MAD I QUIZ I REVISION MAY24

Q4. Let `L = {'a', 'b', 'c', 'd', 'A', 'B', 'C', 'D', '0', '1', ' '}` be a complete character set (i.e., only these characters can be used to represent text in the document). If a document that uses fixed encoding for all characters is created using the character set L and has a disk size of 2 Kilobytes, the number of characters in the documents would be_____. [Take 1 Byte = 8 bits, 1 KB = 1000 Bytes, 1 MB = 1000 Kilobytes and so on.]

[MCQ: 4.5 points]

A. 2000
B. 4000
C. 8000
D. 16000

Answer: B

Storage: Bytes
BW: Mbps

11 characters 3 bits, 2^3= 8, 4 bits, 2^4= 16 characters (a, b) 0, 1> 1 bit, no characters= diff comb= 2^bits each comb >store character

Size = characters x bits required to store each character

Characters= Size/(bits required to store each character)
= 2 x 1000 x 8/ 4 (1 byte = 8 bit)
= 4000 characters

ASCII - 7 bit old
ASCII - 8 bit new
UCS (8 bits)
UCS-2 (2 bytes i.e. 16 bits) 1 byte= 8 bits
UCS-4 (4 bytes)
UTF-8 8 bits

Q8. A mobile client starts from and is cruising away continuously at 60 kmph from the network tower whose network range is 40 km and bandwidth is 120 Mbps. How much data (in Gigabytes) will be consumed by the client who is continuously using the entire bandwidth before completely moving out of the network?

[Take 1 Byte = 8 bits, 1 KB = 1000 Bytes, 1 MB = 1000 Kilobytes and so on.]

[Consider the speed of light in air to be 3 x 10^8 m/sec.]

[MCQ: 3 points]

A. 24
B. 28.8
C. 36
D. 288

Answer: C

| …….>.....40km no network, no BW, no data consumption

Tower

Bandwidth= no of request x size of request

Unit Balancing:
Mbps => Mbps x time(s) => Mb/s x s = Mb/8 = MB
Speed = distance/time
t= d/s
t= 40km/60kmph= 40x hr/60 = 2x 3600s/3 = 2400 s

Data= BW x time = 120 Mbps x 2400 s = 12 x 24 Gb /8
= 36 GB

b- bits, BW Mbps
B- Bytes, storage
1GB= 1000 MB

Q14. Consider the following flask application:

[MSQ: 3 Points]

```
from flask import Flask, request
app = Flask(__name__)
@app.route('/get_value')
def get_value():
val1 = request.args.get("val1")
return "The value is "+ val1
app.run(port = 5000, debug = True)
```

Which of the following route(s) will return `The value is 10` in the browser?

A. `http://127.0.0.1:5000?val1=10`
B. `http://127.0.0.1:5000/get_value?10`
C. `http://127.0.0.1:5000/get_value?val1=10`
D. `http://127.0.0.1:5000/get_value?val1=10&val2=10`

Answer: C,D

```
val1 = request.args.get("val1","20")
```

If val1 then val 1 value, by default it will take val1=20

Q15. Consider the following Python code snippet “code.py”.

[MCQ: 3 points]

Filename: code.py

```
import sys
from jinja2 import Template

vars = sys.argv
myTech = {'mad 1': 'backend', 'mad 2': 'frontend'}

temp = Template("This course focuses on {{var}} development.")

if vars[2] == 'mad 1':
    print(temp.render(myTech['mad 1']))
elif vars[2] == 'mad 2':
    print(temp.render(myTech['mad 2']))
else:
    print("Mention the course name specifically.")
```

What will be the output on the terminal for input: `python code.py mad 1` ?

A.

```
This course focuses on frontend development.
```

B.

```
This course focuses on backend development.
```

C.

```
Mention the course name specifically.
```

D.

```
IndexError: list index out of range
```

Answer: C

Terminal > spaces, consider diff elements

1. var= data
2. Spaces in argument

17. Consider the following HTML document.

[MSQ: 4.5 points]

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
    <style>
        span {
            background-color: yellow;
            color: red;
            ===code1===
        }
        .class {
            border: 2px solid purple;
            color:blue ;
        }
        #id {
            background-color: aqua;
            color: red;
        }
    </style>

</head>
<body>
    <span class="class" id="id">SPAN 1</span>
    <span class="class">SPAN 2</span>
    <span>SPAN 3</span>
</body>
</html>
```

Which of the following is/are correct rendered output for given ===code1===?

A.

If ===code1=== is replaced by:

```
display:inline-block;
border: 2px solid purple;
```

Output:

B.

If ===code1=== is replaced by:

```
display:block;
color: blue;
```

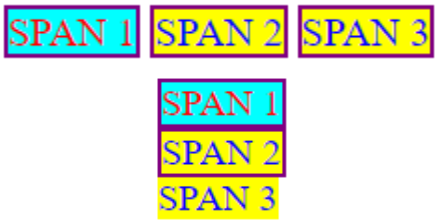
Output:

C.

If ===code1=== is replaced by:

```
color: blue;
border: 2px solid purple;
```

Output :



D.

If ===code1=== is left blank:

Output:

Answer: A,C, D

Size = characters x bits required to store each character

ASCII - 7 bit old
ASCII - 8 bit new
UCS (8 bits)
UCS-2 (2 bytes i.e. 16 bits) 1 byte= 8 bits
UCS-4 (4 bytes)
UTF-8 8 bits

Bandwidth= no of request x size of request

Unit Balancing:
Mbps => Mbps x time(s) => Mb/s x s = Mb/8 = MB
Speed = distance/time

Styling> !important >Inline > Internal > External
Selectors> !important >ID> class> element same selector(class same styling> latest styling)

String > substitute, throw error if var> data, var< data not an issue , $
safe_substitute > print that var as it is ($p)
Jinja > render, var >data blank, var<data, no issue, {{}}

Pass info> 1. var= data {{name}}, name= Ravi
2. Var = data (data mentioned)
3. Directly passing data(dict)

Display: 1.inline>width of content only, elements one beside each other
2. block> whole width, assign width
3. Inline-block: inline+block, width not mentioned, will take content width, allow elements one beside each other, assign width

MVC> Model = database, V = view, C= controller> business logic
response> header(meta information), body (content)
GET> response>header, body, HEAD similar to get response body

curl -X method URL sending request from terminal, browser, hit url> get resource

python -m http.server python server

- *1xx informational response* – the request was received, continuing process
- *2xx successful* – the request was successfully received, understood, and accepted
- *3xx redirection* – further action needs to be taken in order to complete the request
- *4xx client error* – the request contains bad syntax or cannot be fulfilled
- *5xx server error* – the server failed to fulfil an apparently valid request