



Testing

Black Box and White Box

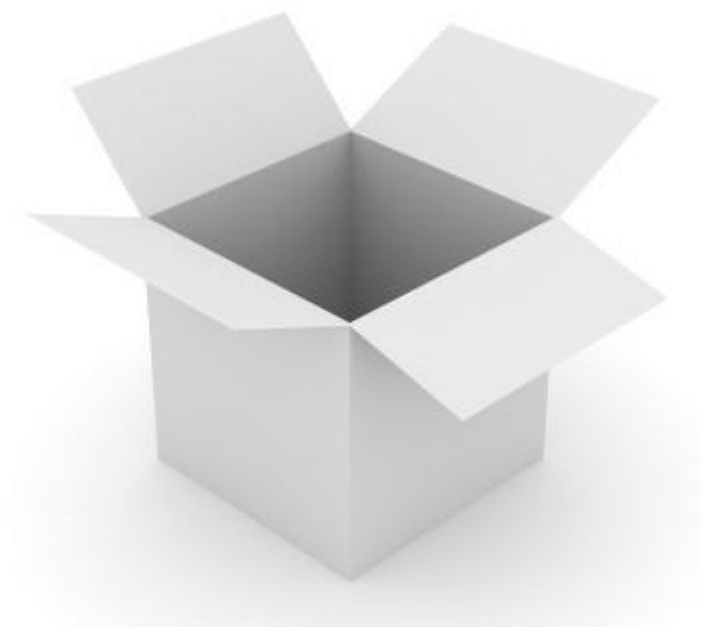




Goal: to find bugs (errors)

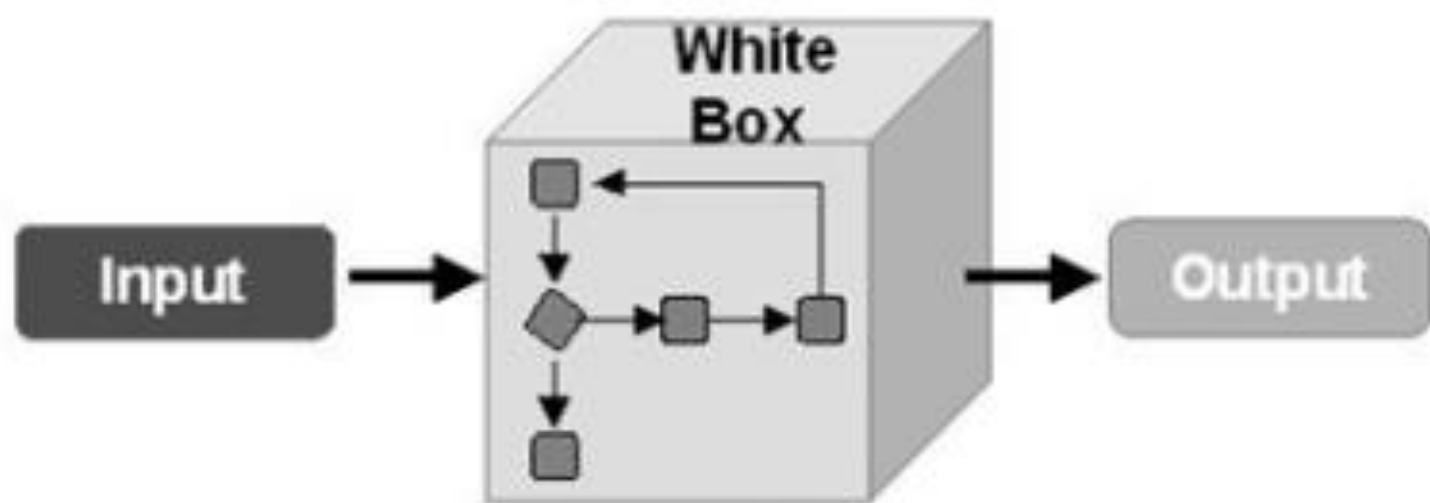
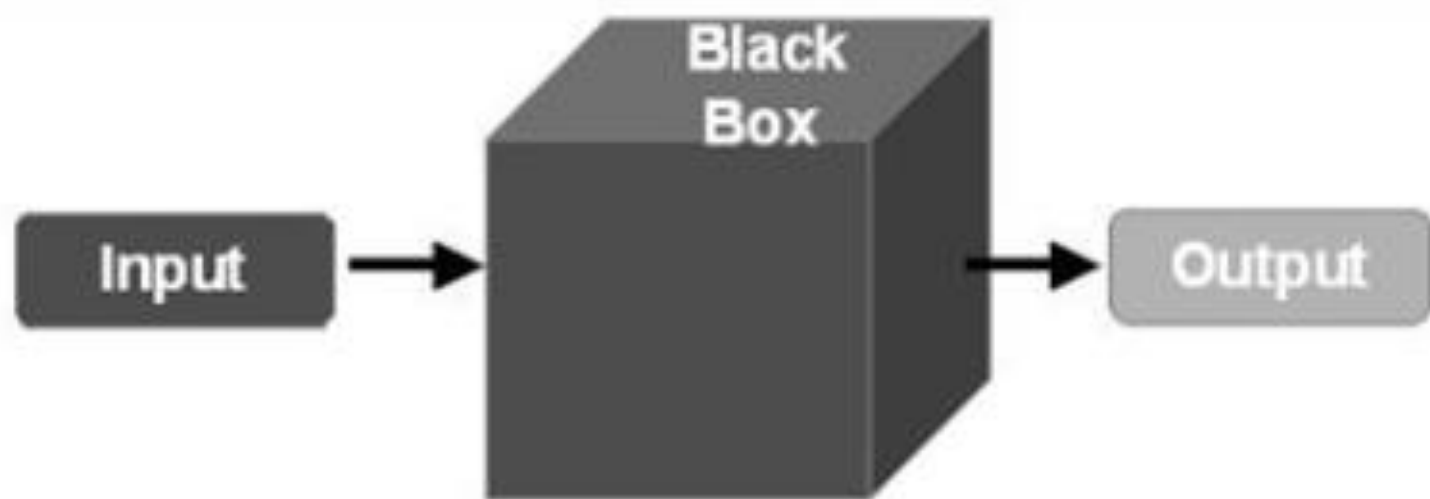
The person who codes it
should not be the person
who tests it.

Too much vested interest.



VS.






Black Box Testing

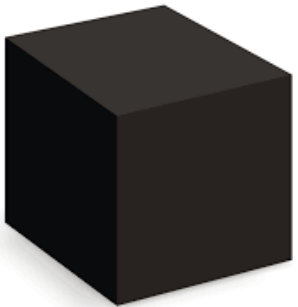
- Testing without looking at the code.
- Only know what is required and what should be produced
- Four kinds of data are tested:
 - (1) Small data
 - (2) Large data
 - (3) Average (normal) data
 - (4) Boundary cases – strange data, odd cases that aren't normally considered

White Box Testing

- Testing that occurs after looking at the code
- Goal is to run every line of code at least once.
- Two major considerations:
 - (1) Ifs
 - Test every clause of the if
 - (2) Loops
 - Avoid the loop
 - Run the loop once
 - Run the loop many times.



Black Box
Test This
App

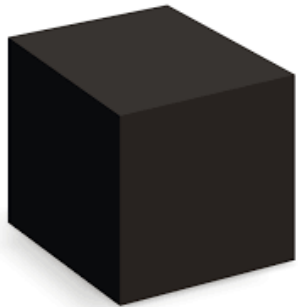


Log in!

First Name:

Alan

Log in



Short name?

Long name?

Average
name?

Boundary-
case name?

Log in!

First Name:

Alan

Log in

lo

Short name?

Wolfeschlegelsteinh
ausenbergerdorff

Long name?

James, Juan, Ramesh,
Wang Xiu Ying

Average
name?

LeeAnn,
Mary-Jo,
La-a

Boundary-
case name?

Log in!

First Name:

Alan

Log in

Your last name contains invalid characters

My last name is "Graham-Cumming". But here's a typical form response when

In 2010, John Graham-Cumming wrote a blog post complaining about sloppy testing.

Ektron Account Registration

In order to access the forum and other areas of the Ektron DevCenter, you need to create an Ektron Account. This account is for you and we will not share your information with any other organization. If you already have an account, use the login form in the right corner of this page to login.

Please fill out the required information below to start using the Ektron Exchange!

Display Name:

First Name:

Last Name:

contains invalid characters

Email Address:

Password:

Re-Enter Password:

Timezone:

Last Name

Name:

Your Last Name may not contain characters such as @, !, * or \$. Please reenter your Last Name using only letters and numbers now.

Falsehoods Programmers Believe About Names

June 17, 2010 in [Uncategorized](#)

[This post has been translated into Japanese for one of our readers: 和訳もあります。]

John Graham-Cumming wrote an [article](#) today complaining about the system he was working with described his last name as having invalid characters. It of course does not, because anything someone tells you is their name is — by definition — an appropriate identifier for them. John was understandably vexed about this situation, and he has every right to be, because **names are central to our identities, virtually by definition.**

I have lived in Japan for several years, programming in a professional capacity, and I have broken many systems by the simple expedient of being introduced into them. (Most people call me Patrick McKenzie, but I'll acknowledge as correct any of six different “full” names, any many systems I deal with will accept precisely

WHO AM I?

My name is Patrick McKenzie (better known as patio11 on the Internets.)

Twitter: [@patio11](#) HN: [patio11](#)

In response, Patrick McKenzie wrote a famous (in certain circles) blog post about boundary cases.

1. People have exactly one canonical full name.

It seems some people believe that you get a name and it never changes. Not so, even in Western countries, where a person may change their name when they marry. In Catholic tradition a person may get a middle name at time of confirmation.

2. People have exactly one full name which they go by.

The author known most often as John Wyndham (author of *The Day of the Triffids*) bore the name John Wyndham Parkes Lucas Beynon Harris, and published books under the names John Beynon and Lucas Parkes, as well as John Wyndham.

3. People have, at this point in time, exactly one canonical full name.

A performer may have a stage name, completely separate from the name on their birth and marriage certificates – they may even have a passport in their stage name.

4. People have, at this point in time, one full name which they go by.

Not so, even in Western countries, where a woman may choose to retain her unmarried name at work (where she is already known by that name), and use her husband's surname on social occasions, and even on legal documents such as mortgages and loans.

21. People's names are globally unique.

Tell that to anyone named John Smith! I have a somewhat less common name, yet I discovered a person with the same name working in the same industry in the same country (Australia).

22. People's names are *almost* globally unique.

Even with the tendency to use unusual spellings of names, it's extremely common to find people who share a full name – try Googling your own name.

23. Alright alright but surely people's names are diverse enough such that no million people share the same name.

The Chinese name Zhang Wei is reported to be shared by over a quarter of a million people.

If we limit the question to surnames, about 20% of the population of South Korea have the surname Kim. About 10% of the population of northern China share the surname Wang, while more than 10% of the population of southern China share the surname Chen. Li comes next in both northern and southern China, making it the most common surname across the country. And nearly 40% of Vietnamese have the surname Nguyen.

Names are far from unique.

39. **People whose names break my system are weird outliers. They should have had solid, acceptable names, like 田中太郎.**

No, your system is badly designed.

This particular example name is perhaps best known as the name of an alien in an anime series (and a manga). There have also been real people with this name.

40. **People have names.**

This one is perhaps the most difficult for which to give solid examples. There was an isolated culture in which no one had names – they referred to everyone in relative terms, such as “my mother’s eldest sister”.

Days in Month

Month:

Feb

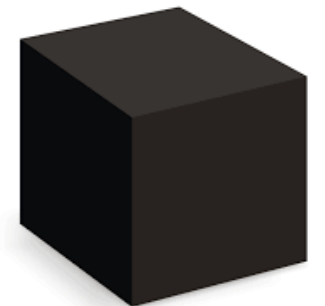
Year:

2020

Enter

29 Days

Black Box
Test This
App



Days in Month

Month:

Feb

Year:

2020

Enter

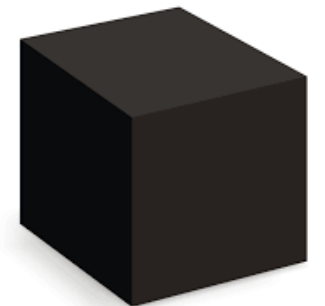
29 Days

Short
month?

Long month?

Average
month?

Boundary-
case month?



Days in Month

Month:

Feb

Year:

2020

Enter

29 Days

Short
month?

Feb 2019

Long month?

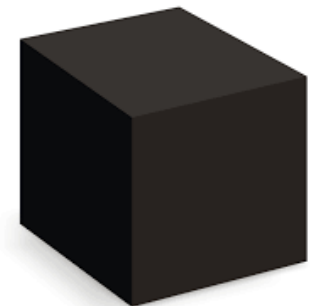
Dec 2019

Average
month?

Aug 2020

Boundary-
case month?

Feb 2020
(leap day)





SUN JUN 17TH

Falsehoods programmers believe about time

Over the past couple of years [I have spent a lot of time](#) debugging other engineers' test code. This was interesting work, occasionally frustrating but always informative. One might not immediately think that test code would have bugs, but of course *all* code has bugs and tests are no exception.

I have repeatedly been confounded to discover just how many mistakes in *both* test *and* application code stem from misunderstandings or misconceptions about *time*. By this I mean both the interesting way in which computers handle time, and the fundamental gotchas inherent in how we humans have constructed our calendar – daylight savings being just the tip of the iceberg.

In fact I have seen so many of these misconceptions crop up in other people's (and my own) programs that I thought it would be worthwhile to collect a list of the more common problems here.

All of these assumptions are wrong

These are **all** false.

1. There are always 24 hours in a day.

2. Years have 365 days.

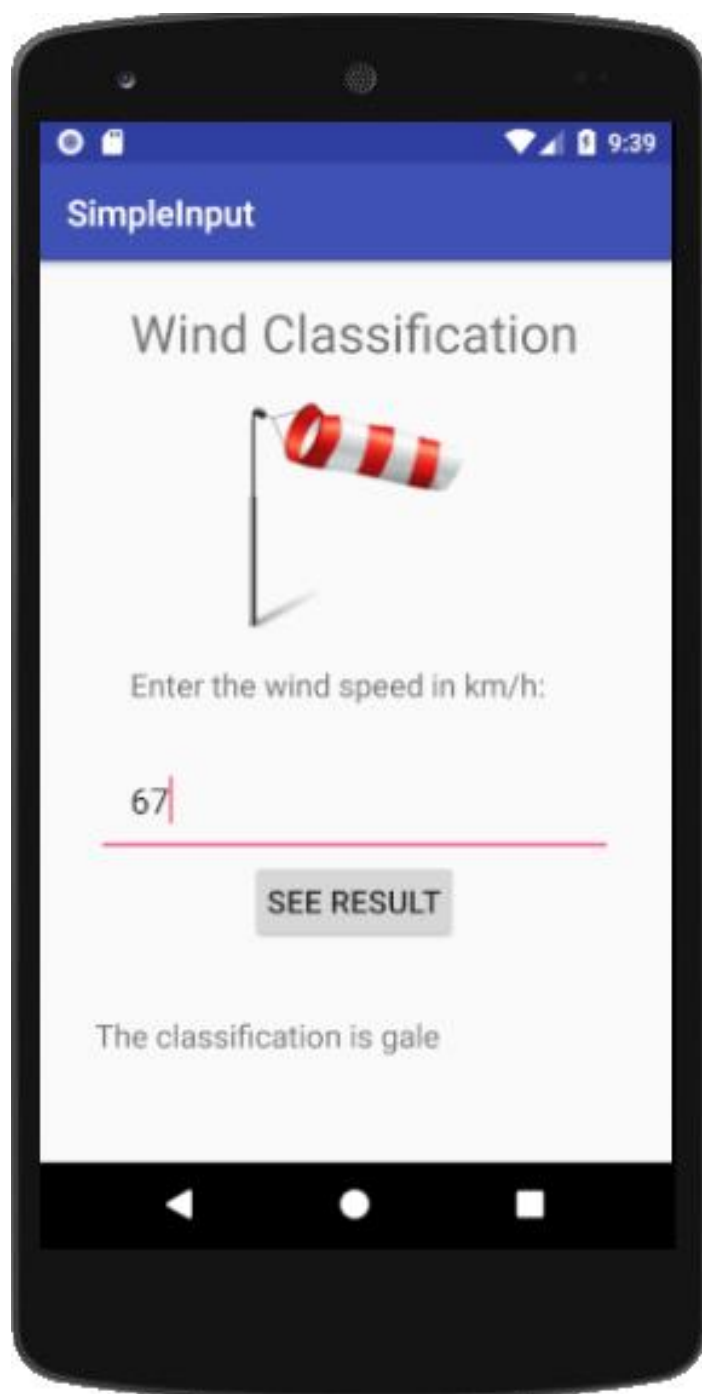
3. Months have either 30 or 31 days.

4. Oh – except February. It is always 28 days long.

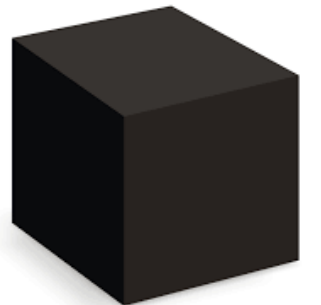
5. Ok, fine. Months have either 28, 29, 30, or 31 days.

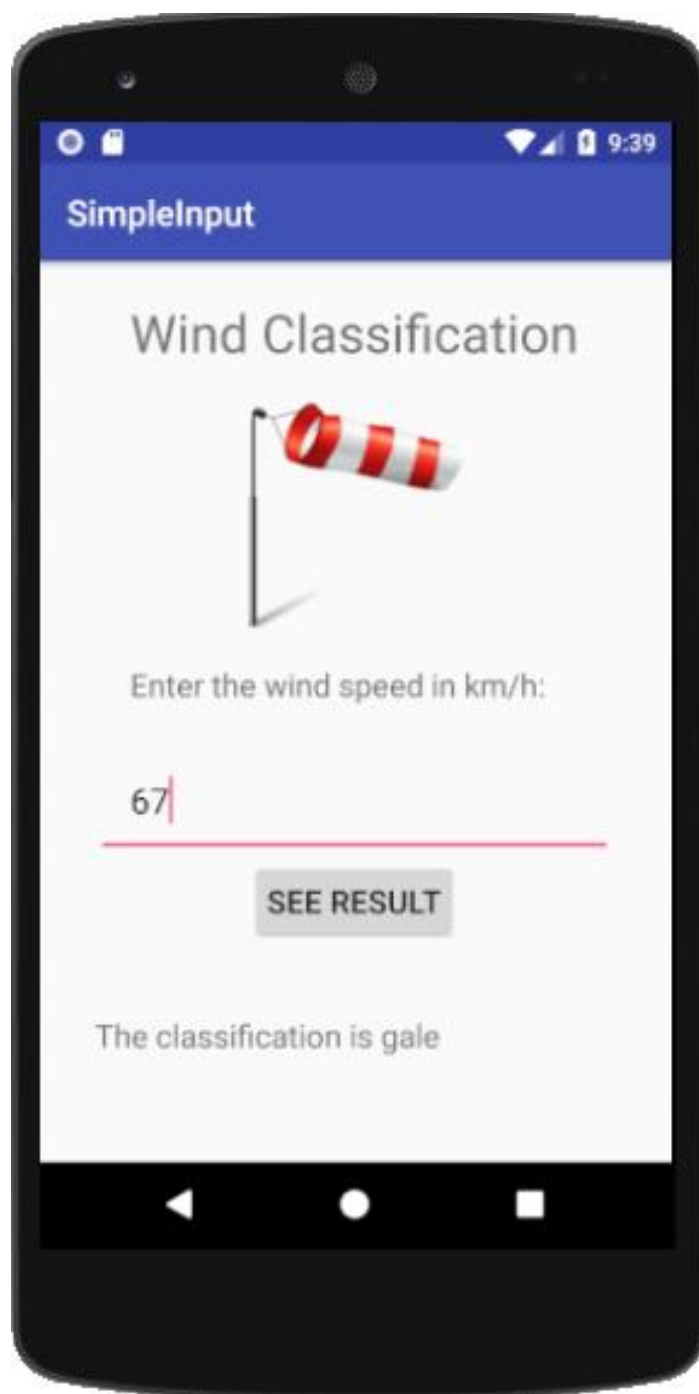
6. There is a leap year every year divisible by 4.

7. Non leap years will never contain a leap day.



Black Box
Test This
App



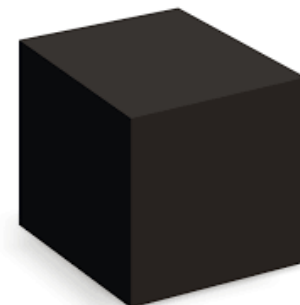


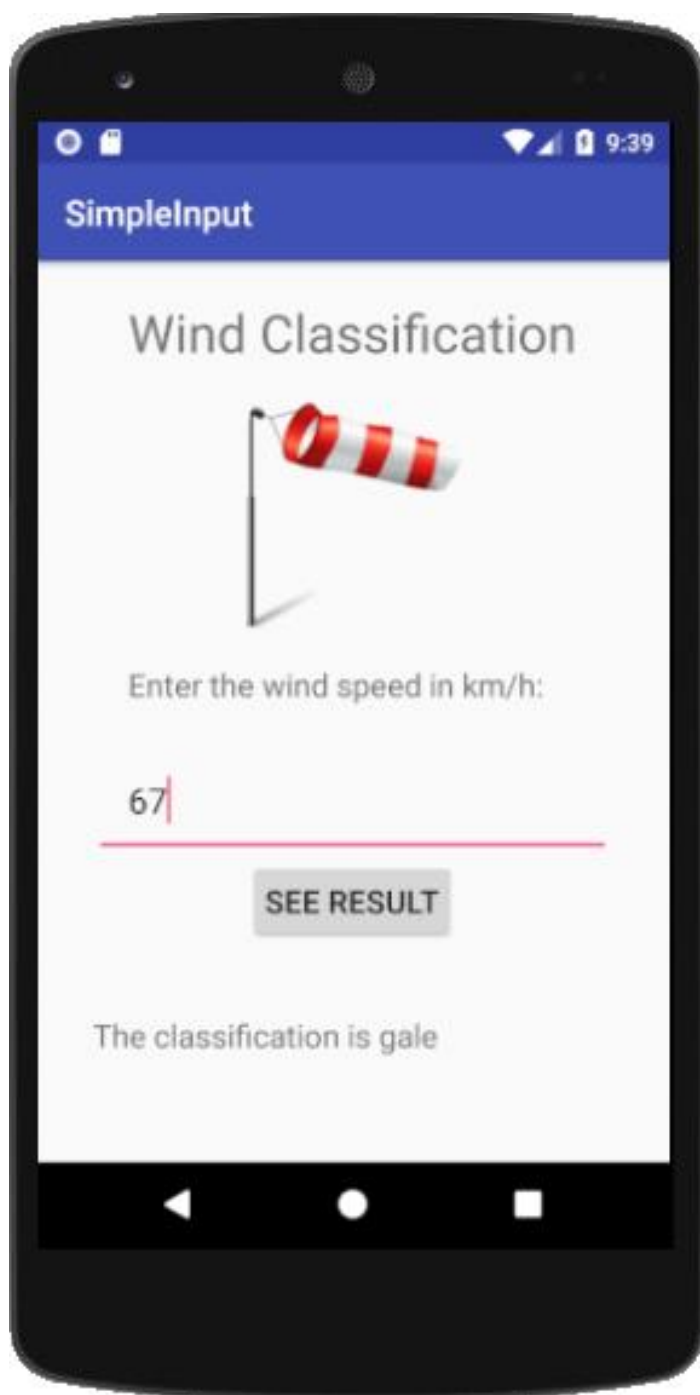
Short
number?

Long
number?

Average
number?

Boundary-
case
number?





Short
number?

23

Long
number?

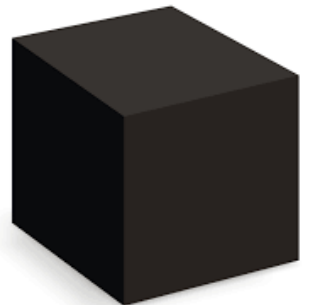
1,234,456,789

Average
number?

134

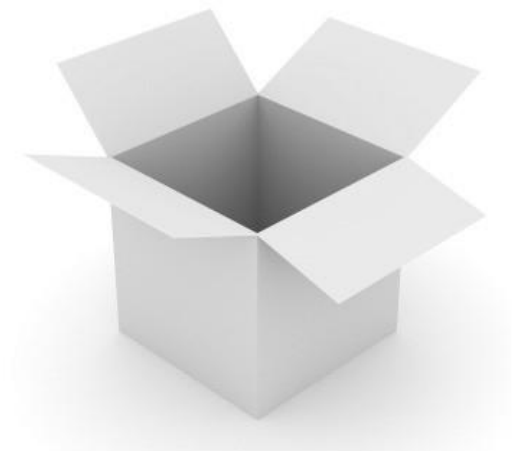
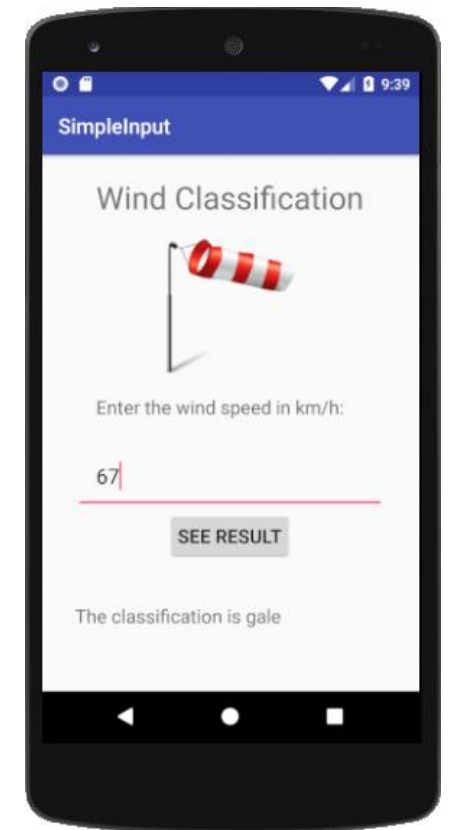
Boundary-
case
number?

-1, 0, 1



```
public void result(View view) {  
    TextView output = (TextView) findViewById(R.id.output);  
    EditText input = (EditText) findViewById(R.id.input);  
    int num = Integer.parseInt(input.getText().toString());  
    if (num < 5)  
        output.setText("The classification is calm");  
    else if (num < 11)  
        output.setText("The classification is light");  
    else if (num < 19)  
        output.setText("The classification is gentle");  
    else if (num < 28)  
        output.setText("The classification is moderate");  
    else if (num < 49)  
        output.setText("The classification is strong");  
    else if (num < 74)  
        output.setText("The classification is gale");  
    else if (num < 117)  
        output.setText("The classification is storm");  
    else  
        output.setText("The classification is hurricane");  
}
```

White
Box Test
This App



```

public void result(View view) {

    TextView output = (TextView) findViewById(R.id.output);
    EditText input = (EditText) findViewById(R.id.input);

    int num = Integer.parseInt(input.getText().toString());

    if (num < 5)
        output.setText("The classification is calm");
    else if (num < 11)
        output.setText("The classification is light");
    else if (num < 19)
        output.setText("The classification is gentle");
    else if (num < 28)
        output.setText("The classification is moderate");
    else if (num < 49)
        output.setText("The classification is strong");
    else if (num < 74)
        output.setText("The classification is gale");
    else if (num < 117)
        output.setText("The classification is storm");
    else
        output.setText("The classification is hurricane");
}

```

Under 5

5-10

11-18

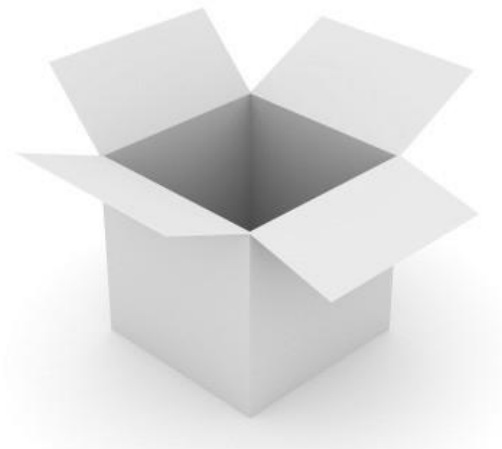
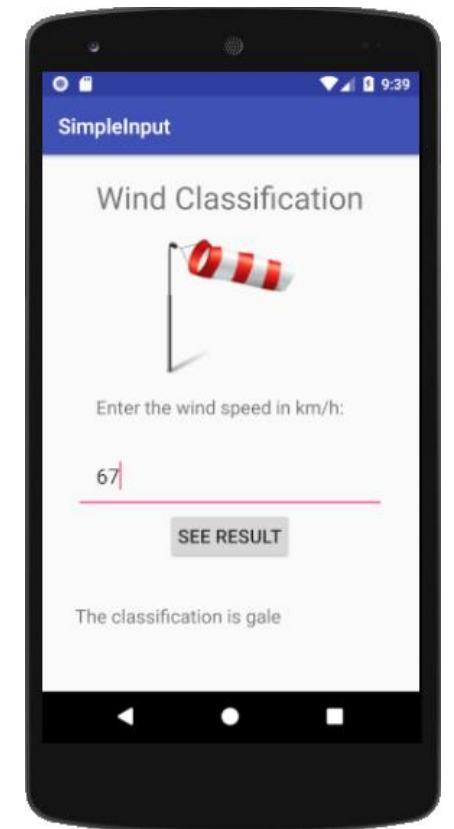
19-27

28-48

49-73

74-116

Over
117




```

public void result(View view) {

    TextView output = (TextView) findViewById(R.id.output);

    EditText input = (EditText) findViewById(R.id.input);

    int num = Integer.parseInt(input.getText().toString());

    if (num < 5)
        output.setText("The classification is calm");
    else if (num < 11)
        output.setText("The classification is light");
    else if (num < 19)
        output.setText("The classification is gentle");
    else if (num < 28)
        output.setText("The classification is moderate");
    else if (num < 49)
        output.setText("The classification is strong");
    else if (num < 74)
        output.setText("The classification is gale");
    else if (num < 117)
        output.setText("The classification is storm");
    else
        output.setText("The classification is hurricane");

}

```

Under 5

3

5-10

6

11-18

13

19-27

23

28-48

34

49-73

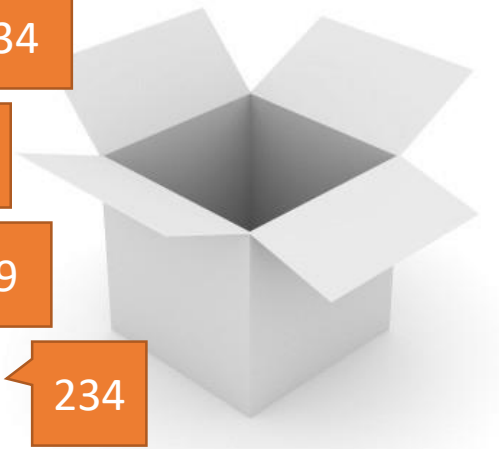
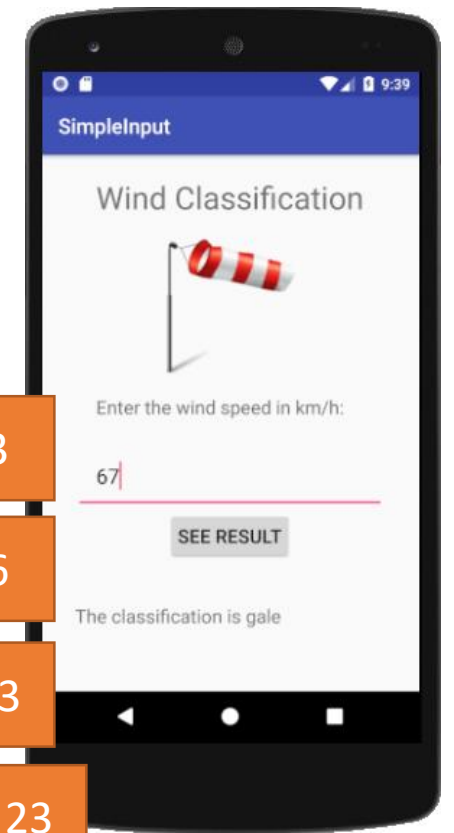
56

74-116

89

Over
117

234



A vote is held after singer A and singer B compete in the final round of a singing competition. Your job is to count the votes and determine the outcome.

The first line of input will be a sequence of characters, each of which will be A or B, representing the votes for a particular singer.

The output will be one of three possibilities:

- “A wins the contest”, if there are more A votes than B votes;
- “B wins the contest”, if there are more B votes than A votes;
- “There is a tie”, if there are an equal number of A votes and B votes.



Sample 1

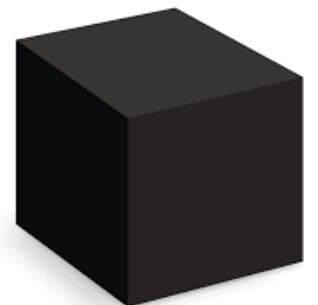
Votes: **ABBABB**

B wins the contest

Sample 2

Votes: **ABBABA**

There is a tie.



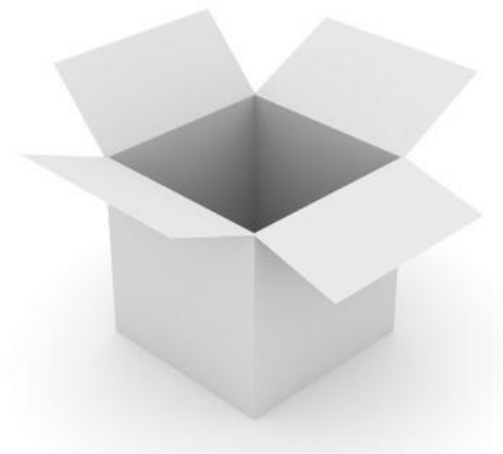
```
Scanner in = new Scanner(System.in);
System.out.print("Enter votes: ");
String vote = in.nextLine();
int a = 0;
int b = 0;

for (int i=0; i<vote.length(); i++) {
    if(vote.charAt(i)=='a')
        a++;
    else if(vote.charAt(i)=='b')
        b++;
}

if(a>b)
    System.out.println("A won.");
else if(a==b)
    System.out.println("Tie");
else
    System.out.println("B won.");
```

Enter votes: **abbabaa**
A won.

White Box
Test This
Code



```
Scanner in = new Scanner(System.in);
System.out.print("Enter votes: ");
String vote = in.nextLine();
int a = 0;
int b = 0;
```

Enter votes: **abbabaa**
A won.

```
for (int i=0; i<vote.length(); i++) {
    if(vote.charAt(i)=='a')
        a++;
    else if(vote.charAt(i)=='b')
        b++;
}
```

Avoid Loop

Loop once

Loop many
times

A in Votes

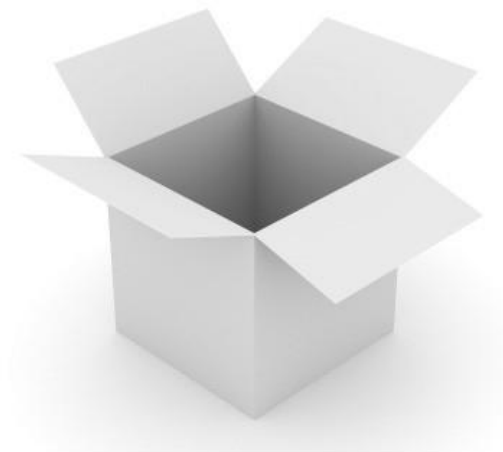
B in Votes

```
if(a>b)
    System.out.println("A won.");
else if(a==b)
    System.out.println("Tie");
else
    System.out.println("B won.");
```

More A than B

Same A and B

More B than A



```

Scanner in = new Scanner(System.in);
System.out.print("Enter votes: ");
String vote = in.nextLine();
int a = 0;
int b = 0;

```

Enter votes: **abbabaa**
A won.

```

for (int i=0; i<vote.length(); i++) {
    if(vote.charAt(i)=='a')
        a++;
    else if(vote.charAt(i)=='b')
        b++;
}

```

Avoid Loop

Empty String

Loop once

One Letter: A

Loop many times

Many letters:
ABBADAFDA

A in Votes

AAAB

B in Votes

BBABB

```

if(a>b)
    System.out.println("A won.");
else if(a==b)
    System.out.println("Tie");
else
    System.out.println("B won.");

```

More A than B

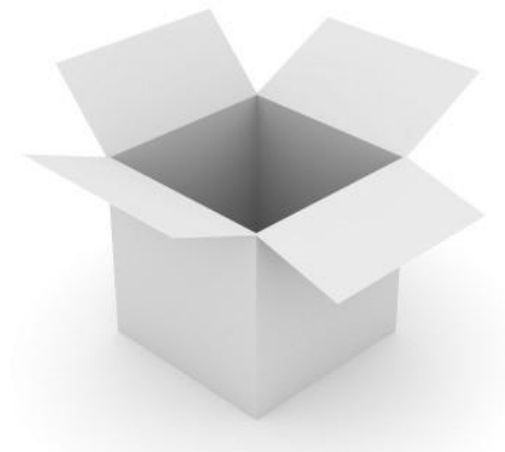
AAABA

Same A and B

AABBAABB

More B than A

BBABBA



Ariane 5 rocket

June 4, 1996

unmanned

a decade of development

cost \$7 billion

rocket and cargo valued at
\$500 million





Forty seconds after its lift-off



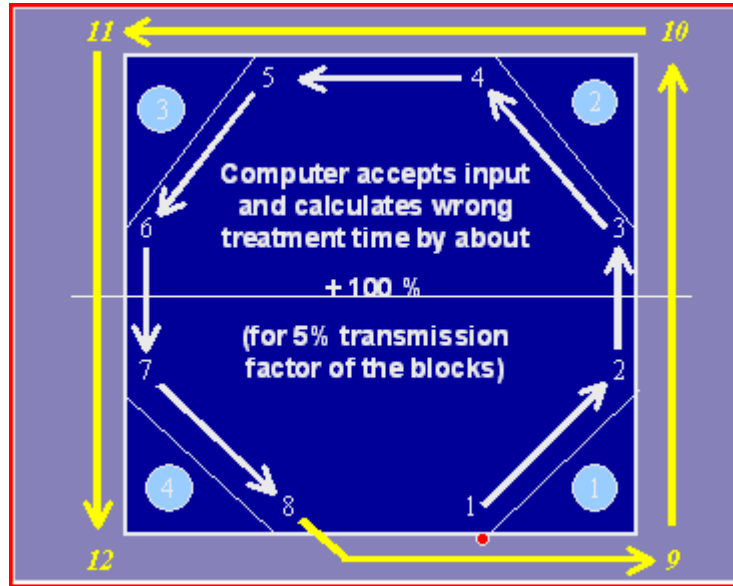
Didn't test the software

The software only handled small for the speed variable. They didn't test average or large data for the speed.

It overloaded memory and the rocket exploded.



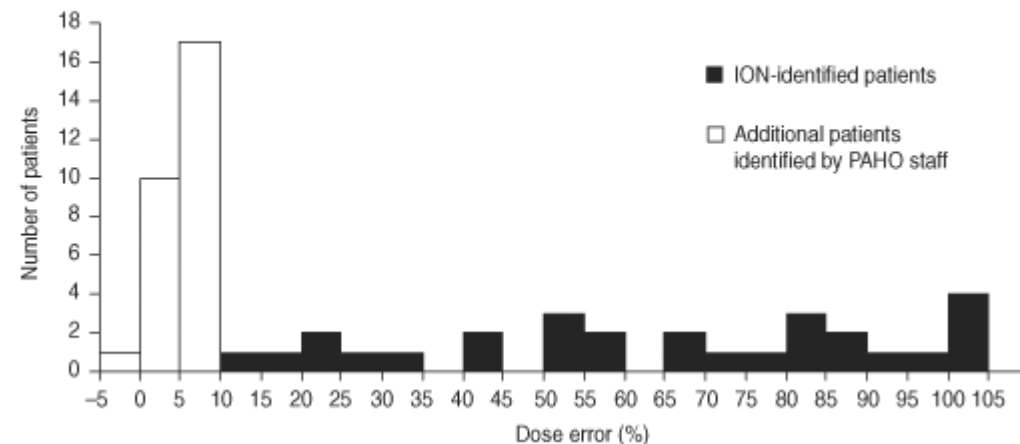
Panama City, August 2000
Cancer Treatment Center
Patient Treatment Planning System
Stopped lethal doses of radiation



While the TPS accepts data for multiple blocks, this was never tested.

It treats multiple doses of radiation as if only one dose was delivered. [Loop testing failed – no multiple loops tested]

Figure 2. Errors in absorbed doses of teletherapy treatments received by 56 patients (percentage error (%) = $[(\text{delivered dose} - \text{prescribed dose}) / \text{prescribed dose}] \times 100$), National Oncology Institute (Instituto Oncológico Nacional, ION) of Panama, 2000–2001



28 people receive over doses
17 die, 11 severely injured.
Error discovered March 2001.

Ariane 5 rocket

- Unmanned rocket with satellites
- A decade of development cost \$7 billion
- Rocket and cargo valued at \$500 million
- Launched June 4, 1996
- Exploded 40 sec after take-off
- Didn't Black Box test "Large Values" for the speed variable. It overloaded memory.

American Megatrends

Cancer Treatment

- Device delivered radiation doses to cancer patients
- Used in Panama City, August 2000 to March 2001
- Didn't white box test "Run Loop Multiple Times"
- It treated multiple doses of radiation as only one dose.
- 28 people receive over doses. 17 die, 11 severely injured.

More Examples

To Practice White and Black Box Testing

Find all:

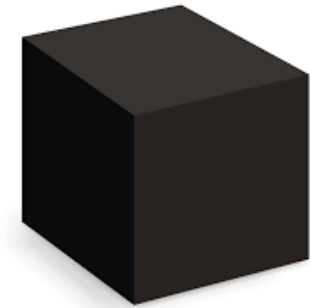
The code looks through the String to find all positions of the requested letter.

Sentence: `i like to eat apples and bananas`

To Find: `a`

`a` found in position:

`11, 14, 21, 26, 28, 30,`



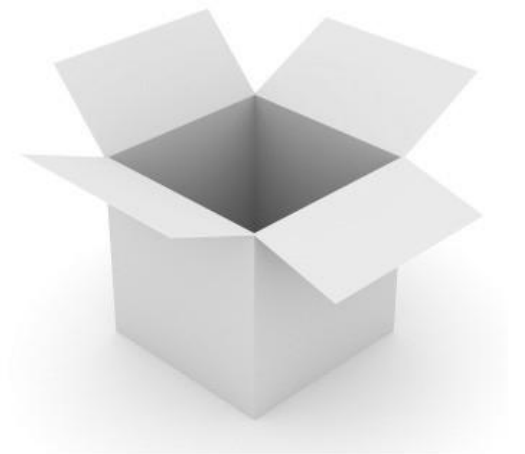
```
import java.util.*;
public class FindAll {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Sentence: ");
        String msg = in.nextLine();
        System.out.print("To Find: ");
        String find = in.nextLine();
        char f = find.charAt(0);

        System.out.println("\n"+f+" found in position: ");
        for(int i=0; i<msg.length(); i++) {
            if(msg.charAt(i)==f)
                System.out.print(i+", ");
        }
    }
}
```

Sentence: **hello hi**
To Find: **h**

h found in position:
0, 6,



Address

Number:

251

Street:

McMurchy

Street Type:

Ave

Enter

Address

1st Line Address:

251 McMurchy Ave S

Enter