**Note**: Please **SUBMIT** each question individually before ending the exam to receive score.

**TIME REMAINING**
**0:16:38**

End Exam

Questions

# Questions

## Propositional Logic

3 points possible (graded, results hidden)

We found 3 children discussing something about cows, fish and cats but it was hard to tell if what they were speaking was True(T) or False(F). Can you apply some sort of Propositional Logic to deduce if what they are saying is True(T) or False(F)

Child-1: Cow can eat meat if and only if Fish can swim.

Child-2: Cow cannot eat meat and Cow can eat meat.

Child-3: Cow cannot eat meat and Cow can eat meat, and, Cow can eat meat if and only if Fish can swim.

Child-1: T/F

Child-2: T/F

Child-3: T/F

Submit

Save

## Binary Search Steps Counter

1 point possible (graded, results hidden)

How many iterations of binary search are required to find **444** in **[54, 57, 60, 76, 182, 204, 240, 306, 435, 444, 487, 495, 555, 574, 603, 612, 896]**?

- ○ 8
- ○ 7
- ○ 5
- ○ 4

Submit

Save

## Valid Binary Search Tree

1 point possible (graded, results hidden)

Suppose that we have numbers between 1 and 100 in a binary search tree and we want to search for the number 89. Which of the following sequences could not be the sequence of nodes examined ?

- ○ [92, 48, 43, 50, 1, 89]
- ○ [18, 33, 99, 89]
- ○ [45, 49, 93, 91, 75, 82, 87, 90, 89]
- ○ [21, 33, 47, 78, 80, 86, 89]

Submit

Save

## Round and Round

1 point possible (graded, results hidden)

We have come upon a 'longRunning' method in our code. In order to check its lengthy execution time, we are calculating its iterations against different inputs.

Can you figure out the number of iterations it will take to execute the following input:

**[0, 1, 16, 17, 18, 19]**

```
function longRunningFunction(array) {
  for ( i = 0; i < length(array); i++ ) {
    idx = i
    for ( j = i + 1; j < length(array); j++ ) {
      if ( array[idx] > array[j] ) {
        idx = j
      }
    }
    swap( array[i], array[idx] )
  }
}
```

- ○ 16
- ○ 15
- ○ 21
- ○ 17

Submit

Save

## Truth Harmony

1 point possible (graded, results hidden)

Braden speaks truth in **83%** of cases and Fred in **33%** of cases. In what percentage of cases are they likely to contradict each other, talking about the same incident.

Submit

Save

## Mode, Mean, Median

1 point possible (graded, results hidden)

M = [82, 59, 72, 82, 'N']

What is the value of N if the mode, mean and median of the list M are equal to each other? Express your answer to the nearest whole number.

Note:

- The mode of a set of data values is the value that appears most often.
- The mean is the average of the numbers: a calculated "central" value of a set of numbers.
- Median is the middle number in a sorted list of numbers.

Submit

Save

## Novel Arrangement

1 point possible (graded, results hidden)

Anaya has three Urdu novels **(B, C, F)** and Four English novels **(A, G, D, E)**. She wants to arrange her novels in a way that following conditions must be met:

- No english novel can be placed immediate after another english novel.

- F must be placed earlier than D.

- G and D must be separated from each other by at least one novel.

- G must be placed immediately before or after B.

- B must be placed immediately after A, but not if C is placed earlier than A.

Choose the best sequence of novels:

- ○ D, B, G, F, E, C, A
- ○ C, F, G, E, D, B, A
- ○ A, F, E, C, G, B, D
- ○ E, C, A, F, G, B, D

Submit

Save

## Set Theory

1 point possible (graded, results hidden)

If

A = {{7}, 2, 3, 4, {3, 6}, 7}

B = {9, 2, {5}, 6, 7}

C = {{7}, 2, 4, 6, 7, 8, 9}

D = {8, {5}, 10, 3, 6}

Then the set ((B U A) - C) ∩ A is:

- ○ {}
- ○ {3, {3, 6}}
- ○ {2, {5}, 3, 6, 7, 8, 9, 10}
- ○ {4, {3, 6}}

Submit

Save

## Inheritance Code Snippet

1 point possible (graded, results hidden)

1 point possible (graded, results hidden)

What will be the output of this code snippet? (__init__ is constructor of class)

```
class A:
    def __init__(self):
        self.calc_i(720)

    def calc_i(self, i):
        self.i = 59 * i;

class B(A):
    def __init__(self):
        super().__init__()
        print("i from B is", self.i)

    def calc_i(self, i):
        self.i = 38 * i;

b = B()
```

You can select only one option.

- ○ 15248
- ○ 27360
- ○ 14951
- ○ 17396

Submit

Save

## Bubble Sort Integration

1 point possible (graded, results hidden)

What will be the condition of following array after 2 iteration(s) of Bubble Sort while sorting in ascending order

[47, 59, 71, 53, 8, 6, 62, 43, 48]

- ○ [62, 8, 48, 53, 47, 59, 6, 43, 71]
- ○ [47, 53, 8, 6, 59, 43, 48, 62, 71]
- ○ [48, 43, 59, 53, 62, 8, 47, 6, 71]
- ○ [6, 43, 59, 8, 62, 53, 47, 48, 71]

Submit

Save

## People In a Row

1 point possible (graded, results hidden)

In a cinema ticket line, A has 6 people ahead of it, while B has 24 people behind it. After they swap the positions, A has 14 people ahead. How many people are there in the line including A and B.

*Answer Format: An Integer, for example:* ***9***

Submit

Save

## Bank Loan

1 point possible (graded, results hidden)

As per agreement with a bank, a businessman had to refund a loan in some equal installments without interest. After paying '18' installments he found that '52.94' percent (approximately) of his loan was refunded. How many installments were there in the agreement?

- 38
- 28
- 40
- 34

Submit

Save

## Custom Series

1 point possible (graded, results hidden)

Assume we have a custom Series such that

First number is: 2

Second number is: 2

Every successing number is calculated as: $F_n$ = 2 x $F_{n-2}$ + 2 x $F_{n-1}$ e.g.

Third number is: $F_3$ = 2 x $F_1$ + 2 x $F_2$ = 2 x 2 + 2 x 2 = 8

What is 7th number in the series

- 56
- 152
- 416
- 1136

Submit

Answer submitted.

## LinkedList

1 point possible (graded, results hidden)

```
function foo(start) {
    if (start == NULL)
        return

    print(start.value)

    if (start.next != NULL)
            foo(start.next.next);

    print(start.value);
```

```
    }
```

What will be the output of the the following function if **start** pointing to **first node** of following linked list?
**[40, 55, 82, 44, 94, 92]**

- 40, 82, 94, 94, 82, 40
- 40, 55, 94, 92, 44
- 40, 44, 92, 55, 82
- 40, 55, 94, 82, 44, 92

Submit

Answer submitted.

## Age Selection

1 point possible (graded, results hidden)

### Table: A

| id | name | age |
|---|---|---|
| 26 | Zara | 47 |
| 52 | Abdullah | 53 |
| 84 | Fatima | 38 |
| 72 | Faran | 51 |
| 75 | Shahryar | 37 |
| 96 | Danish | 34 |

### Table: B

| id | name | age |
|---|---|---|
| 34 | Abdullah | 48 |
| 52 | Fatima | 38 |
| 48 | Zia | 58 |
| 124 | Mahnoor | 27 |
| 175 | Ayesha | 55 |

How many rows does the result of the following SQL query contains?

**SELECT** A.id
**FROM** A
**WHERE** A.age <> **ALL** (**SELECT** B.age **FROM** B **WHERE** B. name in ['Abdullah', 'Shahryar', 'Zia'])

- 6
- 5
- 8
- 7

Submit

Answer submitted.

Answer submitted.

## System Safe State

1 point possible (graded, results hidden)

A system has 17 magnetic tape drives and 5 processes : P1, P2, P3, P4, P5. The allocation of resources and the need for resources by the processes are described in the table. Which of the following is possible **safe state** of the system?

*Hint*: A system is in a **safe state** if there is a sequence in which all the processes can be executed without getting into a **deadlock**.

| processes | Need | Allocated |
|---|---|---|
| P1 | 14 | 1 |
| P2 | 17 | 0 |
| P3 | 17 | 0 |
| P4 | 16 | 3 |
| P5 | 12 | 1 |

- [4, 3, 5, 2, 1]
- [4, 5, 1, 3, 2]
- [5, 4, 2, 3, 1]
- [5, 1, 4, 2, 3]

Submit

Answer submitted.

## Imaginary String Printer

1 point possible (graded, results hidden)

```
function void imaginaryString()
  {
      arr_1 = ['O', 'W', 'K', 'I', 'H', 'Q', 'E', 'P', 'N', 'V', 'L', 'U']
      arr_2 = ['F', 'K', 'C', 'D', 'U', 'Z', 'R', 'L', 'O', 'X', 'H', 'A']

      arr_3 = get_common_values (arr_1, arr_2)
      arr_3 = sort_ascending (arr_3)

      a = [4, 0, 1, 2, 3]
      i = 0

      while (i < length(arr_3))
      {
              print arr_3[ a[i] ]
              i = i + 1
      }

  }
```

What will the imaginaryString() function print?

- OWKIHQEPNVLU
- UHKLO
- Raise Index Error
- 40123

HUOLK

FKCDUZRLOXHA

Submit

Answer submitted.

## Inheritance

1 point possible (graded, results hidden)

What should be the result of running the following pseudocode snippet?

```
class Class1:
    function function_1(self):
         print("a")

    function function_2(self):
         print("b")


class Class2:
    function function_1(self):
         print("c")

    function function_3(self):
         print("d")


class Class3:
    function function_2(self):
         print("e")

    function function_3(self):
         print("f")


class ClassA(Class3, Class1):
    function function_3(self):
         print("h")


class ClassB(Class2):
    function function_2(self):
         print("i")

    function function_3(self):
         print("j")


class ClassC(Class1):
    function function_2(self):
         print("k")

    function function_3(self):
         print("l")


ClassA().function_2()
ClassB().function_3()
ClassB().function_3()
```

ejj

Submit

Answer submitted.

## Find me If you can

1 point possible (graded, results hidden)

Find the missing operators:

### 5 ? 26 ? 9 ? 24 = 28080

```
Operators allowed:  +  -  *
```

Answer format: a+b-c*d

5*26*9*24

Submit

Answer submitted.

## Stacks and Queues

1 point possible (graded, results hidden)

There are two storage systems present, one is a stack and the other queue. The content of the stack is **[10, 5, 7, 0]** and the content of the queue is **[16, 25, 28, 18, 23, 21, 29, 27, 26, 15]** (the first item in both represent the first item stored). The number on each item represent the ID of item.

We have to balance these storage systems (move items between storages so that there are equal number of items in both). Keeping in mind the functionalities of stacks and queues, we have to balance them!

STACK STORAGE: *oldest* **[10, 5, 7, 0]** *newest*

QUEUE STORAGE: *oldest* **[16, 25, 28, 18, 23, 21, 29, 27, 26, 15]** *newest*

**What is the oldest item in queue storage after balancing the storages?**
You can select only one option.

- 15
- 28
- 18
- 5

Submit

Answer submitted.

## Balancing Parantheses

1 point possible (graded, results hidden)

A stack can be used to check whether the parentheses in an expression are balanced or not, by pushing an opening parenthesis to the stack and popping it whenever a closing parenthesis is encountered. What is the maximum possbile number of elements on the stack **at any one time** when evaluating: `( ( ( () ( () ) () () ) ) ) ()` ?

*Answer Format: An Integer, for example:* ***9***

5

Submit

Answer submitted.

## Average Waiting Time

1 point possible (graded, results hidden)

Given the following processes with their arrival and burst time given below, calculate the average waiting time using the First Come First Serve approach.

**Arrival time**: Time when the process is ready for its execution on the CPU.
**Burst time**: Time required by the process to complete its execution on the CPU.
**Waiting time**: Time spent by the process waiting for the CPU after its arrival.

| Process | Arrival Time | Burst Time |
|---|---|---|
| P1 | 2.0 | 11.0 |
| P2 | 4.0 | 6.0 |
| P3 | 5.0 | 9.0 |
| P4 | 6.0 | 14.0 |

3.25

Submit

Answer submitted.

## Travelkitties

1 point possible (graded, results hidden)

Travelkitties is a travel aggregator which allow users to book recreational trips using their app from all arround the world. You've been given a task to find out top 1 travel desitination (city) to help business team in making data driven decisions.
**Note: Travel desitination is arriving city of trip.**

user

| uid | name | age |
|---|---|---|
| 1 | Andy Williams | 41 |
| 2 | Andy Johnson | 29 |
| 3 | Joe Johnson | 52 |
| 4 | Joe Williams | 53 |
| 5 | John Johnson | 54 |

city

| cid | lat | lng | city | country_code |
|---|---|---|---|---|
| 1 | 34.95303 | -120.43572 | San Jose | US |
| 2 | 42.16808 | -88.42814 | Austin | US |
| 3 | 39.96097 | -75.60804 | San Antonio | US |
| 4 | 34.09668 | -117.71978 | Phoenix | US |
| 5 | 46.09273 | -88.64235 | Philadelphia | US |

trips

| tid | uid | origin_id | destination_id |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | 3 | 2 | 5 |
| 2 | 2 | 2 | 5 |
| 3 | 2 | 2 | 5 |
| 4 | 4 | 3 | 5 |
| 5 | 3 | 2 | 4 |
| 6 | 5 | 5 | 5 |
| 7 | 1 | 5 | 5 |

With given tables what would be output of following SQL query:

```
SELECT
  city_name
FROM (
  SELECT
        city AS city_name,
                    count(t.destination_id) AS trips
  FROM city AS c
  INNER JOIN trips AS t
  ON c.cid = t.destination_id
  GROUP BY city, t.destination_id
) AS ranked_trips
ORDER BY trips DESC
LIMIT 1;
```

You can select only one option.

- San Jose
- Phoenix
- Philadelphia
- Austin

Submit

Answer submitted.

## FIFO Page Fault

1 point possible (graded, results hidden)

Currently employed page replacement policy is **FIFO** and the capacity of storing **3** page frames at any instance of time. The page reference string is **XBXWBYWWDXYYYZC**. Can you tell the count of the page faults?

8

Submit

Answer submitted.

## XOR And XNOR

1 point possible (graded, results hidden)

Let **A : "11101111"** , **B=?**, If **{ A (Ex-or) B }** is a resultant string of **ALTERNATE ZEROES [ 01010101 ]** then:

- B is 10111010
- B is 00110111

B is 10110010

Submit

Answer submitted.

## Hash Clash

1 point possible (graded, results hidden)

*An array is used here to represent a Hash Table. Array index starts from **0** and ends at **size_of_array - 1***
Which slot would the number 43 hash to in the following Hash Table?

| -- | 42 | -- | 39 | -- | 49 | 36 | -- | -- | -- | 43 |
|---|---|---|---|---|---|---|---|---|---|---|

size_of_table = 11

The hash function is :

$hash\,(number)\,:\,number\,\%\,size\_of\_table$

For collision resolution use the following rehash function:

$new\_hash\_value\,:\,rehash\,(old\_hash\_value)$
$rehash\,(position)\,:\,(position+2)\,\%\,size\_of\_table$

7

Submit

Answer submitted.

## Identical Stacks

1 point possible (graded, results hidden)

Each row below are the stacks of water bottles with their respective heights(n)

**1. | 3 | 5 | 1 | 4 | 1 | 5 | 4 | 4 |**

**2. | 2 | 5 | 1 | 5 | 3 | 1 |**

**3. | 5 | 2 | 2 | 4 | 2 | 5 | 4 | 5 |**

The rightmost element shows the top of the stack. Adding up the heights of the bottles on a stack will give you the overall height of the stack. You can pop the bottles from each stack any number of times to change the height of the stack.

Determine the maximum height of each stack where all of the three stacks are equal in terms of height.

- 9
- 13
- 12
- 3

Submit

Save

## Employee Salaries

1 point possible (graded, results hidden)

**Table: employee_age**

| emp_id | age |
|---|---|
| 103 | 31 |
| 101 | 38 |
| 102 | 30 |
| 100 | 20 |

**Table: employee_salary**

| emp_id | salary |
|---|---|
| 104 | 35000 |
| 106 | 49000 |
| 102 | 54000 |
| 100 | 45000 |

With given tables what would be output of following SQL query:

```
SELECT
    MIN(eSal.salary)
FROM
    employee_age as eAge INNER JOIN employee_salary as eSal
ON
    eAge.emp_id = eSal.emp_id

    WHERE eAge.age > 20
GROUP BY eAge.emp_id
    HAVING MIN(eSal.salary) > 35000
```

- (•) 54000
- ( ) None
- ( ) 35000
- ( ) 49000

Submit

Answer submitted.

## Algorithm

1 point possible (graded, results hidden)

What is the output of the following code?

```
func min_jumps(arr[], start, end)
{
    if(start == end)
        return 0;

    int min = INT_MAX;  // Max value of int

    for(idx = 1; arr[start] >= idx AND end >= start + idx; idx++)
    {
        int jumps = min_jumps(arr, start + idx, end) + 1;
        if(min > jumps)
            min = jumps;
    }
    return min;
```

```
}

main()
{
    arr[] = [2, 2, 2, 1, 1, 1, 3, 2, 1, 1],
    ans = min_jumps(arr, 0, lenOfArr);
    print ans;
}
```

- ○ 11
- ○ 10
- ○ 6
- ○ 8

Submit

Save

## Process Scheduling

1 point possible (graded, results hidden)

```
Our CPU executes processes in bursts of 100ms and then calculates the next process to execute after each burst.
3 processes are fed into our CPU's process scheduler with the following attributes

Process A
Arrival Time: 0
Burst Time: 600

Process B
Arrival Time: 100
Burst Time: 1600

Process C
Arrival Time: 800
Burst Time: 1400

There are four main algorithms which our CPU uses to schedule processes:
FCFR: Firct Come First Serve
SJF:  Shortest Job First
SRTF: Shortest Remaining Time First
RR:   Round Robin

If we are using the RR algorithm to schedule processes, which will processes will have been completed after 3500 ms?
```

Answer as a comma separated list e.g. A,B or B,C,A

Submit

Save