

UC Berkeley's CS10 Fall 2017 Final Exam: Prof. Dan Garcia

Your Name (first last)

SID

Lab TA's Name

← Name of person on left (or aisle)

Name of person on right (or aisle) →

What's that Smell? Oh, it's Potpourri! (2 pts each for 1-6, low score dropped)

Fill in the correct circles & squares completely...like this: ● (select ONE) ■ (select ALL that apply)

Question 1: Which were *not* discussed in the "Saving the world with computing" lecture? (select ONE)

- A "Digital Body Double" allows experimental treatments to be applied before they'd be done on a patient
- Simulations allow us to ask whether global warming impacts can be reduced if greenhouse gases are cut.
- While some projects involve teams, the majority of the projects were completed by individual researchers.
- The fastest computers used for scientific research have been parallel for a long time.
- None of the Above.

Question 2: Which of the following is *not* usually characterized by the time "after the singularity"? (select ONE)

- Moore's law ends
- Artificial general intelligence (AGI)
- Software writing software
- Systems beyond human control
- None of the Above.

Question 3: When we solved the "10...0" game in class (the game starts with 10 sticks, each player removes 1 or 2, first person to get to 0 wins), it worked fine. However, when we tried to solve the "50...0" game (same rules as the original game, except we start from 50 instead of 10),

it didn't return even though we waited a long time. Why would that be? (select ONE)

- There's actually an error in the solver that only gets triggered when the number gets above 10
- We are stuck in an infinite iterative loop, it'll never return
- The base case is incorrect, and we're stuck in an infinite recursion
- We were doing lots of redundant calculations
- None of the above

Question 4: It would be nice to know if our program from question 3 above would *ever* (or *never*) return. What should you say if someone wanted to write a program that would take in a program (e.g., `solve`) and input (e.g., 50) and return whether that program run on that input will *eventually* return? (select ONE)

- You don't need to write it, it's been done already!
- It can't be written in python, but it can be written in Snap!.
- It can't be written in Snap!, but it can be written in Python.
- It's certainly doable (in both Snap! and Python), but it might take a long time to return its answer.
- It's certainly doable (in both Snap! and Python), but it might need a lot of memory for its computation.
- It's not possible to write that program!
- None of the above

Question 5: Who would be the best person to answer question 4 above? (select ONE)

Gates	Jobs	Turing	Hopper	Yelick	Englebart	Sutherland	None of these
-------	------	--------	--------	--------	-----------	------------	---------------

```
## This code is straight from lecture, provided for
## reference for question 3. (Also, the second line
## below was too long so it wrapped when pasted here)

def children(position):
    return [do_move(position,move) for move in (cont'd)
generate_moves(position)]

def solve(position):
    if primitive(position) != UNDECIDED:
        return primitive(position)
    else:
        values = [solve(c) for c in children(position)]
        if LOSE in values:
            return WIN
        elif TIE in values:
            return TIE
        else:
            return LOSE
```

Question 6: What is $(111_3 + 11_2)$ written in Hex? (select ONE)

<input type="radio"/>							
0xC	0xD	0xE	0xF	0xG	0x10	0x11	None of these

Question 7a-b: Beethoven wasn't the only great composer... (9 = 3+6 pts)

We've provided some helper reporter blocks that work on words.

Block	Description	Example
swap	Swap the first two letters	 eBars
right	Rotate a word to the right (i.e., move the last letter to the front)	 sBear
left	Rotate a word to the left (i.e., move the first letter to the end)	 earsB

a) If **A****D****B****C** is written like this →
what input to the nested blocks
`swap(right())` would cause
them to output **RATS**?

A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



R	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
T	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
S	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(fill in the blank squares on the right)

b) If `swap right` is written like this →
What set of nested blocks
if called on **STOP** would return **SPOT**?

swap	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
right	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
left	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

swap	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
right	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
left	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(fill in the blank squares on the right; you shouldn't need more than we provided)

Question 8: Three two one...launch! (6 = 3+3 pts)

The following is a helper command that increases **SALARY**. Below each script on the right, write ALL the possible values of **SALARY** after each script has run to completion. (The **launch** block allows its input script to run in parallel, and does not wait for its input to be finished before proceeding to the next block below it)

```

Give Raise amount #
script variables old salary new salary
wait pick random 1 to 10 secs
set old salary to SALARY
wait pick random 1 to 10 secs
set new salary to old salary + amount
wait pick random 1 to 10 secs
set SALARY to new salary
set amount to 0
    
```

```

set SALARY to 10
Give Raise 1
Give Raise 3
    
```

```

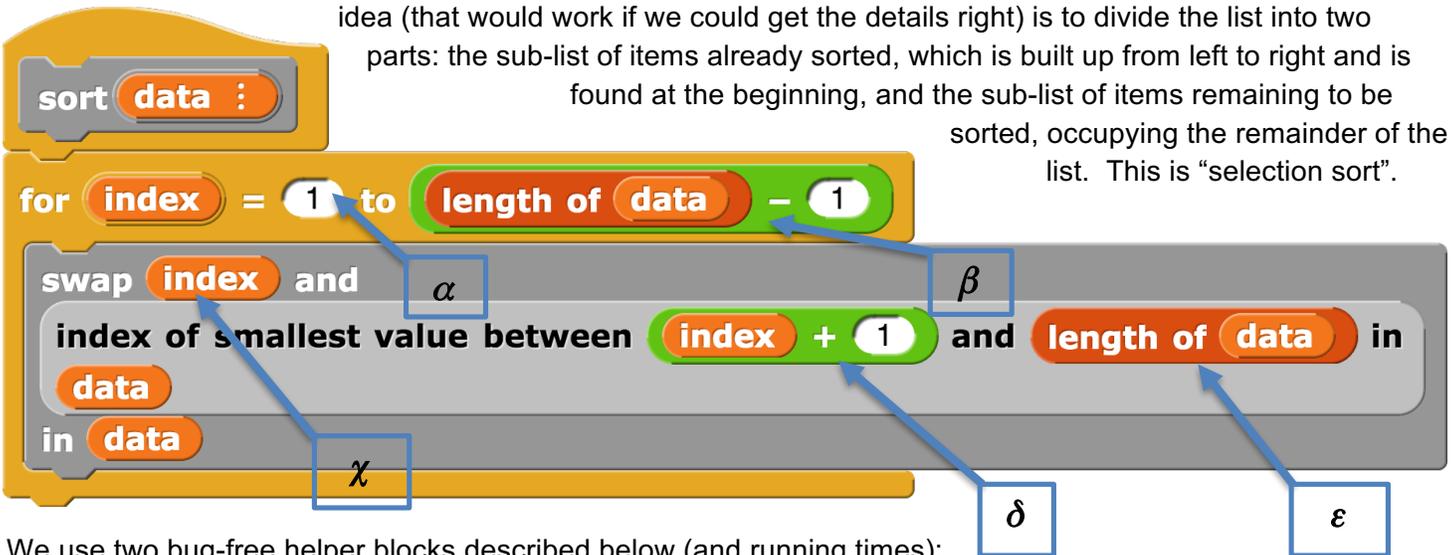
set SALARY to 10
launch Give Raise 1
launch Give Raise 3
    
```

<input type="checkbox"/>	10	<input type="checkbox"/>	10
<input type="checkbox"/>	11	<input type="checkbox"/>	11
<input type="checkbox"/>	12	<input type="checkbox"/>	12
<input type="checkbox"/>	13	<input type="checkbox"/>	13
<input type="checkbox"/>	14	<input type="checkbox"/>	14
<input type="checkbox"/>	15	<input type="checkbox"/>	15

Question 9a-d: Some sort of problem with my code... .. (14 = 2+3+5+4 pts)

SID: _____

We would like to take an unsorted list and *sort* it (into *increasing* order – the smallest element at index 1, the second smallest at index 2, etc). We’ve tried to write code to do this for us, but we believe it has a bug. The idea (that would work if we could get the details right) is to divide the list into two parts: the sub-list of items already sorted, which is built up from left to right and is found at the beginning, and the sub-list of items remaining to be sorted, occupying the remainder of the list. This is “selection sort”.



We use two bug-free helper blocks described below (and running times):

- Search the **data** list for the smallest value between the **left index** and the **right index** (inclusive, i.e., including the elements at the left & right indices) and report the index of the smallest value. (linear)

```
index of smallest value between left index and right index in data
```

- Swap the elements at **left index** and **right index** in the **data** list. (constant)

```
swap left index and right index in data
```

a) What is running time of sort?
(select ONE)

<input type="radio"/>					
constant	logarithmic	linear	quadratic	cubic	exponential

b) If **data** is (4 3 2 5 1), (i.e., a 5-element list whose first element is 4, second element is 3, etc. and whose last element is 1), what is **data** after **sort (data)** runs? Write your answer following the convention to the right, which shows how to write (4 3 2 5 1).

1					
2					
3					
4					
5					

1					
2					
3					
4					
5					

c) What 10-element list (containing the numbers 1 through 10 in some order), when passed to **sort**, would be correctly sorted after **sort** runs? (fill in the blank squares on the right following the convention)

1									
2									
3									
4									
5									
6									
7									
8									
9									
10									

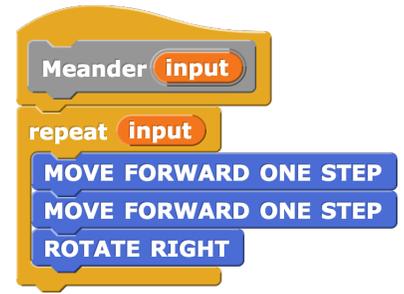
d) Briefly describe the single, very small change needed to fix the bug. (select ONE from each group)

<input type="radio"/>	<input type="radio"/>	expression	<input type="radio"/>	by	<input type="radio"/>	<input type="radio"/>						
Increase	Decrease		α	β	χ	δ	ϵ	1	2			

Question 10: *Meander, down by the school yard...* (6 = 3+3 pts)

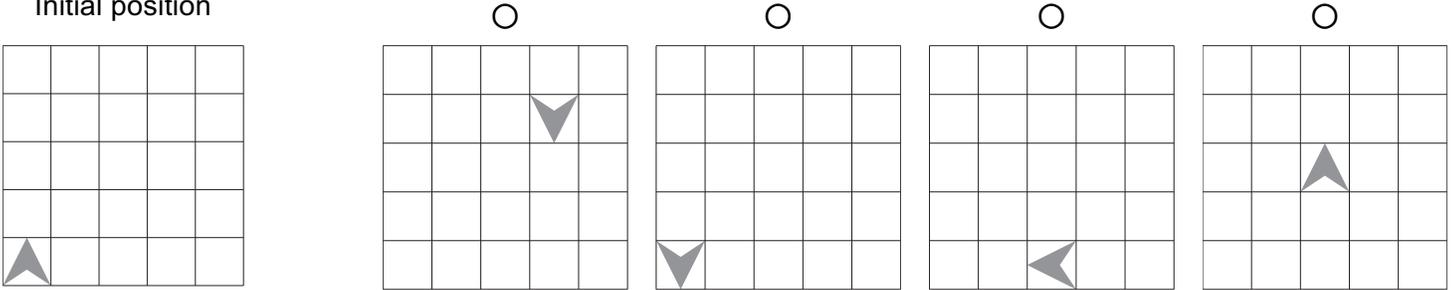
SID: _____

The question below uses a robot in a grid of squares. The robot is initially in the bottom-left square of the grid and facing up. Code for the procedure **Meander** is shown here. Assume the input has been assigned a positive integer value (e.g., 1, 2, 3, ...). The **ROTATE RIGHT** block has the same function as the **Turn ↻ 90 degrees** block.



a) Which shows a possible result of calling **Meander**? (select ONE)

Initial position



b) If the input to **Meander** were ∞ , how many *total different squares would ever be visited*? (select ONE)

<input type="radio"/>																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	∞

Question 11: *Commander data, at your service...* (9 = 3+6 pts)

A large data set (a list of student records) contains information about all students majoring in computer science in colleges across the USA. We don't know how a student record is encoded, but we know it contains:

- The student's gender
- The state in which the student attends college
- The student's GPA on a 4.0 scale

a) Which of the following could be answered by analyzing only information in the data set? (select ONE)

- Do students majoring in computer science tend to have higher GPAs than students majoring in other subjects?
- How many states have a higher percentage of female computer science majors than male computer science majors attending college in that state?
- What percent of students attending college in a certain state are majoring in computer science?
- Which college has the highest number of students majoring in computer science?

b) How could we extract a list of the *states* (with no duplicates) with CS majors from the data set *in the fewest steps*? (select ONE)

- First, use **keep**, and then remove duplicates
- First, use **map**, and then remove duplicates
- First, use **keep**, then use **map**, and then remove duplicates
- First, use **map**, then use **keep**, and then remove duplicates
- First remove duplicates, then use **keep**
- First remove duplicates, then use **map**
- First remove duplicates, then use **keep**, then use **map**
- First remove duplicates, then use **map**, then use **keep**

Question 12: Berkeley Python Flying Circus... (16 = 8Q x 2 pts/Q)

SID: _____

a) What does `["cal", "berkeley", "stanford"][1][2]` evaluate to? (select ONE)

<input type="radio"/>							
"a"	"r"	"t"	"b"	"cal"	"berkeley"	Error	None of the above

b) What does `[x*10 for x in range(3) if x != 1]` evaluate to? (select ONE)

<input type="radio"/>							
[10 20 30]	[0 10 20]	[20 30]	[0 20]	[1 20 30]	[0 1 20]	Error	None of the above

We want to write a dictionary reverser, which will take a dictionary of keys and values and make the values the keys and have the original keys become a list of values.

So, if `mydict` is `{1:3, 2:3, 8:9}`, then

`dictionary_reverser(mydict) → {9:[8], 3:[1,2]}`.

Write this by filling in the blanks with the correct expressions.

```
def dictionary_reverser(D):
    R = {}
    for k in D: # keys
        if cANS in dANS:
            eANS.append(fANS)
        else:
            gANS = hANS
    return R
```

c) What is `cANS`? (select ONE)

<input type="radio"/>								
D	R	k	D[k]	R[k]	k[D]	k[R]	R[D]	D[R]

d) What is `dANS`? (select ONE)

<input type="radio"/>								
D	R	k	D[k]	R[k]	k[D]	k[R]	R[D]	D[R]

e) What is `eANS`? (select ONE)

<input type="radio"/>											
D[k]	R[k]	k[D]	k[R]	R[D]	D[R]	k[D[R]]	k[R[D]]	D[k[R]]	D[R[k]]	R[D[k]]	R[k[D]]

f) What is `fANS`? (select ONE)

<input type="radio"/>											
D	R	k	[D]	[R]	[k]	D[k]	R[k]	k[D]	k[R]	R[D]	D[R]

g) What is `gANS`? (select ONE)

<input type="radio"/>											
D[k]	R[k]	k[D]	k[R]	R[D]	D[R]	k[D[R]]	k[R[D]]	D[k[R]]	D[R[k]]	R[D[k]]	R[k[D]]

h) What is `hANS`? (select ONE)

<input type="radio"/>											
D	R	k	[D]	[R]	[k]	D[k]	R[k]	k[D]	k[R]	R[D]	D[R]

Question 13: Call me maybe... if you can find me in your contact list! (15 pts)

SID: _____

We're trying to write a block that takes in a word fragment (the kind you'd type to a search prompt) and returns the *contacts* (not just the names!!) from a contact list whose names match. We don't know how a contact is

encoded, only that **CONTACT LIST** contains a list of them. There are two helper blocks we provide:

Helper Block	Description
	Given a contact, returns its name
	Given a string and a fragment, return whether the fragment is in the string. Example: 

At the bottom is the skeleton for the block to author. Note that there is a greek symbol below each slot. To write this code, you place the blocks and variables from the left rows in the "slots" indicated by greek symbols in the columns by filling in the grid intersection. The number of squares marked in each row corresponds to the number of times that block/value/variable appears in the solution. Furthermore, there should be *at most* one square marked in each column: one if something is put in this input and none if it is left blank.

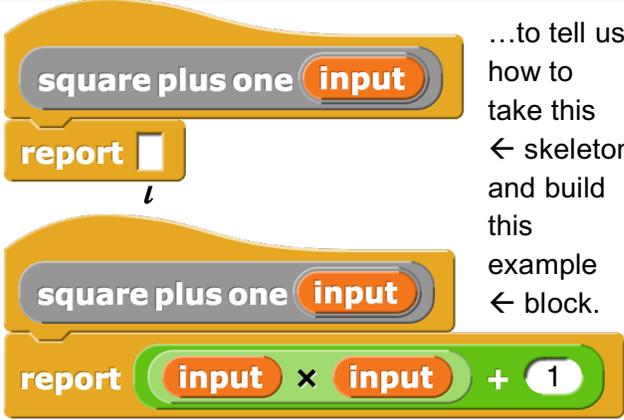
We show an example with the **square plus one** block here.

EXAMPLE

Put these blocks, values, and variables ...

		...into these slots...				
		ι	φ	κ	λ	μ
	φ κ	■				
	λ μ		■			
	1			■		
					■	

...to tell us how to take this ← skeleton and build this example ← block.



Put these blocks and variables ...

		...into these slots							
		α	β	χ	δ	ϵ	ϕ	γ	η
	fragment								
	α								
	β χ								
	δ ϵ								
	ϕ								
	γ η								
	fragment								
	CONTACT LIST								