**CODING BAT SOLVED PROBLEMS**

**Warmup-1**

**1.sleepIN**

The parameter weekday is true if it is a weekday, and the parameter vacation is true if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return true if we sleep in.

**public boolean sleepIn(boolean weekday, boolean vacation) {**

**if(!weekday||vacation){**

**return true;**

**}**

**return false;**

**}**

**2.diff21**

Given an int n, return the absolute difference between n and 21, except return double the absolute difference if n is over 21.

**public int diff21(int n) {**

**if(n<21){**

**return 21-n;**

**}else{**

**return (n-21)\*2;**

**}**

**}**

**3.hasteen**

We'll say that a number is "teen" if it is in the range 13..19 inclusive. Given 3 int values, return true if 1 or more of them are teen.

**public boolean hasTeen(int a, int b, int c) {**

**return (a>=13 && a<=19) || (b>=13 && b<=19) || (c>=13 && c<=19);**

**}**

**4. monkeyTrouble**

We have two monkeys, a and b, and the parameters aSmile and bSmile indicate if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return true if we are in trouble.

**public boolean monkeyTrouble(boolean aSmile, boolean bSmile) {**

**if((aSmile==true && bSmile==true)||(aSmile==false && bSmile==false)){**

**return true;**

**}else{**

**return false;**

**}**

**}**

**5. parrotTrouble**

We have a loud talking parrot. The "hour" parameter is the current hour time in the range 0..23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Return true if we are in trouble.

**public boolean parrotTrouble(boolean talking, int hour) {**

**return(hour<7||hour>20)&&(talking == true);**

**}**

**6.or35**

Return true if the given non-negative number is a multiple of 3 or a multiple of 5. Use the % "mod" operator -- see [Introduction to Mod](https://codingbat.com/doc/practice/mod-introduction.html)

**public boolean or35(int n) {**

**return(n%3==0||n%5==0);**

**}**

**7.icyHot**

Given two temperatures, return true if one is less than 0 and the other is greater than 100.

**public boolean icyHot(int temp1, int temp2) {**

**return((temp1<0&&temp2>100)||(temp2<0&&temp1>100));**

**}**

**8.in3050**

Given 2 int values, return true if they are both in the range 30..40 inclusive, or they are both in the range 40..50 inclusive.

**public boolean in3050(int a, int b) {**

**if(((a>=30 && a<=40))&&((b>=30 && b<=40))||((a>=40&&a<=50)&&(b>=40&&b<=50))){**

**return true;**

**}else{**

**return false;**

**}**

**}**

**9.lastDigit**

Given two non-negative int values, return true if they have the same last digit, such as with 27 and 57. Note that the % "mod" operator computes remainders, so 17 % 10 is 7.

**public boolean lastDigit(int a, int b) {**

**int d,e;**

**d=a%10;**

**e=b%10;**

**if(d==e){**

**return true;**

**}else{**

**return false;**

**}**

**}**

**10.sumDouble**

Given two int values, return their sum. Unless the two values are the same, then return double their sum.

**public int sumDouble(int a, int b) {**

**if(a==b){**

**return (a+b)\*2;**

**}else{**

**return a+b;**

**}**

**}**

**11. makes10**

Given 2 ints, a and b, return true if one if them is 10 or if their sum is 10.

public boolean makes10(int a, int b) {

return(a==10||b==10||a+b==10);

}

**12.in1020**

Given 2 int values, return true if either of them is in the range 10..20 inclusive.

**public boolean in1020(int a, int b) {**

**return((a>=10&&a<=20)||(b>=10&&b<=20));**

**}**

**13.intMax**

Given three int values, a b c, return the largest.

**public int intMax(int a, int b, int c) {**

**if(a>b){**

**if(a>c){**

**return a;**

**}else{**

**return c;**

**}**

**}else if (b>c){**

**return b;**

**}else{**

**return c;**

**}**

**}**

**String-1**

**14.helloName**

Given a string name, e.g. "Bob", return a greeting of the form "Hello Bob!".

**public String helloName(String name) {**

**return "Hello "+name + "!";**

**}**

**15.makeAbba**

Given two strings, a and b, return the result of putting them together in the order abba, e.g. "Hi" and "Bye" returns "HiByeByeHi".

**public String makeAbba(String a, String b) {**

**return((a+b)+(b+a));**

**}**

**16.makeTags**

The web is built with HTML strings like "<i>Yay</i>" which draws Yay as italic text. In this example, the "i" tag makes <i> and </i> which surround the word "Yay". Given tag and word strings, create the HTML string with tags around the word, e.g. "<i>Yay</i>".

**public String makeTags(String tag, String word) {**

**return "<"+tag+">"+word+"</"+tag+">";**

**}**

**17.makeOutWord**

Given an "out" string length 4, such as "<<>>", and a word, return a new string where the word is in the middle of the out string, e.g. "<<word>>". Note: use str.substring(i, j) to extract the String starting at index i and going up to but not including index j.

**public String makeOutWord(String out, String word) {**

**String a= out.substring(0,2);**

**String b= out.substring(2,4);**

**return a+word+b;**

**}**

**18.firstTwo**

Given a string, return the string made of its first two chars, so the String "Hello" yields "He". If the string is shorter than length 2, return whatever there is, so "X" yields "X", and the empty string "" yields the empty string "". Note that str.length() returns the length of a string.

**public String firstTwo(String str) {**

**if(str.length()<2){**

**return str;**

**}else{**

**String a = str.substring(0,2);**

**return a;**

**}**

**}**

**19.firstHalf**

Given a string of even length, return the first half. So the string "WooHoo" yields "Woo".

**public String firstHalf(String str) {**

**String a= str.substring(0,str.length()/2);**

**return a;**

**}**

**20.withoutEnd**

Given a string, return a version without the first and last char, so "Hello" yields "ell". The string length will be at least 2.

**public String withoutEnd(String str) {**

**return str.substring(1,str.length()-1);**

**}**

**21.nonStart**

Given 2 strings, return their concatenation, except omit the first char of each. The strings will be at least length 1.

**public String nonStart(String a, String b) {**

**String c= a.substring(1,a.length());**

**String d= b.substring(1,b.length());**

**return c+d;**

**}**

**22.comboString**

Given 2 strings, a and b, return a string of the form short+long+short, with the shorter string on the outside and the longer string on the inside. The strings will not be the same length, but they may be empty (length 0).

**public String comboString(String a, String b) {**

**if(a.length()>b.length()){**

**return b+a+b;**

**}else{**

**return a+b+a;**

**}**

**}**

**23.left2**

Given a string, return a "rotated left 2" version where the first 2 chars are moved to the end. The string length will be at least 2.

**public String left2(String str) {**

**String a = str.substring(2,str.length());**

**String b = str.substring(0,2);**

**return a+b;**

**}**

**24.right2**

Given a string, return a "rotated right 2" version where the last 2 chars are moved to the start. The string length will be at least 2.

**public String right2(String str) {**

**String a = str.substring(str.length()-2);**

**String b = str.substring(0,str.length()-2);**

**return a+b;**

**}**

**25.theEnd**

Given a string, return a string length 1 from its front, unless **front** is false, in which case return a string length 1 from its back. The string will be non-empty.

**public String theEnd(String str, boolean front) {**

**String a,b;**

**if(front==true){**

**return a=str.substring(0,1);**

**}else{**

**return b=str.substring(str.length()-1);**

**}**

**}**

**26.withouEnd2**

Given a string, return a version without both the first and last char of the string. The string may be any length, including 0.

**public String withouEnd2(String str) {**

**String a;**

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

**return "";**

**}else{**

**return a= str.substring(1,str.length()-1);**

**}**

**}**

**27.middleTwo**

Given a string of even length, return a string made of the middle two chars, so the string "string" yields "ri". The string length will be at least 2.

**public String middleTwo(String str) {**

**int mid = str.length()/2;**

**return str.substring(mid-1,mid+1);**

**}**

**28.endsLy**

Given a string, return true if it ends in "ly".

**public boolean endsLy(String str) {**

**return str.endsWith("ly");**

**}**

**29.nTwice**

Given a string and an int n, return a string made of the first and last n chars from the string. The string length will be at least n.

**public String nTwice(String str, int n) {**

**String a= str.substring(0,n);**

**String b= str.substring(str.length()-n);**

**return a+b;**

**}**

**30.middleThree**

Given a string of odd length, return the string length 3 from its middle, so "Candy" yields "and". The string length will be at least 3.

**public String middleThree(String str) {**

**int v = (str.length()/2);**

**String a= str.substring(v-1,v);**

**String b= str.substring(v,v+2);**

**return a+b;**

**}**

**31.atFirst**

Given a string, return a string length 2 made of its first 2 chars. If the string length is less than 2, use '@' for the missing chars.

**public String atFirst(String str) {**

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

**String a =str.substring(0,2);**

**return a;**

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

**return str + "@";**

**}else{**

**return str + "@@";**

**}**

**}**

**Logic-1**

**32.cigarParty**

When squirrels get together for a party, they like to have cigars. A squirrel party is successful when the number of cigars is between 40 and 60, inclusive. Unless it is the weekend, in which case there is no upper bound on the number of cigars. Return true if the party with the given values is successful, or false otherwise.

**public boolean cigarParty(int cigars, boolean isWeekend) {**

**if (isWeekend==true){**

**return cigars>=40;**

**}else{**

**return(cigars>=40&&cigars<=60);**

**}**

**}**

**33.dateFashion**

You and your date are trying to get a table at a restaurant. The parameter "you" is the stylishness of your clothes, in the range 0..10, and "date" is the stylishness of your date's clothes. The result getting the table is encoded as an int value with 0=no, 1=maybe, 2=yes. If either of you is very stylish, 8 or more, then the result is 2 (yes). With the exception that if either of you has style of 2 or less, then the result is 0 (no). Otherwise the result is 1 (maybe).

**public int dateFashion(int you, int date) {**

**if(you<=2||date<=2){**

**return 0;**

**}else if(you>=8||date>=8){**

**return 2;**

**}else{**

**return 1;**

**}**

**}**