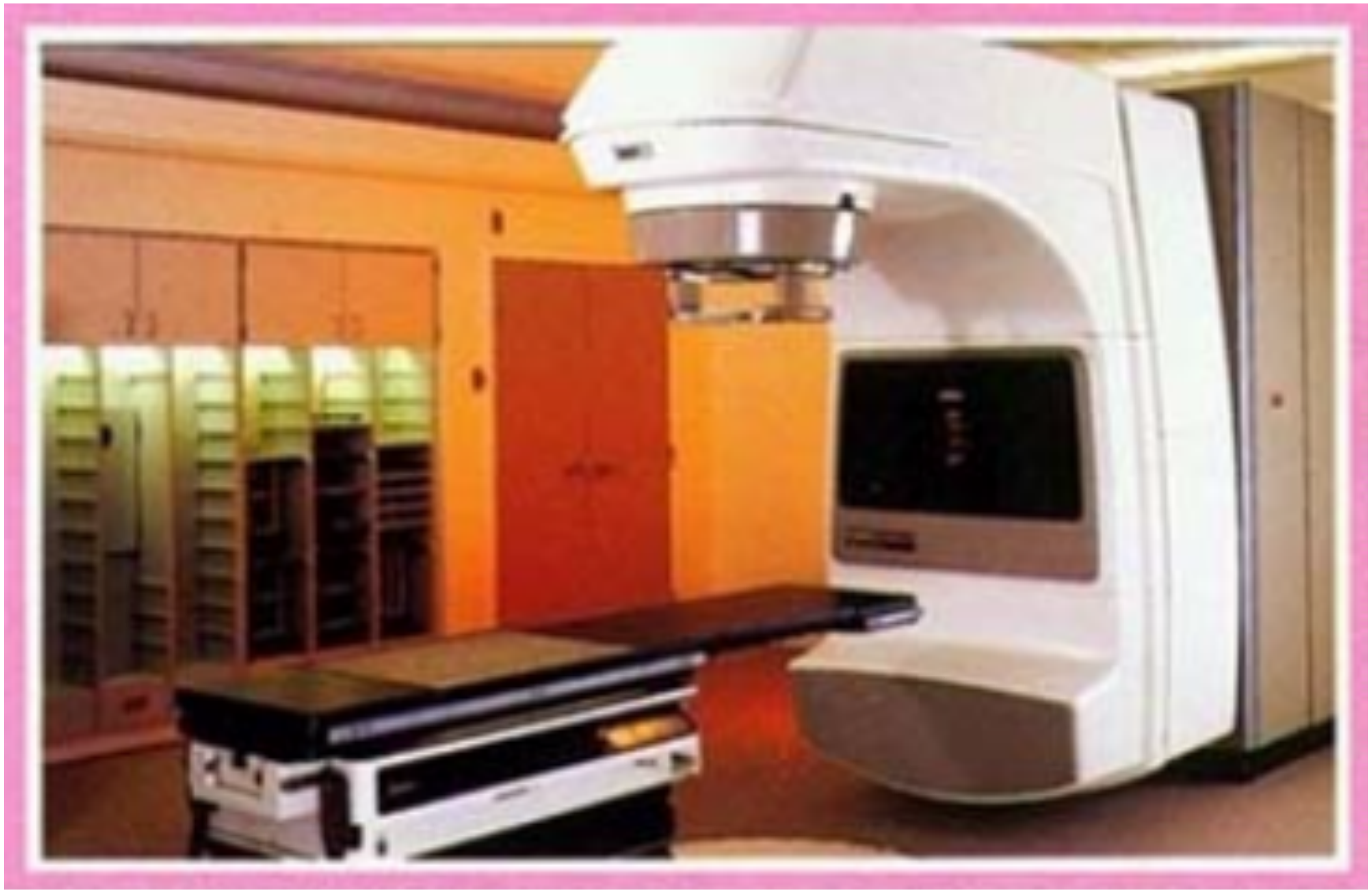
More Fun With Summations

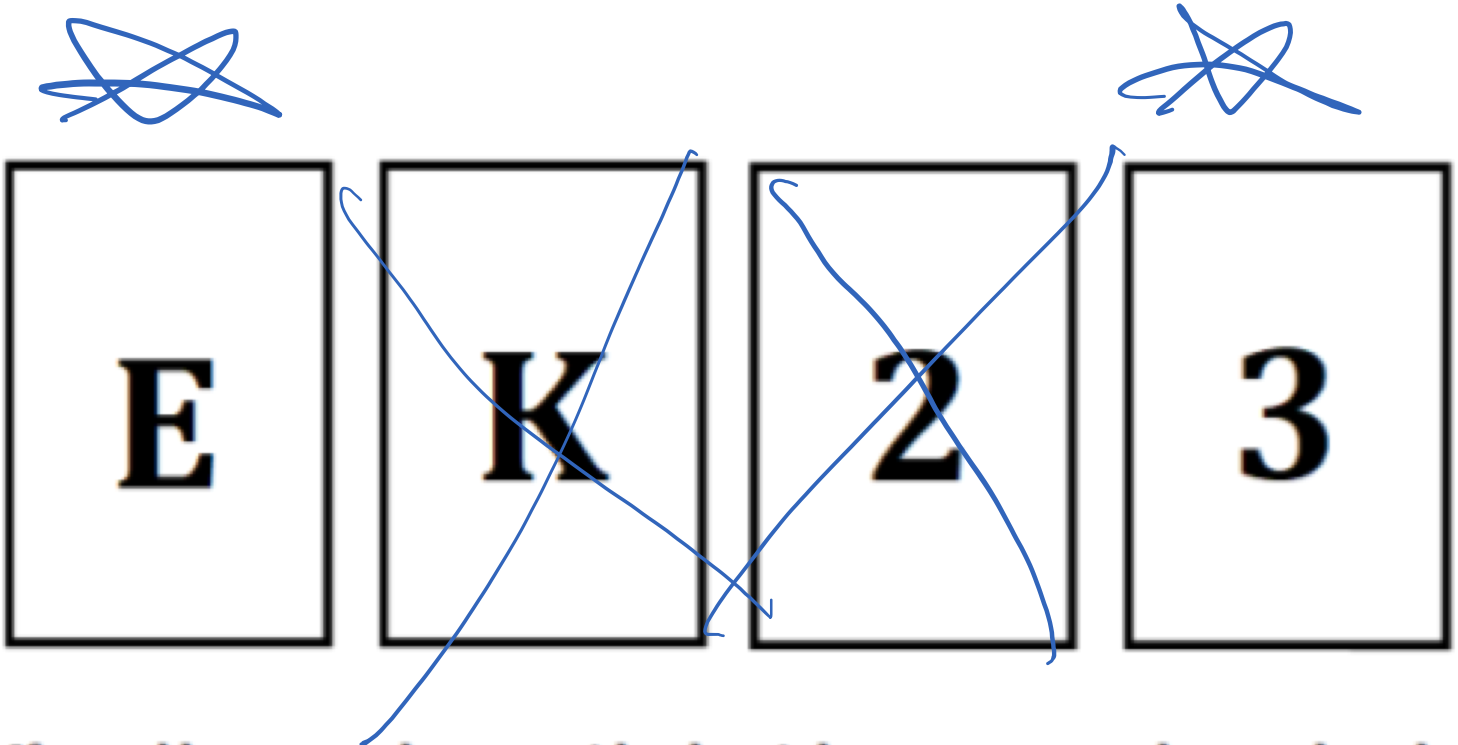
```
int main() {
    int data[N+1];

    for (int i=1; i < N; i *= 2) {
        for (int j=1; j < i; j += 2) {
            data[i] += j;
        }
    }

    return 0;
}
```

Evaluating code/algorithms: correctness



therac.png

Let’s Play A Card Game



Rule: If a card has a vowel on one side, then it has an even number on the other side

cards3.png

Which cards should we flip over to decide if the rule is true?

Testing: Philosophy of Science Point of View

``My proposal is based on an asymmetry between verifiability and falsifiability; an asymmetry which results from the logical form of universal statements. For these are never derivable from singular statements, but can be contradicted by singular statements.’’

– Karl Popper: The Logic of Scientific Discovery 1959

``Program testing can be used to show the presence of bugs, but never to show their absence!’’

– Edsger Dijkstra: Notes On Structured Programming

What is the purpose of testing?

3 input fields

First Name (20 chars)

Last Name (30 chars)

Phone # (10 digits)

⇒ Can't exhaustively test

⇒ Goal: Finding errors

Name: "; delete * from UserTable'

Testing in Homework 4

We give you …

tests for correctness

You give us …

tests for algorithms to grow train

Interface, Encoding, and Implementation

The *interface* of a class is…

The *encoding* of a class is…

The *implementation* of a class is…

Implementation - How we fullfill promises of interface

```
class Barn {
public:
    Barn();
    Barn(const Barn& otherBarn);
    ~Barn();

    void visit();
    void addCow(const string& cowName);
    bool hasCow(const string& cowName);
    static const size_t MAX_COWS = 10;
private:
    Cow cows_[MAX_COWS];
}
```

interface

encoding

(private member functions)