public String changePi(String str) {

String s="";

if(str.length()>1){

if(str.charAt(0)=='p' && str.charAt(1)=='i'){

s=s+"3.14"+changePi(str.substring(2));

}else{

s=s+str.charAt(0)+changePi(str.substring(1));

}

}else{

if(str.length()==1){s=s+ str.charAt(0);}else{ s=s;}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public String noX(String str) {

String s="";

if(str.length()>1){

if(str.charAt(0)=='x'){

s=s+noX(str.substring(1));

}else{

s=s+str.charAt(0) + noX(str.substring(1));

}

}

else{

if(str.length()!=0){

if(str.charAt(0)=='x'){s=s;}else{s=s+str.charAt(0);}

}

}

return s;

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean array6(int[] nums, int index) {

boolean f=false;

if(index<nums.length){

if(nums[index]==6){f=true;}else{ f=array6(nums, index+1);}

}

return f;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public int array11(int[] nums, int index) {

int s=0;

if(index<nums.length){

if(nums[index]==11){

s=s+1+ array11(nums,index+1);

}else{

s=s+0+array11(nums,index+1);

}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean array220(int[] nums, int index) {

boolean f=false;

if(index<(nums.length -1)){

if(nums[index+1]==(nums[index]\*10)) {

f=true;

}else{ f=array220(nums,index+1); }

}else{

f=f;

} return f; }

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public String allStar(String str) {

String s="";

if(str.length()>1){

s=s+str.charAt(0);

if((str.length() -1)>0){ s=s+"\*" + allStar(str.substring(1));}

}else{

if(str.length()!=0){

s=s+str.charAt(0);}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public String pairStar(String str) {

String s="";

if(str.length()>2){

if(str.charAt(0)==str.charAt(1)){

s=s+str.charAt(0)+"\*" + pairStar(str.substring(1));

}else{

s=s+str.charAt(0)+ pairStar(str.substring(1));

}

}else{

if(str.length()==2){

if(str.charAt(0)==str.charAt(1)){

s=s+str.charAt(0) +"\*"+ str.charAt(1);

}else{

s=s+str.charAt(0)+ str.charAt(1);}

}else if(str.length()==1){ s=s+str.charAt(0);} else{s=s;}

}

return s;

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public String endX(String str) {

String s="";

if(str.length()>2){

if(str.charAt(0)=='x'){

s=endX(str.substring(1) ) +"x";

}

else{

s=str.charAt(0)+endX(str.substring(1) ) ;

}

}

else{

if(str.length()==2){

if(str.charAt(0)=='x'){

if(str.charAt(1)=='x'){

s=s+"x"+"x";

}else{

s=s+str.charAt(1) + "x";

}

}else

if(str.charAt(1)=='x'){

s=s+str.charAt(0)+"x";

}else{

s=s+str.charAt(0)+str.charAt(1);}

}else if(str.length()==1){

if(str.charAt(0)=='x'){ s=s+"x";}else{ s=s+str.charAt(0);}

}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public int countPairs(String str) {

int s=0;

if(str.length()>3){

if(str.charAt(0)==str.charAt(2)){

s=s+1 + countPairs(str.substring(1));

}else{

s=s+countPairs(str.substring(1));

}

}else{

if(str.length()==3){

if(str.charAt(0)==str.charAt(2)){ s=s+1;}

}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public int countAbc(String str) {

int s=0;

if(str.length()>2){

if(str.substring(0,2).equals("ab") && (str.charAt(2)=='c' || str.charAt(2)=='a') ){

s=s+1+ countAbc(str.substring(1));

} else{

s=s+ countAbc(str.substring(1));

}

}else{

s=s+0;

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public int count11(String str) {

int s=0;

if(str.length()>2){

if(str.charAt(0)=='1'&&str.charAt(1)=='1'){

s=s+1+count11(str.substring(2));

}else{

s=s+count11(str.substring(1));

}

}else{

if(str.length()==2){ if(str.charAt(0)=='1' && str.charAt(1)=='1'){s=s+1;}}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public String stringClean(String str) {

String s="";

if(str.length()>=2){

if(str.charAt(0)==str.charAt(1)){

s=s+(String)stringClean(str.substring(1));

}else{

s=s+str.charAt(0) + (String)stringClean(str.substring(1));

}

}else{

if(str.length()==1){s=s+str.charAt(0);}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public int countHi2(String str) {

int s=0;

if(str.length()>2){

if(str.charAt(0)=='x' && (str.charAt(1)=='h'&&str.charAt(2)=='i')){

s=s+countHi2(str.substring(2));

}else{

if(str.charAt(0)=='h'&&str.charAt(1)=='i'){

s=s+1 + countHi2(str.substring(1));

}else{

s=s+ 0+countHi2(str.substring(1));

}

}

}

else{

if(str.length()==2){ if (str.charAt(0)=='h' && str.charAt(1)=='i'){s=s+1;}}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public String parenBit(String str) {

String s="";

if(str.length()>0){

if(str.charAt(0)!='('){

s=parenBit(str.substring(1));

return s;

}else if(str.charAt(str.length()-1)==')'){

s=str.substring(0,str.length());

return s;

} else if (str.charAt(str.length()-1)!=')'){

s=parenBit(str.substring(0,str.length()-1));

}else{

s=parenBit(str.substring(1));

}

}

return s;}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean nestParen(String str) {

boolean f=true;

if(str.length()>1){

if(str.charAt(0)=='('){

if (str.charAt(str.length()-1)==')'){f=true && nestParen(str.substring(1,str.length()-1)); } else{

f=false;

}

}else if (str.charAt(str.length()-1) ==')'){

f=false;

} else {

f=f && nestParen(str.substring(1,str.length()-1));

}

}else{

if(str.length()==1){

if(str.charAt(0)=='(' || str.charAt(0)==')'){f=false;}else{f=true;}

}

}

return f;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public int strCount(String str, String sub) {

int s=0;

if(str.length()>sub.length()){

if(sub.equals(str.substring(0,sub.length() ) )){

s=s+1+strCount(str.substring(sub.length()),sub);

}else{ s=s+strCount(str.substring(1),sub);}

}else{

if(str.length()==sub.length()){

if(sub.equals(str)){s=s+1;}

}

}

return s;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean strCopies(String str, String sub, int n) {

boolean b=false;

if(n<=0){b=true; return b;}else{

if(str.length()>sub.length()){

if(sub.equals(str.substring(0,sub.length()))){

if((n-1)<=0){ b=true;}else{

b= strCopies(str.substring(1),sub,n-1) ;

}

}else{

if(n<=0){b=true;}else{

b=strCopies(str.substring(1),sub,n);

}

}

}else{

if(str.length()==sub.length()){

if(sub.equals(str)){ if(n-1<=0){b=true;}else{b= false;}}

else{ if(n<=0){ b=true;}else{b=false;}}

} else{

if(n<=0){b=true;}else{b=false;}

}

}

return b;

}

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean groupSum(int start, int[] nums, int target) {

boolean b=false;

if(target==0){

b=true;

return b;

}else{

if(start<nums.length){

b=groupSum(start+1,nums,target)|| groupSum(start+1,nums,target-nums[start]);

return b;

}else{ if(target==0){b=true; return b;}else{b=false;return b;}}

}

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean groupSum6(int start, int[] nums, int target) {

if(start>=nums.length){return target==0;}

if(nums[start]==6){

if(groupSum6(start+1,nums,target-6)){return true;}

return false;

}else {

if(groupSum6(start+1,nums,target-nums[start])){return true;}

if(groupSum6(start+1,nums,target)){ return true;}

return false;

}

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean groupNoAdj(int start, int[] nums, int target) {

if(start>=nums.length){ return target==0;}

if(groupNoAdj(start+2,nums,target-nums[start])) return true;

if( groupNoAdj(start+1,nums,target)) return true;

return false;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean groupSum5(int start, int[] nums, int target) {

if(start>=nums.length){return target==0;}

if((nums[start]%5)==0){

if(start+1<nums.length &&(nums[start+1]/5)!=1){

if(groupSum5(start+2,nums,target-nums[start] -nums[start+1]))return true;

}else{

if(groupSum5(start+1,nums,target-nums[start])) return true;

}

}

if(groupSum5(start+1,nums,target))return true;

return false;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean groupSumClump(int start, int[] nums, int target) {

if(start>=nums.length) return target==0;

int from=start;

int minus=0;

boolean done=false;

for(int i=start; i<nums.length-1 && !done;){

if(nums[i]==nums[i+1]){

from=i;

minus=minus +nums[i];

i++;

done=false;

} else{done=true;}

}

if( from!=start) {

if(groupSumClump(from,nums,target-minus)) return true;

if(groupSumClump(from,nums,target)) return true;

return false;

}

if(groupSumClump(start+1,nums,target-nums[start]) ) return true;

if( groupSumClump(start+1,nums,target)) return true;

return false;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public boolean splitArray(int[] nums) {

boolean b=false;

if(nums.length==1){

b=false;

}else if(nums.length>1){

if(helper(nums,0)==0){b=true; return b;}

b=false;

}else{

b=true;

}

return b;}

public int helper (int[] nums,int target,int ind){

if(ind<nums.length){

if( helper(nums,target-nums[ind],ind+1) ==0) return 0;

if( helper(nums,target,ind+1)==0) return 0;

if(helper(nums,target+nums[ind],ind+1) ==0)return 0;

}

return target==0?0:1;

}

public int helper(int[] nums,int index){

if(index<nums.length){

if(helper(nums,nums[index],0)==0)return 0;

if(helper(nums,index+1)==0)return 0;

}

return 1;

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\