This exam is worth 10 points, and consists of 25 questions. Each question poses 4 alternatives and has only one correct answer. Once discarded the worst answer, each correct answer earns 10/24 points, and each error deducts 10/72 points. You must answer on the answer sheet.

Let us consider these JavaScript programs:

```
1 // Program: client1.js
2 const zmq = require('zeromq')
3 const rq = zmq.socket('req')
4 rq.connect('tcp://127.0.0.1:8888')
5 rq.send('Hello')
6 rq.on('message', (msg) => {
7    console.log('Response: ' + msg)
8    rq.close()
9 })
```

```
1 // Program: server1.js
2 const zmq = require('zeromq')
3 const rp = zmq.socket('rep')
4 rp.bindSync('tcp://127.0.0.1:8888')
5 rp.on('message', (msg) => {
6    console.log('Request: ' + msg)
7    rp.send('World')
8 })
```

Note that these programs use the REQ-REP communication pattern. We plan to change both programs to use the DEALER-ROUTER pattern, providing the same functionality, and using a **minimal number of segments** *in the messages.*

1 *The changes to be applied on the client1.js program are:*

- **a** const rq = zmq.socket('router') // Line 3
- **b** const rq = zmq.socket('dealer') // Line 3
  rq.send([”,’Hello’]) // Line 5
- **c** const rq = zmq.socket('dealer') // Line 3
- **d** No other choice is correct.

2 *The changes to be applied on the server1.js program are:*

- **a** const rp = zmq.socket('dealer') // Line 3
- **b** const rp = zmq.socket('router') // Line 3
  rp.on('message', (who,msg) => { // Line 5
  rp.send([who,'World']) // Line 7
- **c** const rp = zmq.socket('router') // Line 3
  rp.on('message', (who,sep,msg) => { // Line 5
  rp.send([who,'World']) // Line 7
- **d** const rp = zmq.socket('router') // Line 3

3 *Let us consider a system composed of one client and two server processes. The servers bind and use a REP socket each one, while the client uses a REQ socket connected to both REP sockets. The client sends one request per second and never ends. In that system, if one of the servers is killed, the other two remaining processes get blocked and no other message is transmitted from them or delivered to them.*

Let us assume that we replace the REQ socket at the client process with a DEALER socket, and we adequately adapt the management of message segments in order to ensure that all sent and received messages will be delivered. With this new configuration, the three processes are restarted. What does it happen in this new scenario when one of the servers is killed?

- **a** The same as in the first system: communication gets blocked and no other request is delivered to its destination.
- **b** All client requests are delivered to the remaining server and no message is lost.
- **c** The client raises an exception that usually aborts its execution when that server is killed.
- **d** Communication does not block and the remaining server will still receive a half of the requests sent by the client.

4 *Which of the following stages of the software life cycle is not a part of the software deployment?*

- **a** Software design.
- **b** Software removal.
- **c** Software installation and configuration.
- **d** Software upgrade.

5 *Among these choices, which is the best tool to deploy a service in multiple host computers?*

- **a** Kubernetes
- **b** docker
- **c** docker-compose
- **d** Manual deployment

6 *In order to run a Docker container on a given host computer, we need …*

- **a** A hypervisor.
- **b** A guest operating system adequately installed and configured in the image to be run.
- **c** A host operating system different to that assumed in the image to be run.
- **d** No other choice is true.

Let us assume that image 'tsr-zmq' exists and has the contents and functionality explained in Unit 4 and Lab 3. Let us also assume that this Dockerfile (to be referred to as 'Dockerfile A', although its actual name is 'Dockerfile') has been saved in directory /home/user/docker/config:

```
FROM tsr-zmq
COPY ./tsr.js tsr.js
RUN mkdir broker
WORKDIR broker
COPY ./broker.js mybroker.js
EXPOSE 9998 9999
CMD node mybroker 9998 9999
```

7 *Let us consider Dockerfile A. What command should be used in order to create an image called 'broker' if the current working directory of our shell is* `/home/user/docker`*?*

- **a** docker commit config broker
- **b** docker run broker
- **c** docker build -t broker config
- **d** docker build -t broker

8 *Let us consider Dockerfile A. Where should the image 'tsr-zmq' and files 'tsr.js' and 'broker.js' be placed in order to successfully use that Dockerfile for creating a new image?*

- **a** All the required items (tsr-zmq, tsr.js, and broker.js) should be placed at `/home/user/docker/config`
- **b** The tsr-zmq image should be at the local repository or be present at hub.docker.com, while the files should be at `/home/user/docker/config`
- **c** All required items (tsr-zmq, tsr.js, and broker.js) should be somewhere at the local host, since the docker engine will search and find them without any problem.
- **d** No other choice is correct.

9 *Let us consider Dockerfile A. Which command will run the 'bash' interpreter in a container that uses the 'broker1' image generated using that Dockerfile A?*

- **a** No docker command may achieve that goal.
- **b** docker run bash
- **c** docker run -i -t broker1 bash
- **d** docker-compose up bash

10 *Let us consider that this Dockerfile has been used for creating an image called 'broker2':*

```
FROM tsr-zmq
COPY ./tsr.js tsr.js
RUN mkdir broker
WORKDIR broker
COPY ./broker.js mybroker.js
EXPOSE 9998 9999
ENTRYPOINT ["/usr/bin/node","mybroker"]
CMD ["9998","9999"]
```

Can we use 'broker2' to run the 'bash' shell in a container?

- **a** No, since the broker2 image cannot be built, because a Dockerfile cannot combine ENTRYPOINT and CMD.
- **b** By default, no, since the program that runs in containers generated from the broker2 image is `node` mybroker
- **c** Yes, with this command:
  `docker run -i -t broker2 bash`
- **d** No, since when we combine ENTRYPOINT and CMD in the same Dockerfile only the last of them in the file is considered, and in this example CMD has invalid values.

11 *The first session of Lab 3 starts with a* *<u>manual deployment</u>* *of a broker-worker-client system. In that system, the broker is started first, and the other two components need to know* *<u>the IP address of the broker container</u>*. *The command (or set of commands) that provides that information is"*

- **a** docker images
- **b** docker inspect id
  Assuming that the broker container id has been found previously.
- **c** docker-compose up
- **d** No other choice is correct.

12 *Let us consider that this docker-compose.yml is in directory* `/home/user/docker`

```
version: '2'
   services:
     svca:
       image: imga
       links:
         - svcb
       environment:
         - B_HOST=svcb
     svcb:
       image: imgb
       links:
         - svcc
       environment:
         - C_HOST=svcc
       expose:
         - "9999"
     svcc:
       image: imgc
       expose:
         - "9999"
```

Please, choose the FALSE sentence about the service to be deployed using that file.

- **a** The components in that service will start in this order: svcc, svcb, svca.
- **b** That service can be deployed, but its components svcc and svcb will listen to same host port (9999) at the same time raising a conflict.
- **c** We may deploy one instance of components svcc and svcb and six instances of component svca using this command at /home/user/docker:
  `docker-compose up -d --scale svca=6`
- **d** Once the service has been deployed, we may stop and remove all its started containers using this command at /home/user/docker:
  `docker-compose down`

13 *This is a skeleton of the docker-compose.yml file to be used in the second half of the first session of Lab 3 to automate the deployment of a BCW system:*

```
version: '2'
services:
  cli:
    image: client
    build: ./client/
    links:
     - W
    environment:
     - BROKER_HOST=X
     - BROKER_PORT=9998
  wor:
    image: worker
    build: ./worker/
    links:
     - Y
    environment:
     - BROKER_HOST=Z
     - BROKER_PORT=9999
  bro:
    image: broker
    build: ./broker/
    expose:
     - "9998"
     - "9999"
```

The needed values for W,X,Y and Z to fill such a docker-compose.yml file with minimum contents to correctly manage that deployment (i.e., to guarantee an appropriate starting order and an adequate dependency resolution) are:

a X=bro, Z=bro, and the links clauses are not needed and must be removed, so W and Y don´t need any value.

b W=wor, X=wor, while the links and environment clauses in wor must be removed, so Y and Z don´t need any value.

c No links or environment clauses are needed to automate this deployment. All those sections may be removed, and the resulting deployment will behave correctly. Therefore, no value is needed for W, X, Y and Z.

d W=bro, X=bro, Y=bro, Z=bro.

14 *The second session of Lab 3 introduces a new logger component in the CBW system. Which other CBW components interact with this new logger?*

a Clients and workers send their trace messages to the logger.

b Only the broker sends its trace messages to the logger.

c Clients, workers and broker send their trace messages to the logger.

d Only the workers send their trace messages to the logger.

15 *The second session of Lab 3 proposes the deployment of another type of client (an external client) that will run in another computer, different to the host computer where docker and docker-compose manage the CBW containers. What is needed in the configuration of the broker component in order to enable that external client?*

a A 'ports:' clause is needed in the 'bro' section of the docker-compose.yml file, in order to map port 9998 of the broker container to port 9998 at the host computer.

b A 'ports:' clause is needed in the Dockerfile of the broker image, in order to map port 9998 of the broker container to port 9998 at the host computer.

c Nothing special is needed, we only need to find out the IP of the broker component, using docker ps and docker inspect to this end.

d An 'EXPORTS' instruction is needed in the Dockerfile of the broker image, in order to state which port of the broker container should be used by external clients.

16 *The docker-compose.yml used in the third session of Lab 3 contains a line like this:*

```
image: docker.io/bitnami/mariadb:11.1
```

What is the consequence of replacing the '11.1' part of that line with 'latest' in that deployment?

- **a** That, perhaps, in two years the resulting 'docker-compose.yml' file will become useless.
- **b** None. The resulting deployment behaves and will behave correctly, independently of that change.
- **c** An error since the word 'latest' cannot be a part of a docker image name.
- **d** An error since 'latest' cannot be a part of a system pathname.

17 *If we compare active and passive replication, the passive model is the preferred replication model when operation updates only modify a small part of the service state because":*

- **a** The passive model must send those updates to the back-up replicas and the latter should apply them, while in the active model no update transfer is needed.
- **b** When those updates are small, back-up replicas don´t need them.
- **c** When those updates are small, they may be transferred asynchronously and with no effect on the resulting inter-replica consistency.
- **d** All other choices are correct.

18 *Regarding connectivity failures, when a primary partition model is assumed then:*

- **a** All system nodes go on, and the resulting system consistency is very relaxed.
- **b** System availability is ensured.
- **c** Processes in minor subgroups stop.
- **d** Services use passive replication.

19 *This is the docker-compose.yml file used in the last session of Lab 3:*

```
version: '2'
services:
  mariadb:
    image: docker.io/bitnami/mariadb:11.1
    volumes:
      - 'mariadb_data:/bitnami/mariadb'
    environment:
      - ALLOW_EMPTY_PASSWORD=yes
      - MARIADB_USER=bn_wordpress
      - MARIADB_DATABASE=bitnami_wordpress
  wordpress:
    image: docker.io/bitnami/wordpress:6
    ports:
      - '80:8080'
      - '443:8443'
    volumes:
      - 'wordpress_data:/bitnami/wordpress'
    depends_on:
      - mariadb
    environment:
      - ALLOW_EMPTY_PASSWORD=yes
      - WORDPRESS_DATABASE_HOST=mariadb
      - WORDPRESS_DATABASE_PORT_NUMBER=3306
      - WORDPRESS_DATABASE_USER=bn_wordpress
      - WORDPRESS_DATABASE_NAME=bitnami_wordpress
volumes:
  mariadb_data:
    driver: local
  wordpress_data:
    driver: local
```

How can we manually deploy, with the assistance of the docker-compose command, the mariadb and wordpress components in two different host computers?

- **a** No system deployment is possible, since we don´t know the contents of the Dockerfiles that generated the images to be used.
- **b** No deployment is possible in two computers, since this docker-compose.yml file did not work as intended with a single host computer.
- **c** That deployment is possible without applying any update to this file.
- **d** Splitting the file in two, one per component, and adding a 'ports:' section to the mariadb part and a good value to WORDPRESS_DATABASE_HOST on the other.

20 *If we compare the causal and cache consistency models, which of them is more relaxed than the other?*

- **a** Causal
- **b** Cache
- **c** Regarding their degree of relaxation, they are equivalent.
- **d** Their degree of relaxation cannot be compared.

21 *If we consider the CAP theorem, which of these consistency models may be respected when availability is needed in a partitioned system?*

- **a** Strict
- **b** Causal
- **c** Sequential
- **d** None

22 *Which of these alternatives is a correct difference between the multi-master and the passive replication models?*

- **a** The multi-master model may use a different processing replica (i.e., the master) per request, while the passive model always uses the same primary.
- **b** In the passive model each request is only forwarded to the primary replica, while in the multi-master model each request is broadcast by the client to every replica.
- **c** The multi-master model can handle the arbitrary failure model, while the passive model cannot.
- **d** All other choices are true.

23 *Node.js provides its 'cluster' module to:*

- **a** Run a given program in a set of processes, sharing some resources (e.g., the access to a socket) to implement a locally scalable service.
- **b** All other choices are true.
- **c** Start multiple threads of execution in a single process.
- **d** Deploy a Node.js service in a cluster of computers.

24 *MongoDB uses this replication model:*

- **a** Active replication
- **b** Passive replication
- **c** Multi-master replication
- **d** No replication

25 *To enhance its scalability, MongoDB uses:*

- **a** Readable back-up replicas.
- **b** Horizontal database partitioning.
- **c** All other choices are correct.
- **d** Configuration caching in its mongos components.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

A

DNI NIE PASSPORT

0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

ETSINF - NIST

Second Partial - 23/01/2024

Surnames ....

Name ....

Mark like this Like this do not mark

NOT ERASING, correct with concealer

Second Partial

	a	b	c	d
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				