CT-STEM

Computational Thinking in Science, Technology, Engineering, and Mathematics

Computer Science Skills

Version 0.1

This work is supported by the National Science Foundation under grant CNS-1138461 and is covered by Northwestern University IRB Study STU00058570.

Student Information

First Name

Last Name

Email name@myemail.com

Student ID

School Name

Grade

Date of Birth mm/dd/yyyy

Sex

Race / Ethnicity (check all that apply)

- ☐ Native American
- ☐ Pacific Islander
- ☐ Asian American
- ☐ White (Caucasian)
- ☐ Mexican American or Chicano
- ☐ Puerto Rican
- ☐ Other Latin American
- ☐ Black or African American

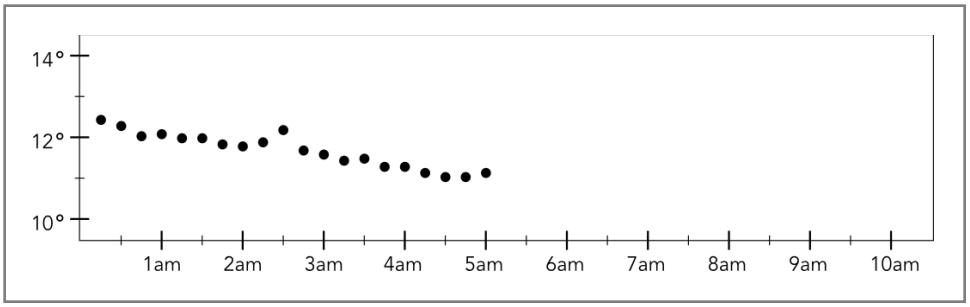
Other

A climate scientist wrote this computer program to measure air temperatures at a weather station.

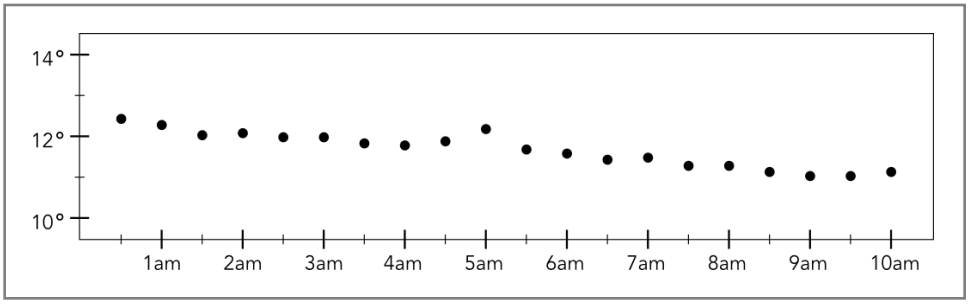
```
repeat 20 times:
   pause-seconds(450)
   record-temperature
```

Item 1: Which of these plots shows her measurements?

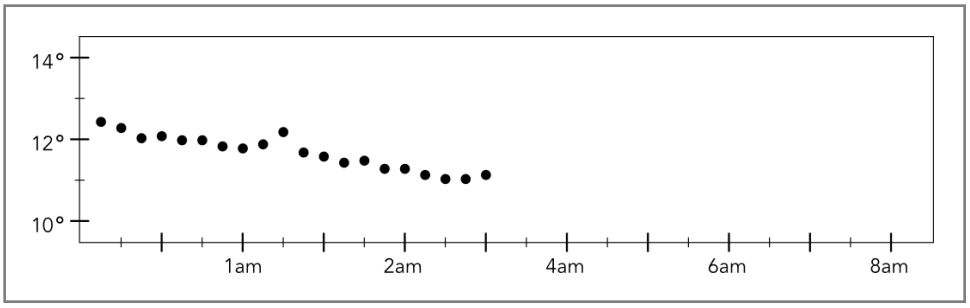
A.



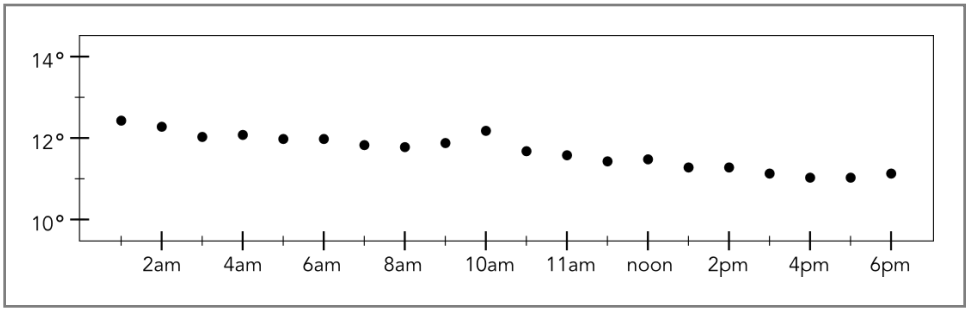
B.



C.



D.



A biologist is studying the beak sizes of a type of bird living on a small island. A sample of his data is shown below.

Year	Month	Bird_ID	Sex	Beak_Size (mm)
2008	01	F113	Male	10
2008	01	F119	Male	6.2
2008	02	F215	Female	11.1
2008	03	F085	Female	4.9
2008	03	F072	Female	6.6
2008	03	F115	Male	6.8
2008	05	F022	Female	10.2
2009	04	F113	Male	10.1
2009	04	F119	Male	6.2

He wrote this program to help understand his data:

```
var sum = 0
var count = 0
for each row in table:
   if (row.Year = 2008) and (row.Sex = "Male") then:
      sum = sum + row.Beak_Size
      count = count + 1

print sum / count
```

Item 2: What does this program do?

A. ◯ Counts all birds measured in 2008

B. ◯ Counts all male birds measured in 2008.

C. ◯ Computes the standard deviation of male beak size in 2008.

D. ◯ Computes the average beak size of male birds measured in 2008.

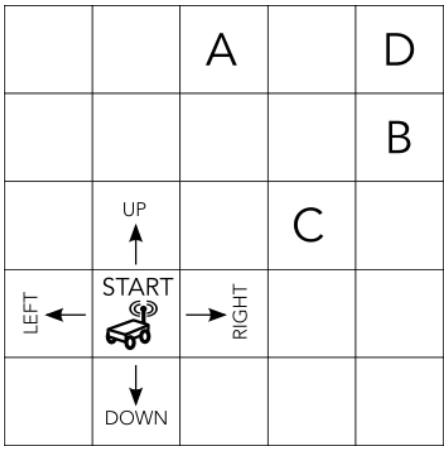
The same biologist wants to know the maximum beak size of all finches from 2008 to 2010. He wrote this program, but it's not working correctly.

```
Line 1:     var maximum = 0
Line 2:     for each row in table:
Line 3:        if (row.Year >= 2008) or (row.Year <= 2010) then:
Line 4:           if (row.Beak_Size > maximum)
Line 5:              maximum = row.Beak_Size
Line 6:     print maximum
```

Item 3: What is the problem and how would you fix it?

Enter your response here

A team of engineering students is building an autonomous robot for a contest. To win, they must program their robot to move around a grid while avoiding obstacles. The robot moves exactly one square at a time either up, down, left, or right.



The team started by writing this program to move their robot.

```
move-up
move-right
move-up
move-right
move-right
```

Item 4a: After running the program, which square will the robot stop on.

A. ◯ Square A

B. ◯ Square B

C. ◯ Square C

D. ◯ Square D

The team tried changing their program to this:

```
move-up
move-right
move-up
move-up
```

Item 4b: Now where will the robot stop?

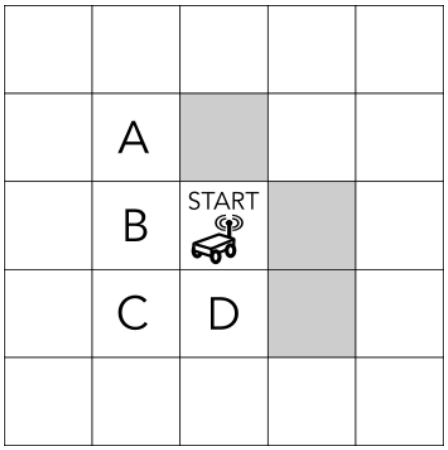
A. ◯ Square A

B. ◯ Square B

C. ◯ Square C

D. ◯ Square D

The team of engineering students added sensors to their robot to help it avoid obstacles (shown as grey squares on the grid). Then they wrote this program:




```
if obstacle-left then:
   move-right
if obstacle-up then:
   move-down
if obstacle-right then:
   move-left
if obstacle-down then:
   move-up
```

Item 5a: On which square will the robot stop?

A. ◯ Square A
B. ◯ Square B
C. ◯ Square C
D. ◯ Square D

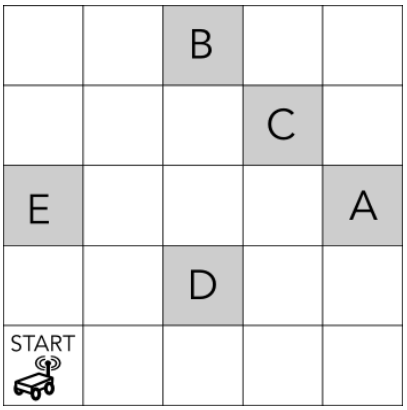
The students then changed their program to this:

```
if obstacle-left then:
   move-right
else if obstacle-up then:
   move-down
else if obstacle-right then:
   move-left
else:
   move-up
```

Item 5b: Now where will the robot stop?

A. ◯ Square A
B. ◯ Square B
C. ◯ Square C
D. ◯ Square D

The students now want to make the robot move in a bigger world. They tried this program, but the robot ran into an obstacle.




```
repeat-forever:
   move-up
   move-right
```

Item 6a: Which obstacle did the robot hit?

A. ◯ Square A
B. ◯ Square B
C. ◯ Square C
D. ◯ Square D
E. ◯ Square E

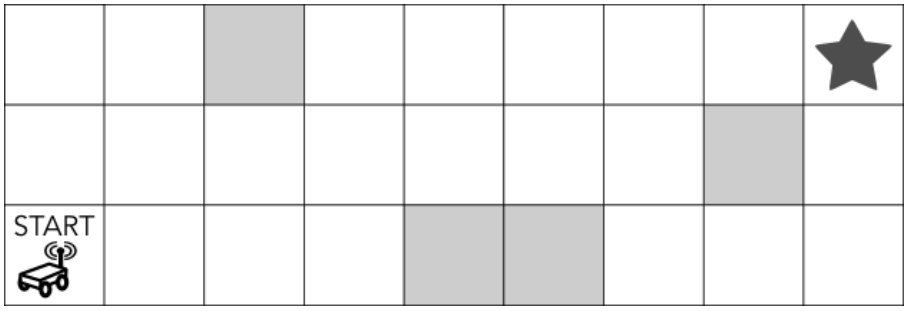
They tried a different program, but the robot still hit an obstacle.

```
repeat-forever:
   move-up
   repeat-until obstacle-right:
      move-right
```

Item 6b: Which obstacle did the robot hit this time?

A. ◯ Square A
B. ◯ Square B
C. ◯ Square C
D. ◯ Square D
E. ◯ Square E

The team needs to program the robot to move from the START square to the STAR square.

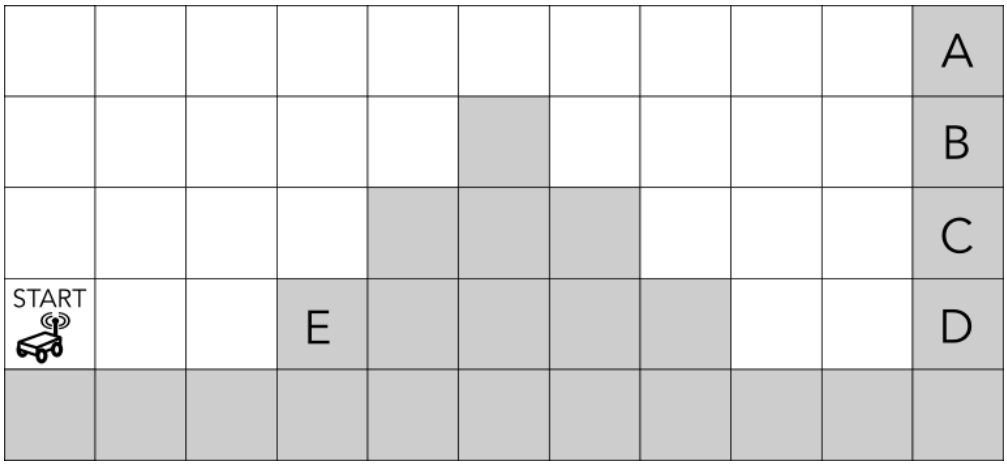


Item 7: Try changing this program to move the robot to the STAR.

```
1  repeat-forever:
2     move-up
3     move-right
4
```

Here are the commands that the robot knows:

`move-up`	`move-down`	`move-left`	`move-right`
`obstacle-up`	`obstacle-down`	`obstacle-left`	`obstacle-right`
`repeat-forever:`	`repeat-until`	`if: else if: else:`	



The team has now tried to get the robot to start avoiding obstacles with this program:

```
define climb-stair-up:
   move-up
   move-right

define climb-stair-down:
   move-right
   move-down

define walk-right:
   if obstacle-right then:
      climb-stair-up
   else:
      right

repeat forever:
   walk-right
```

Item 8: Which obstacle will the robot hit when it runs this program?

A. ◯ Square A
B. ◯ Square B
C. ◯ Square C
D. ◯ Square D
E. ◯ Square E

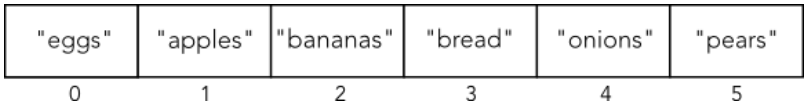
Data Structures

In computer programming, a data structure is a way of organizing information. There are many different kinds of data structures, but two of the most fundamental data structures are lists and lookup tables. For this test, you will read a brief description of these two data structures and then answer some questions about them.

Even if you've never done any computer programming before, there are important clues on this page that will help you answer the questions.

Lists

A list is a type of data structure that stores information in a sequence in a computer's memory. For example, if you wanted a computer program to keep track of items in an online shopping cart, you might place each item in a list.



A list is a good data structure if you need to process many items one at a time. It is also efficient to retrieve the item stored at a particular position in the list. For example, this computer code would get the item at position 4 in the shopping cart.

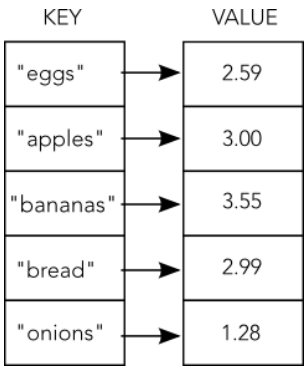
```
print shopping-cart[4]
```

However, lists are not efficient for searching for an item if you don't know its position ahead of time. To search in a list you would have to write a program to examine every item, one at a time.

Lookup Tables

A lookup table is a type of data structure that stores items by names (or key) rather than by position. Each entry in a lookup table consists of a key and a value. For example, you might want to store the cost of each item in your shopping cart. This computer code would print out the cost of an apple in your cart.

```
print shopping-cart["apple"]
```



With a lookup table it is efficient to find an item by its name. However, unlike a list, the items are not stored in any particular order.

Item 9a: A scientist is studying the acidity of sea water. She has entered 500 measurements into a computer over the past year. If she wants to know the minimum, maximum, and average value of all of her measurements, what data structure should she use and why?

A. ◯ A list because the order of the measurements is important.

B. ◯ A lookup table because it is efficient to look up a measurement by its name or key value.

C. ◯ A list because it is good for processing each measurement one at a time.

D. ◯ A lookup table because the order of the measurements is not important.

Item 9b: An astrophysicist need to be able to efficiently find the mass of different stars based on their names. She has a dataset of over 300,000 stars. Which data structure should she use to store this information and why?

A. ◯ A list because the order of the stars is important.

B. ◯ A lookup table because it is efficient to look up a star by its name or key value.

C. ◯ A list because it is good for processing each value one at a time.

D. ◯ A lookup table because the order of the values is not important.

Item 10: A biologist is studying the flowering times of different plants based on environmental conditions. He wants to know how many plants in his greenhouse flowered in the first 10 days of his study, so he wrote this computer program:

```
var count = 0
for each time in flower_times:
   if time <= 10 then:
      count = count + 1
print count
```

Looking at his program, which data structure did he use?

A. ◯ A lookup table because he's looking up each item by name.

B. ◯ A list because he's looking up each item by position.

C. ◯ A lookup table because the order of the items is not important.

D. ◯ A lookup table because lists are not efficient for searching for items if you don't know their posistions ahead of time.

Congratulations!

You're finished. Use any remaining time to go back and check your answers.

CT-STEM Skills Taxonomy

Skills from Computer Science

1. Interpreting Instructions Written for a Computer
2. Defining Instructions for a Computer
3. Developing Modular Computational Solutions
4. Choosing Effective Computational Tools
5. Applying Conditional Logic
6. Effectively Using Recursion and Iterative Logic
7. Creating Abstractions
8. Choosing Efficient Data Structures