encora

Technical Test

DEV - E

Date: 09/04/2025 City/State: Campinas/SP

Course: Computer Engineering Educational Institution: UNICAMP

Course Duration (in years): 5 Current Semester: 7 Graduation Year (expected): 2026

Availability to work: ☐ 20h ☒ 30h ☐ 40h Estimated Start Date: Once hired

Instructions: This test consists of 8 multiple choice questions, 1 algorithm implementation and 1 non-technical question. The algorithm is worth 60% of the total score. The non-technical question must be answered in Portuguese. You may use any blank space on this test as a draft. Use the table below to record your answers. Good luck!

Answer Sheet

	1	2	3	4	5	6	7	8
A								
B	X							
C				X	X			X
D		X	X			X	X	

Question 1

Given:

```
1.  2.import   java.util.Date;
3.  4.public class Example {
5.  6.public static void main(String args[]) {
7.  8.Date d1 = new Date (92, 11, 31);
9.      Date d2 = new Date (94, 11, 31);
10.     method(d1, d2);
11.   System.out.println("d1=" + d1.getYear() + "\nd2=" + d2.getYear());
12.   }
13. } public static void method(Date d1, Date d2) {
        d2.setYear(98);
        d1 = d2;
      }
```

What's the output?

A.
```
d1=92
d2=94
```

B.
```
d1=92
d2=98
```

C.
```
d1=98
d2=98
```

D.
```
d1=98
d2=94
```

Question 2

Given:

```
1.   //******************************
2.   // file A.java
3.   //******************************
4.   package a;
5.   public class A {
6.       private int x;
7.       protected int y;
8.       public int m1() {return x;}
9.   }
10.  //******************************
11.  // file B.java
12.  //******************************
13.  package b;
14.  import a.A;
15.  public class B extends A {
16.      private int z;
17.      public void m2(A a){
18.          z = x;
19.          z = y;
20.          z = a.m1();
21.      }
22.  }
```

Consider the following statements:

I. Line 18 is valid because B extends A

II. Line 19 is not valid because y is protected

III. Line 20 is valid because m1() is public

A. Only I and II are correct

B. Only I and III are correct

C. Only II and III are correct

D. Only III is correct

Question 3

What best defines an abstract class?

A. The class must not have method definitions.
B. The class must have a method definition returning nothing.
C. The class must have a constructor that takes no arguments.
D. The class cannot be instantiated as it is mainly for inheritance.

Question 4

Consider this small binary tree:

```
     14
    / \
   2   11
 / \   / \
1 3 10 30
      /   /
     7   40
```

What is the order of nodes visited using a post-order traversal?

```
A. 1 2 3 7 10 11 14 30 40
B. 1 2 3 14 7 10 11 40 30
C. 1 3 2 7 10 40 30 11 14
D. 14 2 1 3 11 10 7 30 40
```

Question 5

Consider the following statements:

I. A Binary tree is a tree data structure in which each node has at most two child nodes, usually distinguished as "left" and "right", and a tree with ***n*** nodes has exactly ***n***−1 branches which means its height is always n-1.
II. A Queue is a FIFO data structure, which means that the first element added to the queue will be the first one to be removed.
III. A Hash Map is a data structure in which, if there’s no collision among the keys, you can always find an element in O(1) time, even in the worst case.

A. Only I and II are correct
B. Only I and III are correct
C. Only II and III are correct
D. I, II and III are correct

Question 6

~~In the following code, assume that Queue is not thread-safe, there is more than one Producer~~ thread and more than one Consumer thread running and this program is crashing on runtime. In order to fix the code below how should you fill in lines (1), (2), (3) and (4)?

Global variables	
Queue q; (1)	
Producer thread	Consumer thread

runProducer(){
while(true){
    item = new item();
    (2)
    if (q is not full){
      q.enqueue(item);
      (3)
    }
    (4)
  }
}
	runConsumer(){
	while(true){
	    (2)
	    if (q is not empty){
	      item = q.dequeue();
	      (3)
	    }
	    (4)
	  }
	}

A. `(1) mutex m;`
`(2) m.lock();`
(3)
`(4) m.unlock();`

B. (1)
(2)
(3)
(4) `if(Consumer) sleep(1); else sleep(2);`

C. `(1) semaphore guard;`
`(2) wait(guard);`
(3)
`(4) signal(guard);`

D. Alternatives A and C are correct.

Question 7

~~Considering the following tables and data information, what would be the correct result of~~ the SQL command below?

Salesperson			
ID	Name	Age	Salary
1	Abe	61	140,000
2	Bob	34	44,000
5	Chris	34	40,000
7	Dan	41	52,000
8	Ken	57	115,000
11	Joe	38	38,000

Customer			
ID	Name	City	Industry_Type
4	Samsonic	Pleasant	G
6	Panasung	Oaktown	N
7	Samony	Jackson	N
9	Ornange	Hayward	G
8	Hepoul	Cupertino	I

Orders				
Number	Order_Date	cust_id	salesperson_id	Amount
10	8/2/2010	4	2	540
20	5/6/2012	9	7	150
30	3/12/2012	8	5	1,500
40	1/30/2013	4	8	1,800
50	7/14/2009	9	1	460
60	1/29/2012	7	2	2,400
70	2/3/2012	6	7	600
80	4/1/2013	8	2	2,300
90	3/2/2012	6	7	720

```
SELECT Salesperson.Name from Salesperson
WHERE Salesperson.ID NOT IN(
    SELECT Orders.salesperson_id FROM Orders
    INNER JOIN Customer ON Orders.cust_id = Customer.ID
    WHERE Customer.Name = 'Panasung')
AND Salesperson.ID IN
    (SELECT DISTINCT Orders.salesperson_id FROM Orders);
```

A. Bob
Chris
Ken

B. Abe
Bob
Ken

C. Abe
Bob
Chris
Ken
Joe

D. Abe
Bob
Chris
Ken

Question 8

Given this output on a Linux terminal:

```
$ cat linux_distributions.txt
Debian distribution
Ubuntu distribution derived from Debian
Fedora distribution
Red Hat Enterprise Linux distribution derived from Fedora
CentOS distribution derived from Fedora
MINIX and Linux operating system
```

What will be the correct result of the command below?

```
$ cat linux_distributions.txt | grep Fedora | sort
```

A.
```
Fedora distribution
Red Hat Enterprise Linux distribution derived from Fedora
CentOS distribution derived from Fedora
```

B.
```
CentOS derived distribution Fedora from
distribution Fedora
derived distribution Enterprise Fedora from Hat Linux Red
```

C.
```
CentOS distribution derived from Fedora
Fedora distribution
Red Hat Enterprise Linux distribution derived from Fedora
```

D.
```
distribution Fedora
derived distribution Enterprise Fedora from Hat Linux Red
CentOS derived distribution Fedora from
```

Subsets

Write the function getSubSets() to compute and return all subsets of a given set A, which has at most 4 elements.

Use the Set data structure to represent all sets and sub-sets of your solution. A Set is a collection that contains no duplicate elements and the order of elements is irrelevant. Consider the following interface defined for Set:

Method signature	Method description
`boolean add(Element e)`	Adds the specified element to this set if it is not already present (optional operation).
`boolean addAll(Set s)`	Adds all elements from s that are not already present in this set.
`boolean contains(Element e)`	Returns true if this set contains the specified element.
`boolean equals(Set s)`	Compares the specified set s with this set for equality.
`Iterator<Element> iterator()`	Returns an iterator over the elements in this set.
`boolean remove(Element e)`	Removes the specified element from this set if it is present (optional operation).
`int size()`	Returns the number of elements in this set (its cardinality).
`Element[] toArray()`	Returns an array containing all of the elements in this set.

Table: Set interface

Input example:

```
A=[1,2,3]
```

Output for the given example:

```
[[1,2,3], [1,2], [1,3], [2,3], [1], [2], [3], []]*
```

* this is the content of the Set which should be returned by the function.

Your proposed solution can be written in pseudo-code or any well-known language (C, C++, Java, etc) and you are free to implement any auxiliary functions. Besides, write down a comment to the main function explaining how your function will work like the one below.

```
/**
* The function below will ...
* - Obtain the input
* - Iterate over the elements
* …
* - Print the output and return ...
*/
```

Algorithm Solution

Language: Python

```
def recursaoNoSet(set, subSet):
  """
  Uses recursion to evaluate each subset of a set and adds them to a matrix.

  Parameters:
  set (list): A list of numeric values with no repetitions.
  subSet (list of lists): The subsets of the original set.

  Returns:
  list of lists: The matrix containing all subsets, excluding the empty subset.
  """
  if len(set) >= 1:
    for i in set:
      copia_set = set[:]
      copia_set.remove(i)
      recursaoNoSet(copia_set, subSet)
      if set not in subSet:
        subSet.append(set)

  return subSet

def getSubSets(set, subSet):
  """
  Receives a set and returns all its subsets.

  Parameters:
  set (list): A list of numeric values with no repetitions.
  subSet (list of lists): The subsets of the original set.

  Returns:
  list of lists: A matrix containing all subsets of the set.
  """

  subSetFinal = recursaoNoSet(set, subSet)
  subSetFinal.append([])

  return subSetFinal

def main():
  """
  The function below performs the following operations:
  - Receives an input and converts it into a list
  - Calls the getSubSets function, which:
    - Iterates over each element
    - Creates a copy of the original set
    - Removes the current element from the copy
    - Recursively calls itself
    - Adds the resulting subset to the matrix if it's not already present
    - Adds the empty subset
    - Returns the matrix of subsets
  - Sorts the matrix based on the length of each subset
  - Prints the matrix
  """
  print("Enter your set by typing its elements separated by spaces. Its subsets will be returned.")
  print("Exemple: 1 2 3 4")
  entrada = input("Set: ")
  set = list(map(int, entrada.split()))
  subSet = getSubSets(set, [])
  subSetSorted = sorted(subSet, key=lambda x: len(x), reverse=True)
  print(subSetSorted)
  return 0

if __name__ == "__main__":
  main()
```

Qual a disciplina que você mais gostou de cursar na faculdade e por quê? (Responder em português)

Durante a faculdade, a disciplina que mais me marcou foi Estrutura de Dados, cursada no segundo semestre. No início, enfrentei alguns desafios, por ainda estar no começo da faculdade e principalmente para me adaptar à linguagem C, além da intensidade das atividades, que exigiam de 8 a 12 horas de dedicação semanais fora da sala para resolver listas propostas.

Contudo, acredito que é pela repetição e esforço que se aprende algo, por isso estrutura de dados foi uma das matérias que mais me proporcionou aprendizado, tanto em hard skills (como estruturas de árvores, pontos flutuantes, pilhas...) quanto em soft skills, como trabalho em equipe, resiliência e adaptabilidade. Aprendi, acima de tudo, a abordar um problema de programação com profundidade: entendê-lo bem antes de agir, testar diferentes abordagens e não ter medo de errar ou explorar novas soluções. Essa disciplina foi fundamental para me inserir no mundo da programação e ainda me motiva a aprender cada vez.