Java

Assignment 5

1.
```
public static short countBits (int x) {
    short numBits = 0;
    while (x != 0) {
        numBits += (x & 1);
        x >>>= 1;
    }
    return numBits;
}
```

Pr:

x = 1010 &
1 (Mask)

0 ⟶ extracted Sum = 0

Sum = 0 + 1 + 0 + 1 = 2.

2.
```
public static short parity (long x) {
    short result = 0;
    while (x != 0) {
        result ^= (x & 1);
        x >>>= 1;
    }
    return result;
}
```

(3). public static long swapBits (long x, int i, int j) {

```
if ( ((x >> i) & 1) != ((x >> j) & 1)) {
    long bitMask = (1L << i) | (1L << j);
    x ^= bitMask;
}
return x;
}.
```

⇒ 1 0 1 1 0 1 0 x
j i

(4). public static int multiply (int a, int b) {

```
int res = 0;
while (b > 0) {
    if ((b & 1) != 0)
        res = res + a;
    a = a << 1;
    b = b >> 1;
}
return res;
}
}
```

(5).

(6).

⑤.
```
psvm (S[] args) {
    StringBuffer sbf = new StringBuffer(s);
    S.out.p("string buffer =" +sbf.);
    sbf.reverse();
    S.out.p("S. after reverse =" + sbf);
    sc.close();
    }
}
```

⑥.
```
p.s.v.m
    if(s==null || s.isEmpty())
        S.out("String has no words.");
    String[] words = s.split(" ");
    Sys.out.("The string has" + words.length+
                "words.");
    sc.close();
    }
}
```

(7) Password :-

```
Enter Name
Enter DOB
String name = sc.nextLine();
String dob = sc.nextLine();

String psw = (name.substring(0, 2)).
                toLowerCase();

Switch (dob.substring(3,5)) {

case "01": psw = psw+"january";
break;
case "02": psw = psw+"february";
break;
  }
}
S.out.p.("Y. password is = "+psw);
sc.close();
}.
```

(8)

(8) psvm (s[]a) {

Enter A: ;
String a = sc.nextLine();
En B: ;
String b = sc.nextline();

String res = String.valueOf(Integer.valueOf(a)
+ Integer.valueOf(b));

S.out.(" the result is = " + res);
sc.close();
}
}

(9). p.s.v.printString (int columnNumber) {

```
StringBuilder columnName = new StringBuilder();
while (columnNumber > 0) {
    int rem = columnNumber % 26;
    if (rem == 0) {
        columnName.append("Z");
        columnNumber = (columnNumber/26) - 1;
    }
    else {
        columnName.append((char)((rem-1)+'A'));
        columnNumber = columnNumber/26;
    }
}
S.out(columnName.reverse());
}
p.s.v.m(S[] ar) {
    for (i=1; i<=27; i++) {
        printString(i);
        S.out("i is encoded to "+i);
    }
}
```

(11).

(13) (a).

(12)
```
class ReverseWords {
    p s v m ( s[] A) {
        String s[] = "you shall not pass ".split(" ");
        String ans = "";
        for (int i = s.length-1 ; i>= 0 ; i--) {
            ans += s[i] + " ";
        }
        S.out ("Reversed String : " + ans);
    }
}
```

(13) (a)
```
class GFG {
    static int frequencyDigits (int n, int d) {
        int c = 0;
        while (n>0) {
            [if (n%.10 == d)
                c++;
            n = n/10;
        }
        return c;
    }
    p.s.v.m
    int N = 1122322;
    int D = 2
    S.out (frequencyDigits(N,D));
    }
}
```

(14). static int frequencyDigits (int[] n, int d){
int c = 0;
for (int i = 0; i < n.length; i++) {
if (n[i] % 10 == d)
c++;
n[i] = n[i] / 10;
}
return c;
}
p.s.v.m ( ) {
int[] a = {1, 2, 4, 5, 5, 6, 7, 3};
int d = 5;
S.out (frequencyDigits (a, d));
}
}

count Prime Number :-

(15)

```
int A[] = { 4,5,9,6,7,7,7,5,3 }
int count = 0;
for(i=0; i< A.length ; i++) {

    int a = 2, temp = 0;
    while ( A[i] <
    while ( a <= ( A[i]/2 )) {
        if ((A[i] % a) == 0) {
            temp ++;
            break;
        }
        a++;
    }
    if (temp == 0 && i != 1) {
        count++;          (A[i] % 2 == 1)
    }
}
S.out.("T. Prime No. is :"+count);
```

(16).
```
public static void m(S.[] ar) {
    int [] arr = new int[] {1,2,3,4,5};
    int n = 3;
    S.out ("Original arr.. ");
    for (int i=0; i< arr.length ; i++) {
        Sy.out (arr[i] + " ");
    }
    for (int i= 0; i<n ; i++) {
        int j, last;
        last = arr[arr.length -1];
        for (j= arr.length -1 ; j> 0; j--) {
            arr[j] = arr[j-1];
        }
        arr[0] = last;
    }
    S.out();
}
```

(17) Static int removeDuplicates (int arr[], int n){

```
if (n==0 || n==1)
    return ;
int[] temp = new int[n];
int j = 0;
for (int i=0; i<n-1; i++)
    if (arr[i] != arr[i+1])
        temp[j++] = arr[i];

temp[j++] = arr[n-1];
for (int i=0; i<j; i++)
    arr[i] = temp[i];
return j;
}
```

psvm

```
int arr[] ={1,2,2,3,4,4,4,5,5};
int n = arr.length;
n = removeDuplicates (arr,n);

for (int i=0; i<n ;i++)
    S.out (arr[i] + " ");
}.
```

(18) Merging of Sorted array :-

```
public s. v. mergeArrays (int [] arr1, int [] arr2,
                          int n1, int n2, int [] arr3){
    int i=0, j=0, k=0;
    while (i< n1){
        arr3[k++] = arr1[i++];
    }
    while (j< n2){
        arr3[k++] = arr2[j++];
    }
    Arrays. sort (arr3);
}
public st. v. main (String[] args){

    int arr1[] = {1,3,5,5};
    int n1 = arr1. length;
    int arr2[] = {2,4,6,8};
    int n2 = arr2. lungth;
    int arr3[] = new int[n1+n2];

    mergeArrays (arr1, arr2, n1, n2, arr3);
    s. out ("Array After maging");
    for (int i=0; i< n1+n2; i++)
        s. out (arr3[i] + " ");
}
```

(19). Insert an element in an array in spec. position.

```
prob. 8. int [] insertX (int n, int arr[], int x,
                                    int pos) {
    int i;
    int newarr[] = new int[n+1];
    for (i=0; i<n+1; i++){

        if (i < pos - 1)
            newarr[i] = arr[i];
        else if (i == pos - 1)
            newarr[i] = x;
        else
            newarr[i] = arr[i+1];
    }
    return newarr;
}
p s v m (S[] ar) {
    int n = 10;
    int i;
    int arr[] = {1,2,3,4,5,6,7,8,9,10};
    S.out("Initial array: \n" + Arrays.to
                        String(arr));

    int x = 50;
    int pos = 5;
    arr = insertX(n, arr, x, pos);
```

```
S.out.("\n" + x + "\n"
        + pos + ": ".
        + Arrays.toString(arr));
}
```

(20). Get, Set, Constructor → Student :- [Array].

```
class Student {
    String name;
    int rn, age;
    public String getName() {
        return name;
    }
    int getRn() {
        return rn;
    }
    int getAge() {
        return age;
    }
    void setName(String name) {
        this.name = name;
    }
    void setRn(int rn) {
        this.rn = rn;
    }
```

```
void setAge (int age) {
    this.age = age;
}
Student (String name, int rn, int age) {

    this.name = name;
    this.rn = rn;
    this.age = age;
}
void display () {

    S.out ("Name : "+ getName()+ "\nRn:"
        + getRn() + "\nAge : "+getAge());
}
}
p. class eg19 {
    p. s. v. main (S[] a) {
        Scanner s = new Scanner (System.in);
        Student [] S1 = new Student [5];
        for (int i=0; i<5 ; i++) {
            "Enter name");
        String name = s.next ();
            "Enter Rn
            Enter Age.

        S1 [i] = new Student (name, rn, age);
        S1 [i] . display ();
    }
}
```

(21).
```
class Num {
    int [] arr ;
    int [] getArr () {
        return arr ;
    }
    void setArr (int [] arr) {
        this.arr = arr ;
    }
    Num (int [] arr) {
        this.arr = arr ;
    }
    static int [] reverseArray (int arr [], int
                    start , int end) {
        int temp ;
        while ( start < end) {
            temp = arr [start ] ;
            arr [start ] = arr [end ] ;
            arr [end ] = temp ;
            start ++ ;
            end -- ;
        }
        return arr ;
    }
}
p. class aq20 {
```

```
p. s v. main (String[] a) {
    int arr[] = {1, 2, 3, 4, 5, 6};
    Num n1 = new Num(arr);
    int res[] = n1.reverseArray
                (arr, 0, arr.length - 1);
    for (int i=0; i< res.length; i++)
        S.out (res[i] + " ");
}
}
```

(22) Wrapper class :-

```
p. s. v. m ( ) {
    Integer i = new Integer(42);
    byte b = i.byteValue();
    short s = i.shortValue();
    S.out(i + " " + b + " " + s);
}
```

(23) Value Of method :-

```
p s v m ( ) {
    Integer i = Integer.valueOf("1010", 2);
    Float f = Float.valueOf("5.6f");
    S.out(i + " " + f);
}
```

Auto Boxing :-

(24).
```
p. s. v. m (s) {
    List<Integer> a = new ArrayList<>();
    for(int i=0 ; i<10 ; i++) {
        a.add(i);
    }
}
```

Auto Unboxing :-

```
p. s. v. m (String ar[]) {
    ArrayList<Integer> nums = new
        ArrayList<Integer>();

    nums.add(1);
    nums.add(15);
    nums.add(20);
    S.out("Total is " + addNumbers
        (nums));

}
```

p.s

```
int addNumbers(List<Integer> nums){
    int total = 0;
    for (Integer num : nums){
        total += num;
    }
    return total;
}
```