DATA 6
Fall 2025

Lisa Yan
Quiz 3

Your name: _____

Your student ID: _____

Your Berkeley email: _____

Your room location: _____

Student ID of the person to your left: _____

Student ID of the person to your right: _____

You have 50 minutes. There are 3 questions of varying credit. (36 points total)

Question:	HC	1	2	3	Total
Points:	1	12	23	0	36

For questions with **circular bubbles**, you may select only one choice.

- ○ Unselected option (Completely unfilled)
- ◎ Don't do this (it will be graded as incorrect)
- ● Only one selected option (completely filled)

For questions with **square checkboxes**, you may select one or more choices.

- ■ You can select
- ■ multiple squares
- ☑ (Don't do this)

Anything written outside the answer boxes or ~~crossed out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation. For coding questions with blanks, you may write at most one statement per blank and you may not use more blanks than provided.

As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others. I will follow the rules of this exam.

Honor Code (HC): I have read and agree to the honor code above.

(1 point) Sign your name: _____

SID: ____________________

Q1 *What Would Python Do? (WWPD)* **(12 points)**

Consider the `truthy` function:

```
def truthy(x):
    return bool(len(x) and x[3])
```

What is the result of evaluating each function call below? If Python will error, select "Error".

Q1.1 (0.5 points) `truthy(make_array(0, 1, 3, 2))`

- ○ `True`
- ○ `False`
- ○ `None`
- ○ Error

Q1.2 (0.5 points) `truthy("")`

- ○ `True`
- ○ `False`
- ○ `None`
- ○ Error

Q1.3 - Q1.7 (5 points) The NumPy function `np.diff` takes in an array `arr` and returns an array of size `len(arr)-1` with elements equal to the difference between adjacent elements. For example, if `example` is a size-5 array with elements 2, 0, −3, 3, and 5, then `np.diff(example)` returns a size-4 array with elements −2, −3, 6, and 2.

Complete the code below so that the function `manual_diff` functions equivalently to `np.diff`. In other words, `manual_diff(example)` should return a size-4 array with elements −2, 3, 6, and 2.

```
def manual_diff(arr):
    ret_arr = make_array() # empty array; no items
    prev = arr.item(0)


    for i in np.arange(__________, __________):
                        Q1.3         Q1.4

        val = ______________________________
                           Q1.5
        diff = val - prev


        ret_arr = ________________________________________
                                   Q1.6

        prev = ____________________
                       Q1.7

    return ret_arr
```

Q1.8 (6 points) Integer division mathematically is **dividend** ÷ **divisor** = **quotient**, with a **remainder**. Consider the code:

```
1 def compute_remainder(dividend, divisor):
2     remainder = dividend
3     while remainder __<A>__ divisor:
4         remainder -= __<B>__
5     return remainder
```

For each of the assigned values of `<A>` and `<B>`, respectively, what are the results of evaluating `compute_remainder(10, 3)` and `compute_remainder(15, 3)`? Fill in the blanks with integer values. If running the code would cause Python to error, write "Error".

`__<A>__`	`__<B>__`	`compute_remainder(10, 3)`	`compute_remainder(15, 3)`
`>=`	`divisor`		
`>`	`divisor`		
`>=`	`1`		

(The rest of this page is intentionally blank.)

SID: _____

Q2 *The Billboard Top 100* **(23 points)**

The Billboard Top 100 is a weekly song ranking chart in the United States. Consider the weekly charts in 2025:

chart_week	rank	title	performer
2025-01-04	1	All I Want For Christmas Is You	Mariah Carey
2025-01-04	2	Rockin' Around the Christmas Tree	Brenda Lee
...	...	...	...
2025-11-08	1	The Fate Of Ophelia	Taylor Swift
2025-11-08	2	Golden	HUNTR/X
...	...	...	...
2025-11-08	100	Favorite Country Song	HARDY

Table 1: Some rows from the `billboard` table (4,500 rows total).

Each row of `billboard` (Table 1) is a song's weekly ranking on the Billboard Top 100 chart in 2025. There are 45 weeks as of Nov 10, 2025, so there are 4,500 rows. Variables:

- `chart_week`: The chart week date as a string (YYYY-MM-DD); all dates are Saturdays.
- `rank`: The current rank of this song on the Billboard Top 100 chart: 1 is highest, 100 is lowest.
- `title`: The title of the song.
- `performer`: The performer/artist of the song.

Your friend defines the below function `hot_or_not` which returns "hot" if the provided `rank` is better than 50 and "not bad" if `rank` is worse than 50. Recall that better rankings are lower numbers, e.g., a rank of 10 is "hot".

```
1 def hot_or_not(rank):
2     if rank < 50:
3         return "hot"
4     elif rank > 50:
5         return "not bad"
```

Your friend's function does not quite work. What is the result of evaluating each function call below? If Python will error, select "Error".

Q2.1 (0.5 points) `hot_or_not(50)`

◯ `"hot"` ◯ `"not bad"` ◯ `None` ◯ Error

Q2.2 (0.5 points) `hot_or_not(-10.5)`

◯ `"hot"` ◯ `"not bad"` ◯ `None` ◯ Error

Q2.3 (3 points) Complete the function `get_day`, which takes a `chart_week` string and returns the day of month as an integer, e.g., `get_day("2025-01-04")` returns `4`, and `get_day("2025-10-25")` returns `25`. Select all expressions that correctly implement `get_day`.

Note: `int("04")` evaluates to `4`.

```
def get_day(date):
    val = ... # your expression here
    return int(val)
```

- ☐ `date[-2:-1]`
- ☐ `date[-2:]`
- ☐ `date.split('-')[2]`
- ☐ `date[8] + date[9]`
- ☐ `"+".join([date[8], date[9]])`
- ◯ None of the above

Q2.4 - Q2.7 (6 points) Assuming a correct implementation of `get_day`, complete the code below to get all rankings from the *first chart week* each month, which is a `chart_week` with a day numbered 1 through 7, inclusive.

The resulting table (see Table 2) should have the same columns as `billboard`.

chart_week	rank	title	performer
2025-01-04	1	All I Want For Christmas Is You	Mariah Carey
2025-01-04	2	Rockin' Around the Christmas Tree	Brenda Lee
...	...	...	...
2025-11-01	1	The Fate Of Ophelia	Taylor Swift
2025-11-01	2	Golden	HUNTR/X
...	...	...	...
2025-11-01	100	3am	Loe Shimmy & Don Toliver

Table 2: Some rows of the table output when your code is run (1,100 rows total).

```
(
  billboard
    .with_columns("day", billboard.apply(get_day, "chart_week"))

    .____Q2.4____(____Q2.5____, ____Q2.6____)

    .____Q2.7____("day")
)
```

The Billboard Top 100 chart is updated weekly on Saturdays. Songs are ranked according to an algorithm that computes points based on streaming plays and sales in the United States:

> ... the Hot 100 [takes] into account paid subscription streams (a 1-point value per play), ad-supported streams (a 2/3-point value per play) and programmed streams[1] (a 1/2-point value per play). Those values are then applied to the chart's formula alongside all-genre radio airplay and digital song sales data ...in descending order of significance. The shift to a multi-level streaming approach ... is reflective of a global push to measure streams in a revenue-reflective and access-based manner. ...

Q2.8 (2 points) Recall **validity** is a quality concerning how accurately a measure captures something in the real world. A friend wants to use a song's Billboard Top 100 ranking as a measure of how well-known the song is in the United States. Based on the description above, list one potential risk to validity that this measurement might have. Limit: 1-2 sentences.

Q2.9 - Q2.10 (5 points) Complete the code below such that when run, `billboard_full` (Table 3, below) is a copy of the `billboard` table with an additional column `full_title`, which is the song title in quotes followed by the artist, separated by the string `" by "`.

chart_week	rank	title	performer	full_title
...	...	...	...	...
2025-11-08	1	The Fate Of Ophelia	Taylor Swift	“The Fate of Ophelia” by Taylor Swift
2025-11-08	2	Golden	HUNTR/X	“Golden” by HUNTR/X
...	...	...	...	...
2025-11-08	100	Favorite Country Song	HARDY	“Favorite Country Song” by HARDY

Table 3: Some rows from `billboard_full` (4,500 rows total) after running the code below.

```
def get_full_title(title, performer):

    return _____
                                        Q2.9

billboard_full = billboard.with_columns("full_title",

            _____)
                                        Q2.10
```

[1] programmed streams: automated radio streams where users cannot select songs

For your convenience, we display `billboard_full` again:

chart_week	rank	title	performer	full_title
...	...	...	...	...
2025-11-08	1	The Fate Of Ophelia	Taylor Swift	“The Fate of Ophelia” by Taylor Swift
2025-11-08	2	Golden	HUNTR/X	“Golden” by HUNTR/X
...	...	...	...	...
2025-11-08	100	Favorite Country Song	HARDY	“Favorite Country Song” by HARDY

Table 4: Some rows from `billboard_full` (4,500 rows total). Identical to Table 3.

Recall from homework: Of the 42 weeks that “Luther” by Kendrick Lamar & SZA ranked on the Billboard Top 100, the song ranked **first** 13 times, i.e., achieved rank 1 in 13 different weeks.

Create the `weeks_at_one` table (Table 5, right), which has the number of weeks each song ranked **first**, in descending order.

full_title	1
“Luther” by Kendrick Lamar & SZA	13
“Ordinary” by Alex Warren	10
“Golden” by HUNTR/X	8
...	...

Table 5: The `weeks_at_one` table.

Q2.11 (6 points) Consider the below **shuffled** lines of code:

```
1 weeks_at_one = weeks_at_one.select("full_title", 1)
2 weeks_at_one = weeks_at_one.relabeled("count", "1")
3 weeks_at_one = weeks_at_one.sort(1, descending=True)
4 weeks_at_one = billboard_full
5 weeks_at_one = weeks_at_one.group("full_title")
6 weeks_at_one = weeks_at_one.pivot("rank", "full_title")
```

Select and order the lines of code that produce the resulting `weeks_at_one` table shown (Table 5). In the blanks below, identify by line number which lines of code should be evaluated, in order. You may not need all blanks; mark an ‘X’ if you don’t use a blank.

For example, if your answer is lines 1, 3, 2 (evaluated in that order), fill in:

1st: 1	2nd: 3	3rd: 2	4th: X	5th: X	6th: X

Your answer:

1st:	2nd:	3rd:	4th:	5th:	6th:

SID: _____

Q3 *Just for fun!* **(0 points)**

Q3.1 Draw something fun, or write a message for the staff! Or leave this blank!