Palindrome recursive

```
public static boolean
palindromecheck(String temp) {
boolean result = false;
temp = temp.t oLo wer Case();
temp = temp.r epl ace All (" ",
" ");
temp = temp.r epl ace All ("\"
", " ");
temp = temp.r epl ace All ("\ -
\?", " ");
temp = temp.r epl ace All ("!",
" ");
temp = temp.r epl ace All -
("\' ", " ");
temp = temp.r epl ace All -
("\\.", " ");
temp = temp.r epl ace All -
(",", " ");
if (temp.l en gth() > 1) {
if (temp.c ha rAt(0) == temp.c -
har At( tem p.l eng th( )-1)) {
palind rom ech eck (te mp.s ub -
str ing(1, temp.l eng th( )-1));
result = true;
}
else {
result = false;
}
}
return result;
}
```

Sorting(Simple one)

```
public static void
sorting(String[] list) {
for (int x = 0;x < list.l eng -
th;x++) {
for (int y = x + 1 ;y < list.l -
eng th;y++) {
// indexo farray
if (list[ x].c om par eTo (li -
st[y]) > 0) {
```

Sorting(Simple one) (cont)

```
> continue;
}
else if (list[x].compareTo(list[y]) == 0) {
continue;
}
else {
String temp = list[x];
list[x] = list[y];
list[y] = temp;
}
}
System.out.print("Pass: "+x);printlist(list);
}
}
```

File IO (Write)

```
public class write {
public static void main(S -
tring[] args) {
Buffer edW riter writer = null;
File file = new File("l og.t -
xt ");
try {
System.ou t.p rin tln ("te st");
writer = new Buffer edW rit -
er(new FileWr ite r(f ile));
writer.wr ite ("Hello World.W -
at ??? \n");
writer.ap pen d("Hello Mars. -
\n");
}
catch (FileN otF oun dEx ception
e) {
e.prin tSt ack Tra ce();
}
catch (IOExc eption e) {
e.prin tSt ack Tra ce();
```

File IO (Write) (cont)

```
> }
finally {
try {
if (writer != null) {
writer.close();
}
}
catch (IOException e) {
e.printStackTrace();
}
}
}
}
```

Node recursive

```
public static void
inOrderTraverse(Node root)
{
if (root == null) {
}
else {
//Syst em.o ut.pr int (" " +ro -
ot.id); //pre order
inOrde rTr ave rse (ro ot.l -
eft);
System.ou t.p rin t(" " +ro -
ot.id); // in order
inOrde rTr ave rse (ro ot.r -
ight);
//Syst em.o ut.pr int (" " +ro -
ot.id); // post order
}
}
```

getSnippet

```
public static String
getSnippet(Word root, int
window)
{
if(root == null) return null;
String Builder str = new
String Bui lder();
str.ap pen d("[ " +ro ot+ "] ");
Word pointer = root.g etP rev -
iou sWo rd();
for(int i = 0; i < window; i++)
{ if(pointer == null)
{
break;
}
else
{ str.in sert(0, pointe r.g -
etW ord ()+ " ");
pointer = pointe r.g etP rev -
iou sWo rd();
}
}
if(pointer != null) str.in -
sert(0, "... ");

pointer = root.g etN ext Word();
for(int i = 0; i < window; i++)
{ if(pointer == null)
{
str.ap pen d("... ");
break;
}
else
{ str.ap pen d(p oin ter.ge -
tWo rd( )+" ");
pointer = pointe r.g etN ext -
Word();
}
}
```

getSnippet (cont)

```
> if(pointer != null) str.append("... ");

return str.toString().trim();
}
```

How to create array of int and double

```
int a[]=new int[5];
data = new Double [10];
```

Switch case

```
int day = 4;
switch (day) {
   case 6:
        Sys tem.ou t.p rin tln -
("Today is Saturd ay");
        break;
   case 7:
        Sys tem.ou t.p rin tln -
("Today is Sunday ");
        break;
    def ault:
        Sys tem.ou t.p rin tln -
("Lo oking forward to the
Weeken d");
}
// Outputs " Looking forward to
the Weeken d"
```

Compareto

```
result will be int
> 0 : greater
=0 : similar
<0: least
```

isFullBinTree (Node)

```
public static boolean
isFullBinTree(Node root)
{
boolean result = false;
if (root == null) {
result = true;
}
else {
if (root.left != null &&
root.right != null) {
if (!isFu llB inT ree (ro ot.l -
eft) || !isFul lBi nTr ee( roo -
t.r ight)) {
result = false;
}
else {
result = true;
}
}
else if (root.left == null &&
root.right == null) {
result = true;
}
else {
result = false;
}
}
return result;
}
```

binaryfind

```
public static boolean
binaryfind(int[][] matrix, int
window) {
        boolean check = false;
       int length = matrix -
[0].length * matrix [1].le ngth;
int low = 0;
int high = sorted Wor ds.s ize()
- 1;
int mid = 0;
while (low <= high) {
mid = low + ((high - low)/2);
if (sorte dnu mbe r[mid] ==
target) {
return true;
}
else if (sorte dnu mbe r[mid] <
target) {
low = mid + 1;
}
else {
high = mid - 1;
}
}
return check;
}
```

File IO (Read apache)

```
//load file
String text = null;
try {
text = FileUt ils.re adF ile -
ToS tri ng(new File(f ile name),
" UTF -8");
} catch (IOExc eption e) {
e.prin tSt ack Tra ce();
}
```

File IO (Read apache) (cont)

```
> //clean text
text = text.toLowerCase().replaceAll("\\W+",
" ").replaceAll("\\s\\w\\s", " ");

//System.out.println(text);
//tokenize words
String[] tokens = text.split("\\s+");

//store tokens in words
sortedWords = new ArrayList<Word>();
for(int i = 0; i < tokens.length; i++)
{ sortedWords.add(new Word(tokens[i],i));
if(i > 0)
{ sortedWords.get(i).setPreviousWord(sort-
edWords.get(i-1));
sortedWords.get(i-1).setNextWord(sorted-
Words.get(i));
}
}
//sort word
Collections.sort(sortedWords);
```