L#4. Problem Solving Tools and Program Design in Computer Programming: The Selection Structure

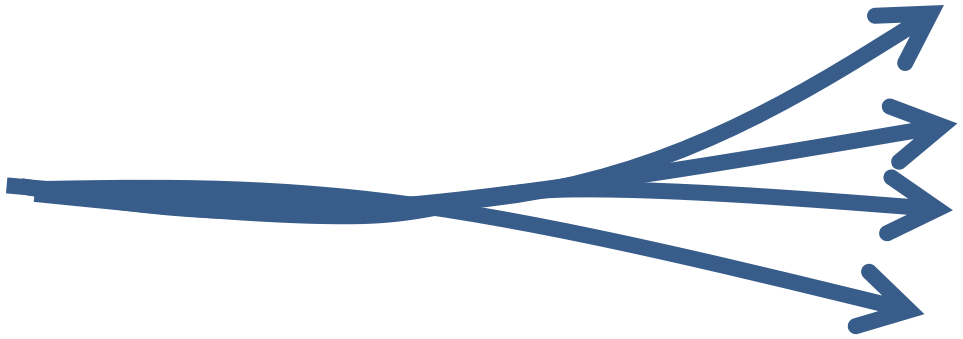
Problems with Solutions Requiring Selection

August 19, 2019

Background

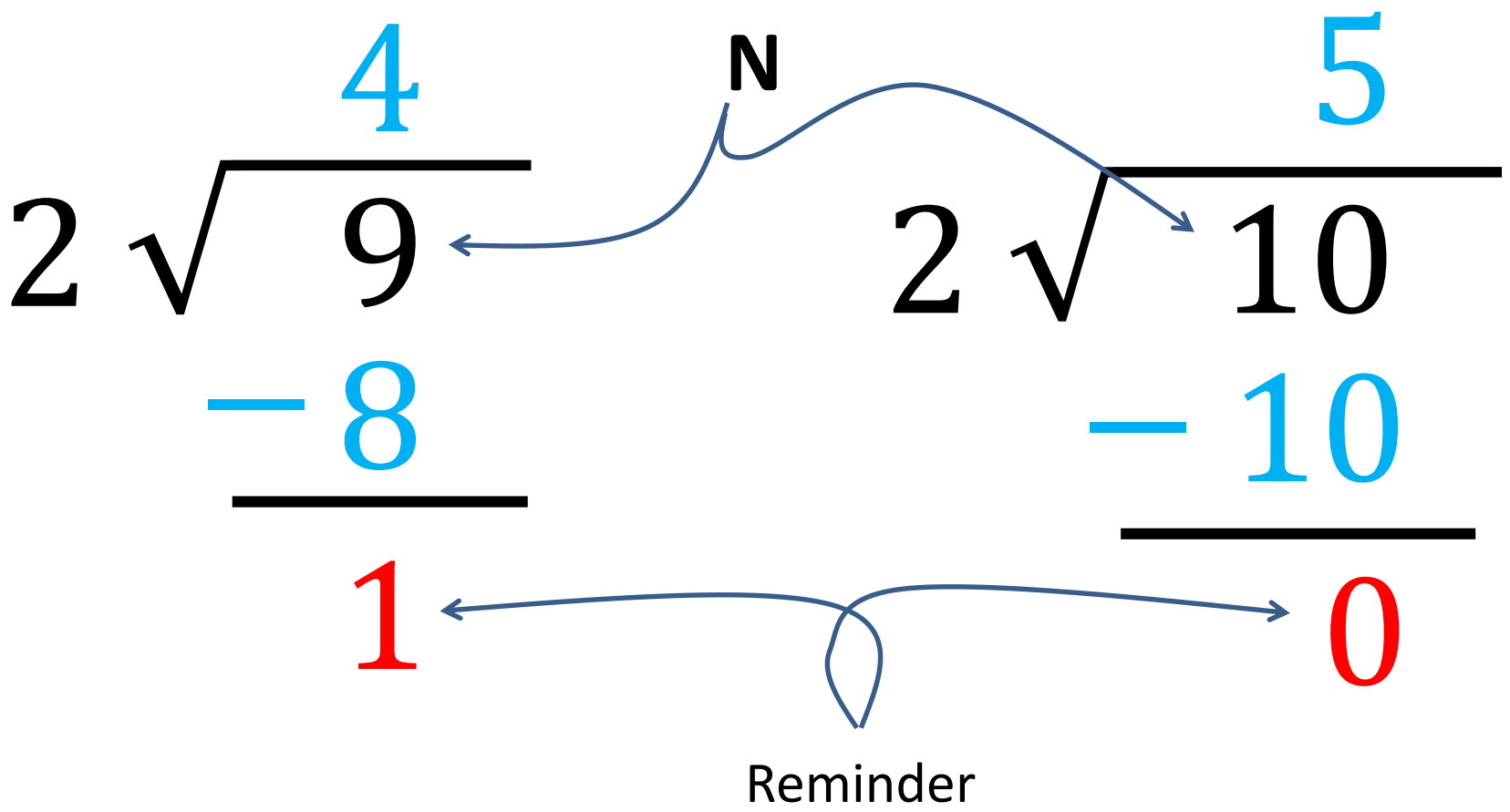
- Up to this point, we have solved problems with solutions which were from the start to the end, each line in pseudocode (or shape in the flowchart) was executed once, in order, sequentially.

- What happens if the problem requires a solution that needs alternate paths depending on the input?



Example-1

Write a program that will report if a given integer is odd.



If reminder of N/2 is one (or not zero) then N is odd

Condition-Action Decomposition (CAD)

Write a program that will report if a given integer is odd

The condition and the action may be obvious:

– *If number is odd, print the number.*

CA Decomposition:

CONDITION	ACTION
N % 2 == 0, TRUE	PRINT “not odd”
N % 2 == 0, FALSE	PRINT “odd’

Alternatively:

CONDITION	ACTION
N % 2 != 0, TRUE	PRINT “odd”
N % 2 != 0, FALSE	PRINT “ not odd’

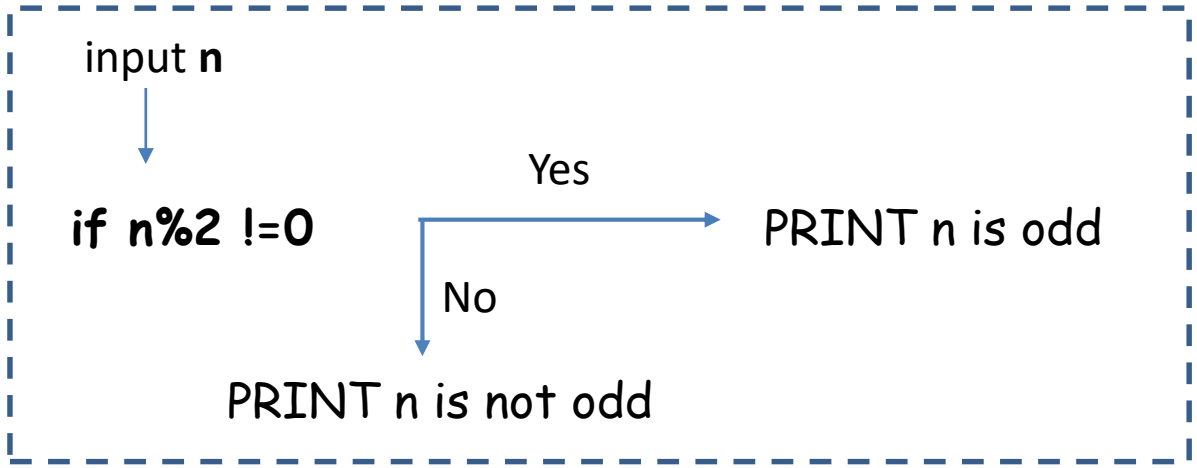
CONDITION	ACTION
N % 2 == 1, TRUE	PRINT “odd”
N % 2 != 1, FALSE	PRINT “ not odd’

Cursory Sketches (CS)

Write a program that will report if a given integer is odd

A Cursory Sketch is another tool, very helpful for the decisions you will need in a program.

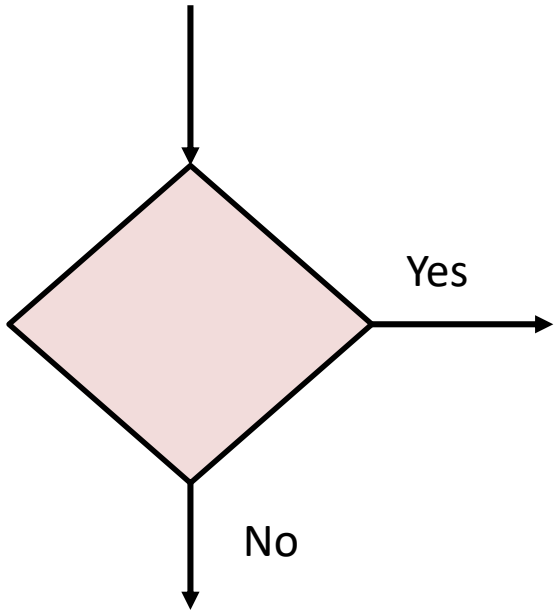
For instance, an initial sketch of our sample problem yields the following:



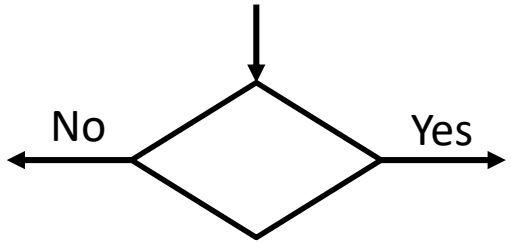
cursory sketch: Bosquejo superficial del flujo del programa mostrando condiciones y acciones tomadas. They are pretty much the same as a flowchart without the representacion of program actions with geometric shapes

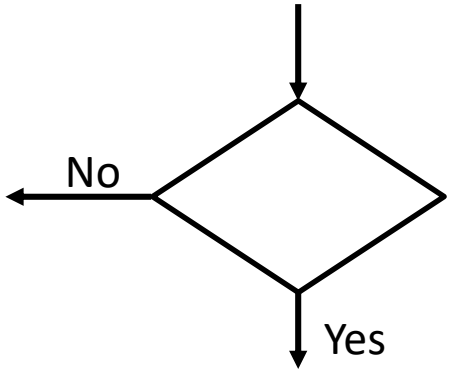
Flowchart

For decision-making (or selection), there is a brand new shape: the diamond (a rhomb).

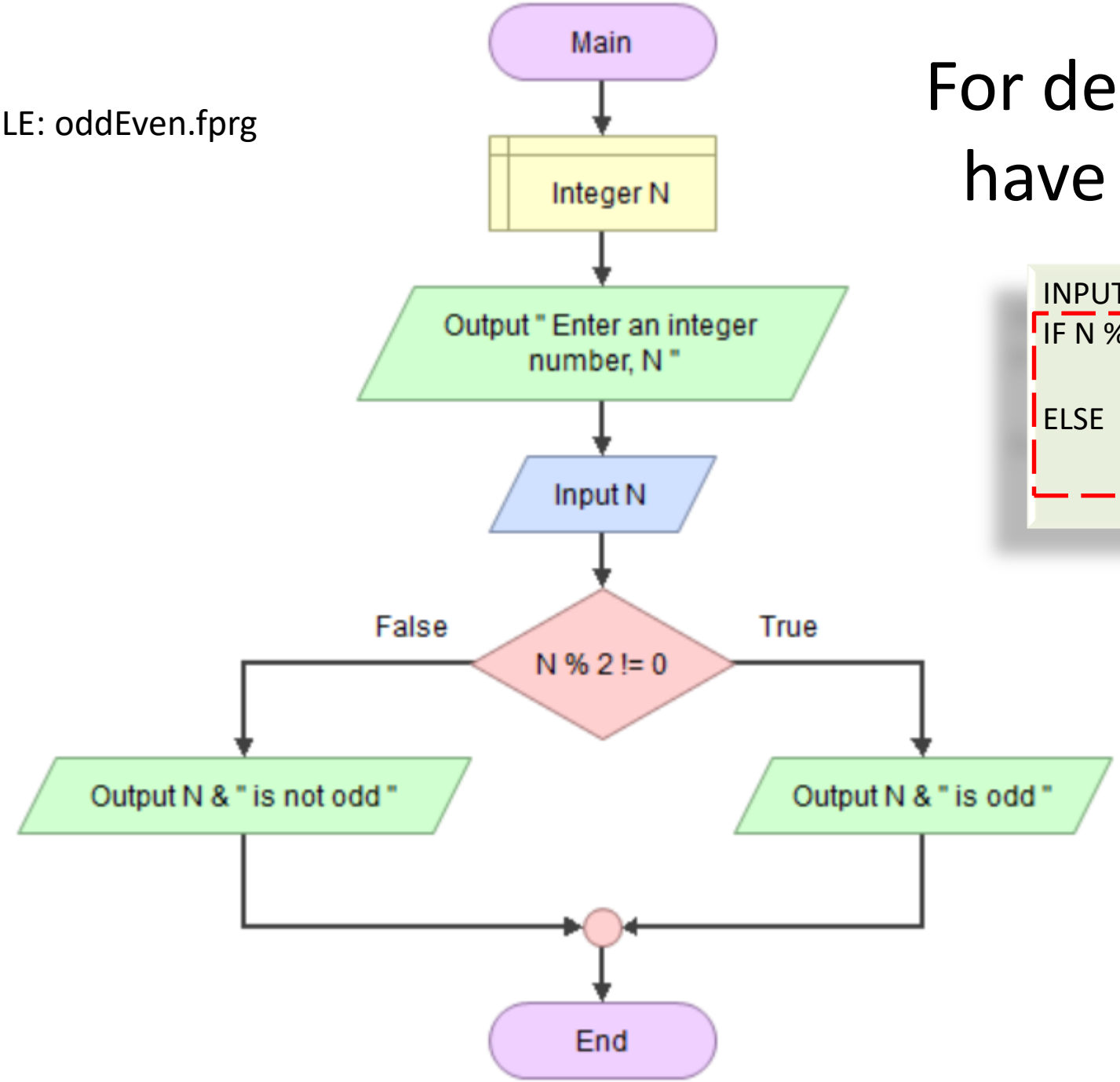


- Selection (or decision).
- The “question” or “condition” being asked goes inside the diamond.
- An arrow representing information goes into the diamond
- An arrow for each “answer”: ‘yes’ or ‘no’ protrudes from the diamond.
- You are free to place arrows in other positions on the diamond, e.g.,





FILE: oddEven.fprg



For decision making we have a new structure

```
INPUT N
IF N % 2 != 0
        PRINT N, " is odd "
ELSE
        PRINT N, " is not odd "
```

If structure

- Mark each of the arrows coming from the decision element with the appropriate "answer," (true or false; yes or no)

- Dependency
- Top down drawing

When to use logical decision-making?

- Whenever you need to make a decision (e.g., such as whether or not a number is odd), you need to implement a logical decision making structure in your solution, i.e., if statement.

- To identify such situations in a problem statement, you look for cases or conditions, which show after words such as, ‘if’, ‘whether’, (expresiones condicionales que conducen a hacer o dejar de hacer algo)

- From these cases, you can decipher the condition that the decision will be based on, and the actions that will be taken when this condition is true or false.

Relational Operators

Relational or comparison operators	
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
==	equal to
!=	not equal to

Programming languages provide the following relational operators (sometimes with slightly different syntax) to make mathematical comparisons between variables or data:

```
e.g.,
a=1, b =2
IF a>b
   a=a+1
ELSE
   b=b-1
```

Three symbols are used for not equal, depending on language:

!=
<>
~=

Selection Symbols

- It is interesting to notice that these can be paired up:

Symbol	opposite to
>	<=
<	>=
==	!=

EXAMPLE:

```
SET score=87
IF score>=90
    PRINT 'you got A'
ELSE
    PRINT 'you didn't got A'
```

Explanation: IF score>=90 is false, then you don't need to ask again if score<90. It's implied

‘AND’ True Table

Cond1	Operator	Cond2	Result
T	AND	T	T
T	AND	F	F
F	AND	T	F
F	AND	F	F

Expressions containing ‘*and*’ are true only if both conditions are true

If any of the conditions is *false*, the whole expression is false, E.G.:

```
IF day ==“Saturday” AND weather ==“sunny”
   PRINT “Let’s go to the beach!”
```

‘OR’ True Table

Cond1	Operator	Cond2	Result
F	OR	F	F
T	OR	F	T
F	OR	T	T
T	OR	T	T

Statements containing ‘*or*’ are false when both conditions are false.

When you have more than two conditions, execute them left to right, one by one. If operators are different, operator precedence applies: AND goes first, OR goes next.

If any of the conditions is *true*, the whole expression is true, E.G.:

```
INPUT month
IF month=="June" OR month=="July" OR month=="August"
    PRINT "Yahoo! Summer vacation!"
```

! Negation True Table

!	Cond	result
!	T	F
!	F	T

Example
a=1; b=3
c=a>b # c is FALSE
c=!c # c changed to TRUE

NOT EQUAL TO: != [or also: ~=, <>]

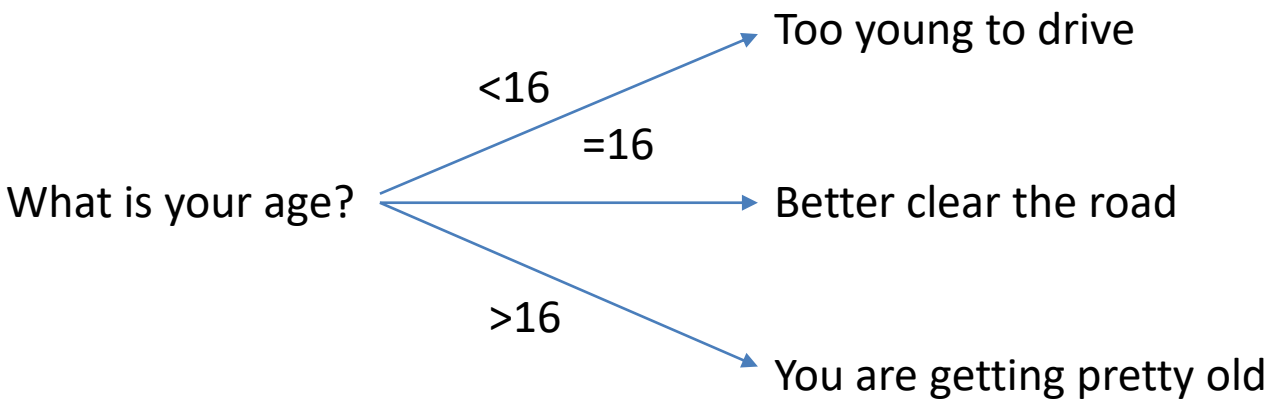
- Example:

```
INPUT day
IF day != 'Saturday' AND day !='Sunday'
PRINT 'Yuk, another class day'
```

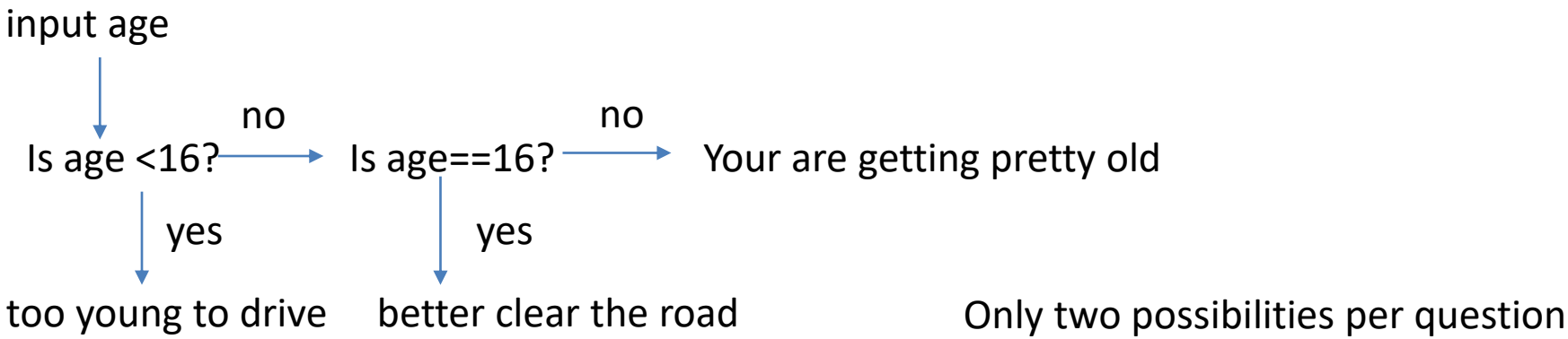
IMPORTANT

- All conditions in computer programming must evaluate to two possibilities a Yes/No (or True/False) question. You cannot, for instance, have a condition which directly branches into three possibilities like this:

Problem:
Ask user age and advice him/her if he/she can drive



The right ways is:



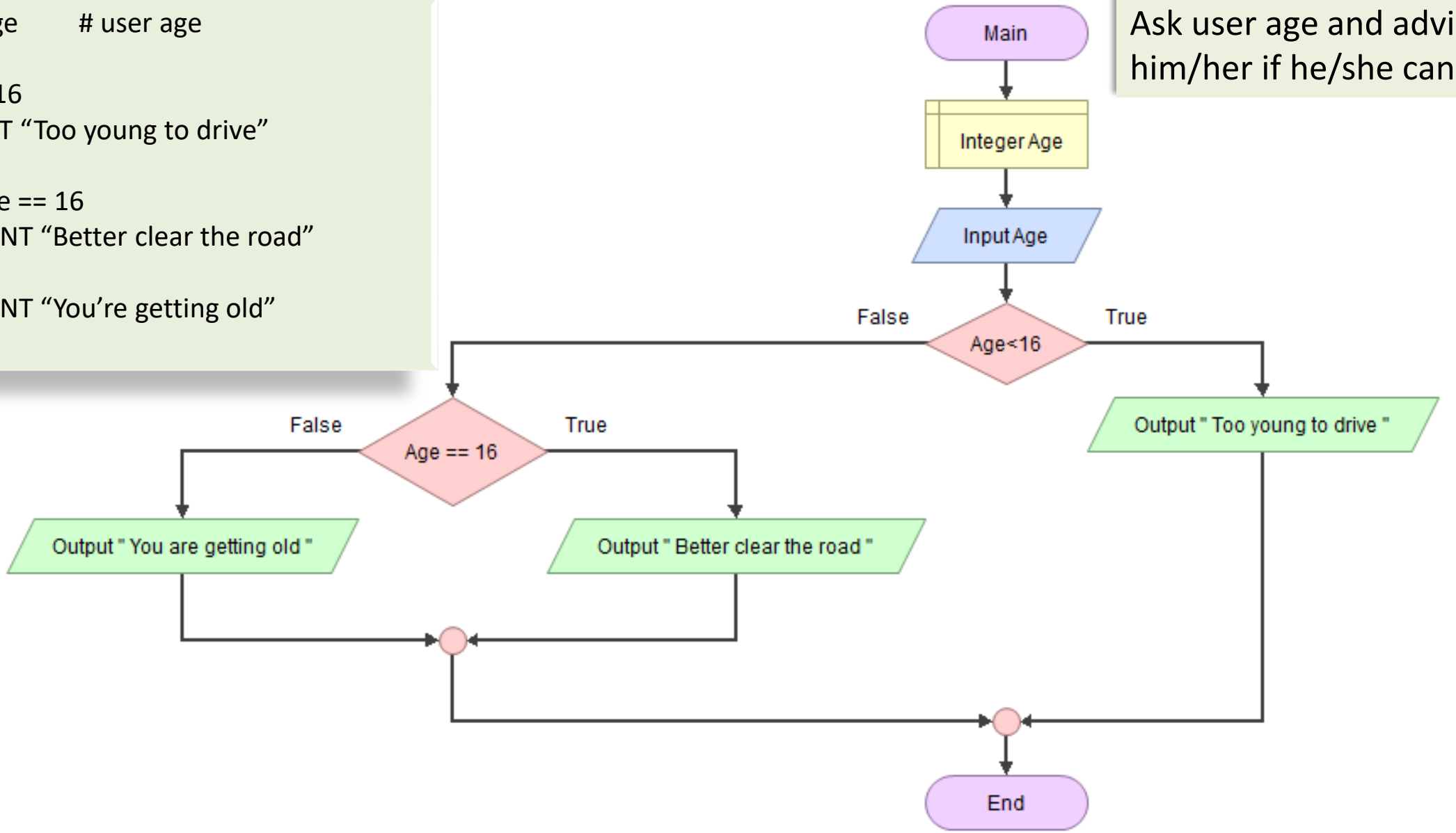
Only two possibilities per question (or logical expression)

Nested if statements

```
INPUT age       # user age

IF age < 16
    PRINT “Too young to drive”
ELSE
    IF age == 16
      PRINT “Better clear the road”
    ELSE
      PRINT “You’re getting old”
```

Problem:
Ask user age and advice him/her if he/she can drive



Selection Structure: Three versions

Allows the program flow to run into one of several choices

```
IF condition
        statements
```

```
IF condition
        statements-1
ELSE
        statements-2
```

```
IF condition-1
        stats-1
ELSEIF condition-2
        stats-2
        …
ELSEIF condition-n
        stats-n
ELSE
        stats-n+1
```

NESTED structures:

```
IF condition
        statements-1
ELSE
        statements-2
        IF condition-2
          statements-3
```

Flowgorithm don't have the multialternatives if statement, sorry.

"ELSEIF" is one word;
"ELSE" and its statements are optional

Solve previous problem with multialternatives

Selection Structures can be nested,i.e., one inside another. NO limits for nesting.

PYTHON Selection Structure: Three versions

Allows the program flow to run into one of several choices

```
if <condition>:
        statements
```

```
if <condition>:
        statements-1
else:
        statements-2
```

```
if <condition-1>:
        stats-1
elif <condition-2>:
        stats-2
        …
elif <condition-n>:
        stats-n
else:   # optional
        stats-n+1
```

NESTED structures:

```
if <condition-1>:
        statements-1
else:
        statements-2
            if <condition-2>:
              statements-3
```

Python uses ":" after each branch of if structure

python has also these three versions of if statements. In python small caps key words are used instead.

"elif" is one word (it is an abbreviation of else if)
One "else" and its statements are optional
Don't forget ":" symbol

Example-2

Based on the temperature, either tell the user to bring a heavy jacket (colder than 45 degrees), light jacket (between 45 and 75 degrees), or no jacket at all.

- *Look for cases ("diferentes situaciones, casos")*
- *They are usually constructs that can be reworded into if statements.*

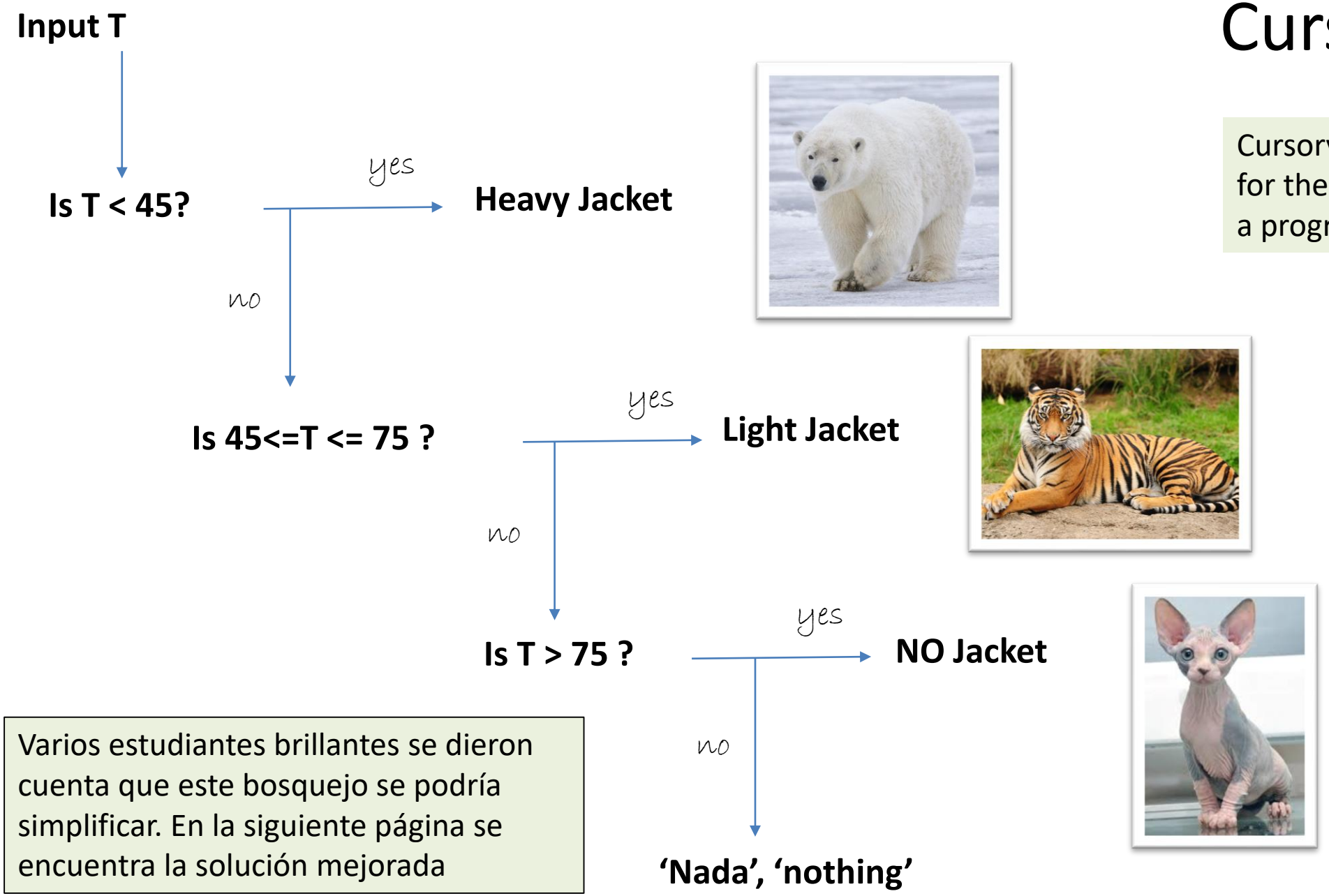
CA Decomposition:

- **Condition:** If the temperature is less than 45 degrees
- **Action:** Tell user to bring a heavy jacket

- **Condition:** If the temperature is between 45 and 75 degrees
- **Action:** Tell user to bring a light jacket

- **Condition:** If the temperature is greater than 75 degrees
- **Action:** Tell user not to bring any jacket

Cursory Sketch

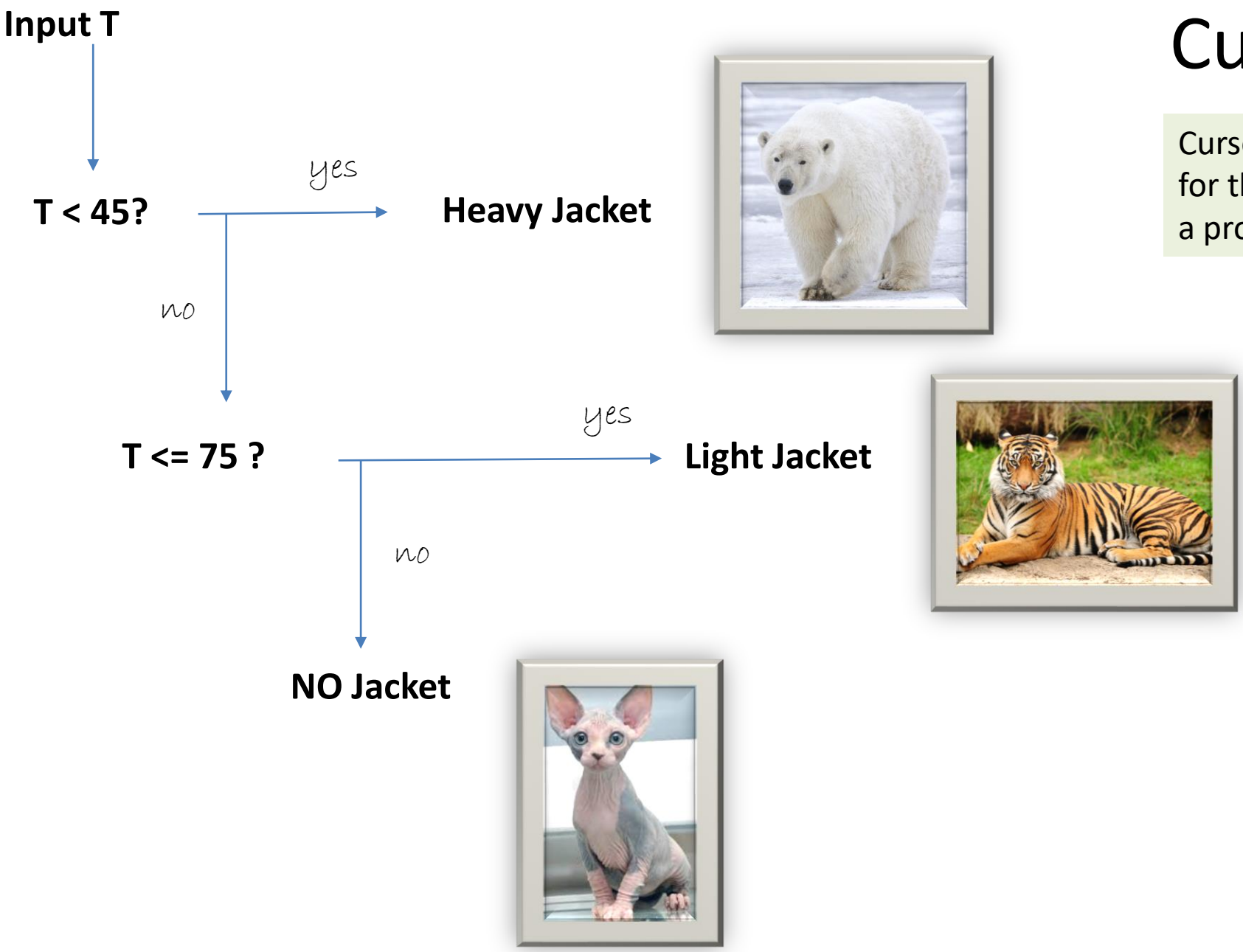


Cursory sketches are very helpful for the decisions you will need in a program.

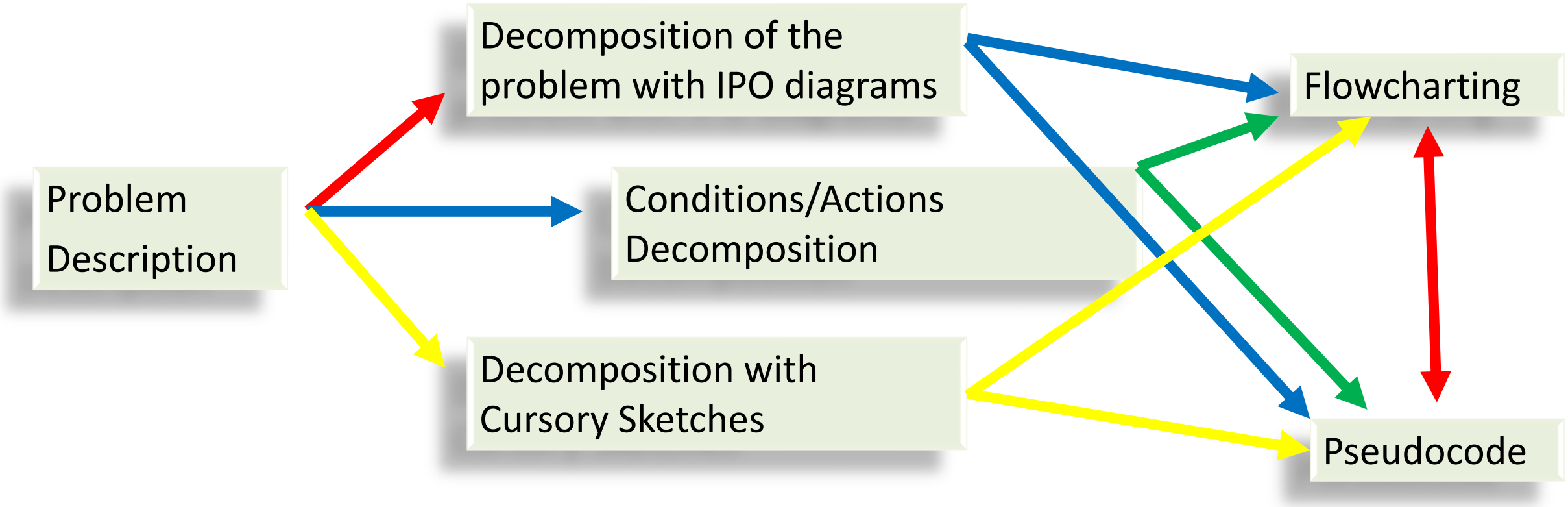
Varios estudiantes brillantes se dieron cuenta que este bosquejo se podría simplificar. En la siguiente página se encuentra la solución mejorada

Cursory Sketch

Cursory sketches are very helpful for the decisions you will need in a program.



Tools for Problem Solving and Program Design

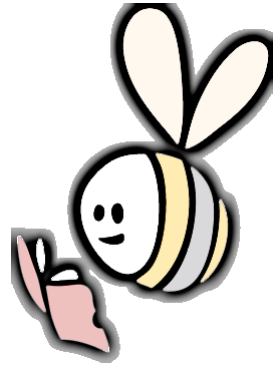


How many tools did you apply to arrive at the pseudocode or flowchart?

Suggested Exercises: Selection Section

Ex 25. Movie Theater. Write a program that tells the user what type of movie they can attend based on their age, if they are with their parents, if they are students (depending on weekday), and their amount of money

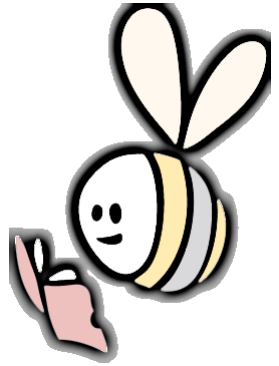
Under 13	G
Under 13 w/parent:	G, PG
13 and Over and Under 16:	G, PG
Under 16 w/parent	G, PG, R
16 and Over	G, PG, R
Student (Monday trough Thursday)	$4.00
Matinee	$4.50
Evening	$6.50



Suggested Exercises: Selection Section

- ***Ex 26 Bank Account****. Write a program that will take as input the user's bank account balance and the type and level of account they have. Based on this information and the below rate table, determine the interest rate they are receiving.*

Type of Account	Level	Minimum Balance	Interest Rate
Student	Standard	$25	1.3 %
Personal	Standard	$0	1.2 %
Personal	Gold	$1000	1.9 %
Personal	Gold	$5000	2.3 %
Business	Standard	$1500	1.7 %
Business	Platinum	$10000	2.5%



Suggested Exercises: Selection Section

Ex-27. Restaurant Tip. *Write a program that will take as input the type of restaurant the user ate at, the cost of the meal, the number of people in his/her party, and how good the service was.*

- Determine the dollar amount of the tip, based on:
  - Base Tip:
    - Food Truck 5%
    - Diner: 12%
    - Good Restaurant: 15%
    - Fancy Restaurant: 20%
  - Additions/Subtractions:
    - Poor Service: -2%
    - Good Service: +0%
    - Excellent Service: +2%
  - Number of parties:
    - 1-5 in party: +0%
    - 6-10 in party: +3%
    - more than 10: +5%

Quiz: Friday before midnight (Aug 23) Section 096
Flowgorithm+Pseudocode
Email Subject: Selection Structure
Test the program with $15.15 cost of the meal before tax, before tip.