

```

public class Suite1 {
    public static void main(String args[]) {
        int x, n, i, xi = 1, t = 0;
        System.out.print("x = ");
        x = Lire.i();
        System.out.print("n = ");
        n = Lire.i();
        for(i = 0; i <= n; i++) {
            t += xi;
            xi *= x;
        }
        System.out.println(t);
    }
}

public class Suite2 {
    public static void main(String args[]) {
        int x, n, a, i, xi = 1, t = 0;
        System.out.print("x = ");
        x = Lire.i();
        System.out.print("n = ");
        n = Lire.i();
        for(i = 0; i <= n; i++) {
            System.out.print("a" + i + " = ");
            a = Lire.i();
            t += xi * a;
            xi *= x;
        }
        System.out.println(t);
    }
}

=====

public class Test_tri {
    public static void main(String args[]) {
        int T[] = { 1, 2, 2, 5, 7, 6, 8, 9 };
        int i;
        boolean trie = true;
        for(i = 0; i < T.length - 1; i++) {
            if(T[i] > T[i + 1]) {
                trie = false;
                break;
            }
        }
        System.out.println(trie ? "oui" : "non");
    }
}

=====

public class Phrase {
    public static void main(String args[]) {
        String phrase, mot[] = new String[3];
        int n_mots = 0, debut, fin, i, j;
        System.out.print("Phrase: ");
        phrase = Lire.S();
        debut = 0;
        while(debut < phrase.length() && n_mots < 3) {
            fin = phrase.indexOf(' ', debut);
            if(fin == -1)

```

```

        fin = phrase.length();
        mot[n_mots] = phrase.substring(debut, fin);
        n_mots++;
        debut = fin + 1;
    }
    if(n_mots < 3) {
        System.out.println("Non (moins de trois mots)");
    } else if(debut < phrase.length()) {
        System.out.println("Non (plus de trois mots)");
    }
    for(i = 0; i < 3; i++) {
        if(mot[i].length() == 0) {
            System.out.println("Non (le mot " + i + " n'en est pas un)");
            break;
        }
        for(j = 0; j < mot[i].length(); j++) {
            char c = mot[i].charAt(j);
            if((c < 'a' || c > 'z') && (c < 'A' || c > 'Z')) {
                System.out.println("Non (le mot " + i + " contient autre chose
" +
                "qu'une lettre)");
                break;
            }
        }
        if(j < mot[i].length())
            break;
    }
    String sujet, verbe, complement;
    if(i == 3) {
        sujet = mot[0];
        verbe = mot[1];
        complement = mot[2];
        if(((sujet.equals("je") || sujet.equals("il")) &&
verbe.endsWith("e")) ||
            (sujet.equals("tu") && verbe.endsWith("es")) ||
            (sujet.equals("nous") && verbe.endsWith("ons")) ||
            (sujet.equals("vous") && verbe.endsWith("ez")) ||
            (sujet.equals("ils") && verbe.endsWith("ent")))) {
            System.out.println("Oui");
        } else {
            System.out.println("Non (sujet inconnu ou conjugaison
incorrecte)");
        }
    }
}
}
}

```

```

public class Tetris {
    public static boolean peut_tomber(int [][]tetris) {
        int x, y;
        for(y = 0; y < 20; y++) {
            for(x = 0; x < 10; x++) {
                if(tetris[y][x] >= 11 && tetris[y][x] <= 19 &&
                    (y == 0 ||
                    !(tetris[y - 1][x] == 0 ||
                    (tetris[y - 1][x] >= 11 && tetris[y - 1][x] <= 19)))) {
                    return false;
                }
            }
        }
    }
}

```

```

    }
}
return true;
}

public static void descend(int [][]tetris) {
    int x, y;
    for(y = 0; y < 20; y++) {
        for(x = 0; x < 10; x++) {
            if(tetris[y][x] >= 11 && tetris[y][x] <= 19) {
                tetris[y - 1][x] = tetris[y][x];
                tetris[y][x] = 0;
            }
        }
    }
}

public static void pose_et_supprime(int [][]tetris) {
    int x, y, y2;
    boolean plein;

    y = 0;
    while(y < 20) {
        plein = true;
        for(x = 0; x < 10; x++) {
            if(tetris[y][x] >= 11 && tetris[y][x] <= 19)
                tetris[y][x] -= 10;
            if(tetris[y][x] == 0)
                plein = false;
        }
        if(plein) {
            for(y2 = y + 1; y2 < 20; y2++) {
                for(x = 0; x < 10; x++) {
                    tetris[y2 - 1][x] = tetris[y2][x];
                }
            }
            for(x = 0; x < 10; x++) {
                tetris[19][x] = 0;
            }
        } else {
            /* il ne faut passer à la ligne suivante que si on n'a pas
dÃ©calÃ© */
            y++;
        }
    }
}

public static void affiche(int [][]tetris) {
    int x, y;
    String c;
    System.out.println();
    for(y = 19; y >= 0; y--) {
        System.out.print("| ");
        for(x = 0; x < 10; x++) {
            c = "" + tetris[y][x];
            if(c.length() == 1) {
                c = "0" + c;
            }
            if(c.equals("00")) {
                c = "..";
            }
        }
    }
}

```

```

        }
        System.out.print(c + " ");
    }
    System.out.println("|");
}
System.out.println("+-----+");
}

public static void main(String args[]) {
    int tetris[][] = {
        { 1, 1, 3, 2, 2, 2, 7, 1, 7, 2 },
        { 1, 3, 3, 0, 2, 7, 7, 1, 1, 2 },
        { 1, 5, 0, 0, 0, 7, 0, 5, 5, 0 },
        { 1, 0, 0, 0, 0, 0, 0, 4, 5, 0 },
        { 0, 0, 0, 0, 0, 0, 4, 4, 4, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 0, 16, 0, 0, 0, 0, 0, 0 },
        { 0, 0, 16, 16, 16, 0, 0, 0, 0, 0 },
    };
    while(peut_tomber(tetris)) {
        descend(tetris);
        affiche(tetris);
    }
    System.out.println("exit");
    pose_et_supprime(tetris);
    affiche(tetris);
}
}

```