i=0; i<arr.length; i++)
{
  if (arr[i] > FL)
  {
     TL = SL;
     SL = FL;
     FL = arr[i];
  }
  else if (arr[i] > SL)
  {
     TL = SL;
     SL = arr[i];
  }
  else if (arr[i] > TL)
  {
     TL = arr[i];
  }
sop (FL+" "+SL+" "+TL+" ");
```

4/8/23

* Binary Number:-

1) using in-built:-

```
Public class Binary num {
  Psvm(-) {
  int no=5;
   sop (Integer.toBinaryString(no));
  }
}
```

```
FL        SL.

FL = SL;      TL = SL;
SL = FL;      SL = arr(i);
FL = arr(i);
```

2) using without in-built

```
Public class Binarywithoutbuilt in
Psvm (-){
  int no=4;
  string n=" ";
  while (no!=0)
  {
    int rem=no%2;
     n = rem+n;
     no = no/2;
  }
  sop (n);
  }
}
```

3) ASCII Table:-

```
Public class ASCII{
  Psvm (-){
for (int i=0; i<=127; i++)
{
  int no=i;
  string sum=" ";
  while (no!=0)
  {
    int rem = no%2;
    sum = rem+sum;
     no = no/2;
  }
  sop(i+"—>"+sum);
  }
}
}
```

4) ASCii with char, num & hex

```
Public class ASCii2 {
  PSVm (-){
    for (int i=0; i<=127; i++)
    {
      int no=i;
      String sum= " ";
      while (no!=0)
      {
        int rem=no%2;
        sum = rem+sum;
        no=no/2;
      }
      sop ((char) i+"="+ i+"→" +sum);
    }
  }
}
```

7/8/23

→ WAP TO print 1st letter uppercase in each word

```
Class uppercase{
    Public static void main ( )
    {
      String s1= "hi java how are you";
      char[] ch= s1.toCharArray();
      ch[0]=(char)((char) ch[0]
      ch[0]= (char)(ch[0]-32);
      for (int i=1; i<ch.length; i+1)
      {
        if (ch[i]=='  )
        {
          ch[i+1]=(char)(ch[i+1]-32);
        }
      }
      sop (ch);
    }
}
```

8/8/23

-> WAP To find the distinct number or unique number in an array.

i =	0	1	2	3	4	5	6	7
	1	2	3	1	4	3	5	4
j =	0	1	2	3	4	5	6	7

```
int[] arr = {1,2,3,1,4,3,5,6};
for (int i=0; i<arr.length; i++)
{
    boolean flag = true;
    for (int j=0; i<arr.length; j++)
    {  if (arr[i]==arr[j] && i!=j) {
        flag = false;
        break;
    }
}
    if( flag==true)
    {
      sop (arr[j]);
    }
}
```

arr[i] == arr[j] && i != j	
0 == 1 ✗	
0 == 2 ✗	1 != 1
1 == 1 ✓	false
1 == 2 ✗	
1 == 3 ✗	
1 == 1 ✓	1 != 1
2 == 1 ✗	
2 == 2 ✓	2 != 2
2 == 3 ✗	✗
2 == 1 ✗	
2 == 4 ✗	
2 == 3 ✗	
2 == 5 ✗	

Collection using for-each Loop

```
import java.util.*;
int[] arr = {1, 2, 3, 2, 3, 4, 5, 5};

set <Integer> s1 = new TreeSet<Integer>();
for (int no: arr)
{
    s1.add(no);
}
sop (s1);
```

o/p: [1, 2, 3, 4, 5]

17/8/23

① WAP TO print vowels after consonents

```
String str = "java"
string vowels = " ";
string consonents = " ";
for (int i=0; i< str.length(); i++)
 {
    char ch= str.charAt(i);
    if(ch=='a' || ch=='e' || ch=='i' || ch='o' || ch='u'
     {
        vowels+= ch;
     }
     else
       {
        consonents+ = ch;
       }
     }
      sop (consonents + " " + vowels);
```

② WAP TO ~~print~~ Check given string is palindrome or not without rev & equal method & charAt method

```
static boolean palindrome (string str)
 {
    char[] ch= str.toCharArray;
    int i=0, j= ch.length -1;
    while (i<j)
     {
       if (ch[i] != ch[j])
      {
        return false;
      }
       i++;
       j++;
     }
      return true;
 }
  PSVM (-) {
  string str = "MOM";
  if ( palindrome (str))
   {
```

```
sop ("palindrome");
}
else{
    sop("not palindrome");
  }
}
```

3) WAP To print The next vowel of the given character.

```
psvm (-){
  ch = 'c';
  if ( ch>='a' && ch<'e')
  {
    sop ('e');
  }
else if (ch>='e' && ch<'i')
    {
      sop('i');
    }
else if (ch>='i' && ch<'o')
    {
      sop ('o');
    }
else if (ch>='o' && ch<'u')
    {
      sop ('u');
    }
else ~~(ch>=~~ 'u' && ~~ch<~~
    {
      sop ('a');
    }
}
```

4) I/P: String str = "program"

O/P: uuuiveo

```
psvm (-){
  char[] ch = str.toCharArray();
  for (i=0  i<ch.length; i++)
  Ch = '
  {
    if ( ch[i]>='a' && ch[i]<'e')
      {
        sop ('e');
      }
```

D1

```
else if ( ch[i] >= 'e' && ch[i] < 'i')
        {
            sop ('i');
        }
else if (ch[i] >= 'i' && ch[i] < 'o')
        {
            sop ('o');
        }
else if (ch[i] >= 'o' && ch[i] < 'u')
        {
            sop ('u');
        }
    else
        {
            sop ('a');
        }
}
```

② <u>Panagram</u>

```
import java.util.TreeSet;
String str = "The quick brown fox jumps over the lazy
                dog".
    str = str.toLowerCase();
    TreeSet t1 = new TreeSet();
    for (int i=0; i<str.length(); i++)
    {
        t1.add(str.charAt(i));
    }
    if (t1.size() == 26)
    {
        sop ("Panagram");
    }
}
```

6. input : "2a4b1c"
   O/P : aabbbbc

```
        Sop ("u");
    }
    else (ch[i] >= 'u' && ch[i] < 'a')
    {
        Sop ("a");
    }
}
```

Q5. WAP to check given sentance is Panagram (or) not.

```
import java.util.TreeSet;
String str = "The quick Brown fox jumps Over the lazy
        dog";
    Str = Str.toLowerCase();
    TreeSet tl = new TreeSet();
    for (int i=0 ; i<str.length() ; i++)
    {
        tl.add(Str.charAt(i));
    }
    if (tl.size() == 26)
    {
        Sop ("Panagram");
    }
```

18/8/23

Q1. input : "a2b4c1" ; } → using inbuilt method :
Output : aabbbbc

in → 0 1 2 3 4 5
a 2 b 4 c 1
1 2 3 4 5 6 → length

```
→ PSVM (—) {
    String str = "a2b4c1" ;
    for(int i=0 ; i< str.length(); i++)
    {
        if (charecter.isAlphabetic (str.charAt(i)))
        {
            Sop (str.charAt(i));
        }
        else {
            int no = character.getNumericValue (str.charAt(i))
            for(int k=1 ; k< no ; k++)
            {
                Sop(str.charAt (i-1));
            }
        }
    }
}
```

0<6 ; 1 ; 0 ; a → true ; 2 ; 1 ; 1<2 ; 1-1=0 ; (i) = 0 ⇒ a

Q2. i/p :- 2a4b1c ; o/p :- aabbbbc using inbuilt

```
PSVM (-) {
    String str = "2a4b1c";
    for(int i=0; i< str.length() ; i++)
    {
        if (character. isAlphabetic (str.charAt(i)))
        {
            Sop (str.charAt(i));
        }
        else
        {
            int no = charcter.getNumericValue (str.charAt(i));
            for(int k=1 ; k<no ; k++)
            {   Sop (str.charAt(i+1));
            }
        }
    }
}
```

Q3. i/p : 2a2b2c o/p: aabbcc without using inbuilt

```
PSVM (-) {
    String str = "2a2b2c";
    char[] ch = str.toCharArray ();
    for (int i=0 ; i<ch.length ; i++)
    {   if (ch[i] >= '0' && ch[i] <= 'a')
        {   Print (ch[i] , ch[i+1]);
        }
    }
}
static void Print (char ch1 , char ch2)
{
    int no = (int) (ch1 - 48) ;
    for (int k=1 ; k<=no ; k++)
    {   Sop (ch2);
    }
}
```

Q4. i/p : a2b2c2 o/p : aabbcc without using inbuilt

```
PSVM (-) {
    String str = "a2b2c2" ;
    char[] ch = str. tocharArray () ;
    for (int i=0 ; i<ch.length ; i++)
    {   if (ch[i] >= '0' && ch[i] <= 'a')
        {   print (ch[i], ch[i-1]);
        }
    }
}
```

```
static void Print (char ch1, char ch2)
{
    int no = (int) (ch1-48);
    for (int k=1 ; k<= no ; k++)
    {
        Sop( ch2) ;
    }
}
```

19/8/2023

~~int~~ Q1. IP address :

```
import java.net. InetAddress ;
import java.net. Inet4Address ;
import java.net. Inet6Address ;
Import java.net. UnknownException ;
class IPAddressValidation
{
    PSVM (—)
    {
        String str= "126.166.50.50" ;
        try {
            InetAddress ipa = InetAddress.getByName (str);
            if (ipa instanceof Inet4Address)
            {
                Sop (ipa + " is valid IPV4 Address") ;
            }
            else if (ipa instance of Inet6Address) ;
            {
                Sop (ipa + " is valid IPV6 Address") ;
            }
        }
        Catch (UnknownHostException e)
        {
            Sop("Invalid IP address") ;
        }
    }
}
```

Output: / 126.166.50.50 is valid IPV4 Address.

21/8/2023

Q1. WAP to print maximum repeated char in a string.

i/p :- hello o/p :- L → 2

```
PSVM (—) {
    String str = "hello";
    int[] array = new int[122];
    for (int i=0; i< str.length; i++)
    {
        char ch = Str.charAt(i);
        array[ch]++;
    }
    int max = -1;
    char ch = ' ';
    for (int i=0; i<str.length(); i++)
    {
        if (max < array[str.charAt(i)])
        {
            max = array[str.charAt(i)];
            ch = str.charAt(i);
        }
    }
    Sop (ch + " → " + max);
}
```

(OR)

```
if (array(i)!=0)
{
    if (max<array(i))
    {
        max = array(i);
        ch=str.charAt(i);
    }
}
```

OR

Q2. WAP to print mini repeated char in a string.

```
String Str = "hello";
int[] array = new int[122];
for (int i=0; i<str.length; i++)
{
    char ch = Str.charAt(i);
    array[ch]++;
}
int min = 10;
char ch = ' ';
for (int i=0; i<str.length(); i++)
{
    if (min< array[Str.charAt(i)]).
    {
        min = array[Str.charAt(i)];
        ch = Str.charAt(i);
    }
}
Sop (ch + " → " + min);
```

33. i/p :- Java is Easy
o/p :- aavj si yase.

→ PSVM(—) {

```
String str = "Java is easy";
char[] ch = Str.tocharArray();
for (int i=0; i<ch.length; ~~i++~~)
{
    int k=i; // k=0 // k=5 // k=8
    while(i<ch.length && ch[i] != ' ')
    {
        i++;
    }
    char temp = ch[k]; // j
    ch[k] = ch[i-1]; // a
    ch[i-1] = temp; // j
}
Sop(ch);
```

⟹ Presentation :

Q1. all with method with parameter :
i) Bubble Sort ;
ii) first char should be upperCase [without using inbuilt]

22/8/2023

One-D array

arr = | 1 | 2 | 3 | 4 | 5 | 6 |

```
int[] arr = {1,2,3,4,5,6};
for (int i=0; i<arr.length; i++)
{
    Sop (arr[i] + " ");
}
```

Two-D array

```
int [][] = {{1,2,3,4}, {5,6,7,8}
{9,10,11,12}, {13,14,15,16}};
for(int row=0; row< arr.length;
        row++)
{
    for(int col=0; col< arr[row].length
        ; col++)
    {
        Sop( arr[row][col] + " ");
    }
    Sopln ();
}
```

Col: 0	1	2	3
r0 0,0	0,1	0,2	0,3
1 1,0	1,1	1,2	1,3
2 2,0	2,1	2,2	2,3
3 3,0	3,1	3,2	3,3

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Q2. Output : Bheema :

```
import java.util.Scanner;
public class TwoDname {
  PSVM (-) {
    Scanner sc = new Scanner (System.in)
    Sop("enter which row");
    int a=sc.nextInt();
    Sop("enter which col");
    int b= sc.nextInt();
    String[][] str={{"Dharmaraya", "Arjuna", "Bheema"},{"Nakula",
    "Sahadeva", "Karna"},{"Dhuryodhana", "Bheshma", "Krishna"}};
  for(int row=0; col< str.length ; row++)
  {
    for (int col=0 ; col< str[row].length ; col++)
    {  Sop(str[a][b]);
       break;
    }
    Sop();
    break;
  }
}
}
```

Q3. o/p : 2 is in 4 index :

```
PSVM (-) {
 int[] arr={4,1,3,5,2,8,6,7};
 int k=2;
 for(int i=0; i<arr.length; i++)
 {
   if(arr[i]==k)
   {
     Sop(k+"is in"+i+"index");
   }
 }
}
}
```

Q4. i/p :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

o/p : 1+6+11+16 = 34

```
PSVM (-) {
 int[][] arr = {{1,2,3,4},{5,6,7,8},
  {9,10,11,12},{13,14,15,16}};
 int sum=0;
 int n = arr.length;
 for(int i=0; i<n ; i++)
 {   sum= sum+arr[i][i];
 }
 Sop(sum);
}
```

Q5. i/p :

23	2	3	4
5	15	7	8
9	15	21	12
13	14	19	16

o/p : 36

Diff of both

```
PSVM (-) {
 int[][] arr={{23,15,21,16},{5,15,7,8},
  {9,15,21,12},{13,14,19,16}};
 int sum=0;
 int sum1=0;
 int n=arr.length;
 for(int i=0 ; i<n ; i++)
 {
   sum= sum+ arr[i][i];
   sum1= sum1+ arr[i][n-1-i];
 }
 Sop(sum- sum1);
```

Q6. i/p :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

o/p : 34

```
PSVM (-) {
 int[][] arr={ ------ };
 int sum=0;
 int n = arr.length;
 for(int i=0; i<n ; i++)
 {   sum=sum+arr[i][n-1-i];
 }
 Sop(sum);
```

Q7. output : In ~~2~~ 3 row 3 col krishna is there

```
import java.util. Scanner ;
public static TwoD {
  PSVM (-) {
    Scanner sc = new Scanner (System.In);
    ~~int~~ Sop ("enter which row");
    int a = sc.nextInt() ;
    Sop ("enter which col");
    int b = sc. nextInt();
    String ~~str~~ [][] str = {- - -} ;
    for (int row=0 ; row<str.length ; row++)
    {
      for (int col =0 ; col< str [row] .length ; col++)
      {
        Sop("in"+ a + "row" + b+ "col" + str[a][b] + "is there");
        break ;
      }
      sop () ;
      break;
    }
  }
}
```