

Q1) Put a circle around the symbol of the best answer for each of the following

1. Which of the following features must be available in a programming language to be suitable for writing embedded systems
  - a) Provide constructs for low-level control over hardware.
  - b) Use floating point representation.
  - c) Have features to analyze data.
  - d) All of the above.
2. An orthogonal programming language
  - a) Has a relatively small set of primitive constructs that can be combined in a relatively small number of ways to get the desired result.
  - b) Has a fewer number of exceptions because every possible combination is legal.
  - c) Is good with respect to reliability.
  - d) All of the above.
3. Which of the following features a programming language need to have in order to be good with respect to reliability
  - a) Data types.
  - b) Support for abstraction.
  - c) Exception handling.
  - d) All of the above.
4. Having a small number of manageable features and constructs, makes a programming language good with respect to
  - a) Readability.
  - b) Writability.
  - c) Reliability.
  - d) All of the above.
5. A language with too many operators and special symbols is
  - a) Good with respect to writability
  - b) Bad with respect to reliability
  - c) Bad with respect to readability
  - d) All of the above.
6. Java is partly compiled and partly interpreted (hybrid implementation) makes it
  - a) Good for writing portable programs.
  - b) Good for writing efficient programs.
  - c) Good for reducing the cost of training.
  - d) All of the above.
7. Classes, inheritance and polymorphism designed in programming languages as a
  - a) The Von-Neumann architecture.
  - b) Programming methodologies.
  - c) People preferences.
  - d) None of the above.

Q2)

A) Which programming language is better with respect to reliability Java or C++? Explain why?

Java

List features or constructs makes it so (Explain why)

Different question same concept and answer (Compare between Java and C++ with respect to reliability).

1- Because it checks the index of the array if it is out of bounds or not.

2- C++ has pointer which are not reliable.

3- Java also has a strong type checking while C++ has a weak type checking.

B) Which programming language is better with respect to the cost of execution Java or C++? Give at least 2 reasons.

C++

List features or constructs makes it so (Give at least 2 reasons)?

1- It does not check for proper indexing, it allows the user to go out of bounds in the array.

2- Compiler translate to machine language directly.

C) Which programming language is better with respect to portability Java or C++? Explain why?

Java

List features or constructs makes it so (Explain why)?

It will translate to intermediate language before translating to machine language.

D) Discuss two advantages of C++ compared to Java.

1- Compiler translate to machine language directly good for cost of execution

2- Pointers have a better flexibility in terms of writability as it allows us to access address in memory

E) Discuss how pointers in C++ affects

1)writability

Good for writability since it allows us to use (arrays, objects etc) address in memory.

2)reliability

Bad for reliability since it does not check for proper indexing, and has a weak type checking.

F) Why were modular languages developed?

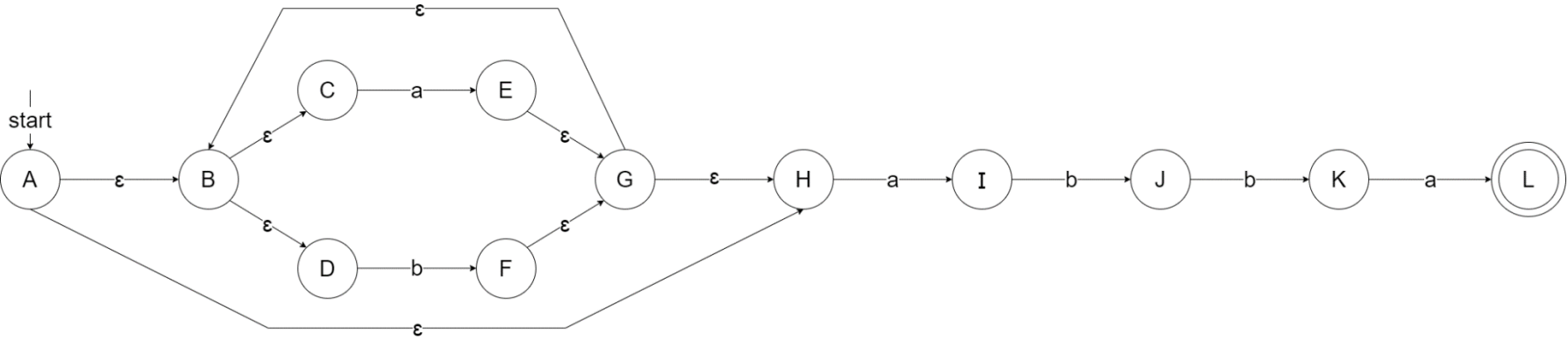
1- To support huge applications.

2- Divide and conquer approach for development.

3- Data abstraction.

G) Discuss how Von Neumann architecture influenced the design of modern programming languages.

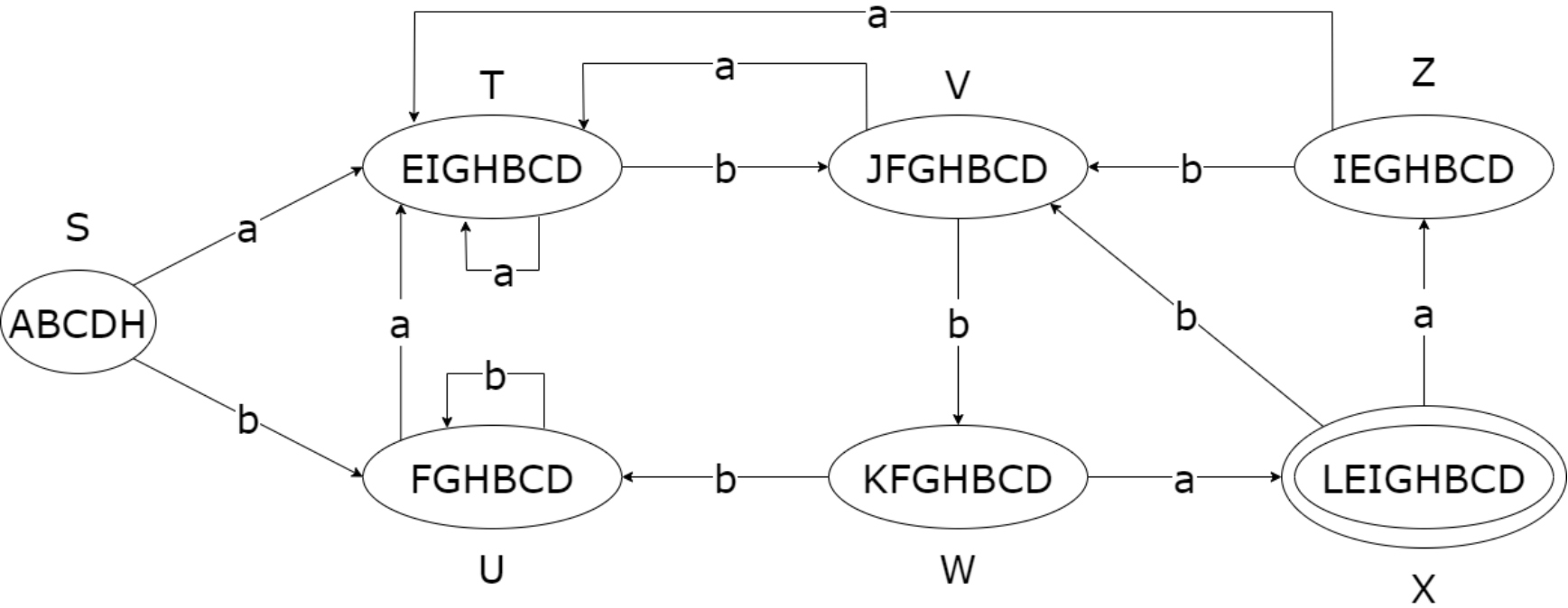
Q3) Consider the following NFA.



A) Describe the language accepted by this NFA using a regular expression.

$(a+b)^*abba$

B) Convert the above NFA into a DFA.



C) Represent the DFA using a table.

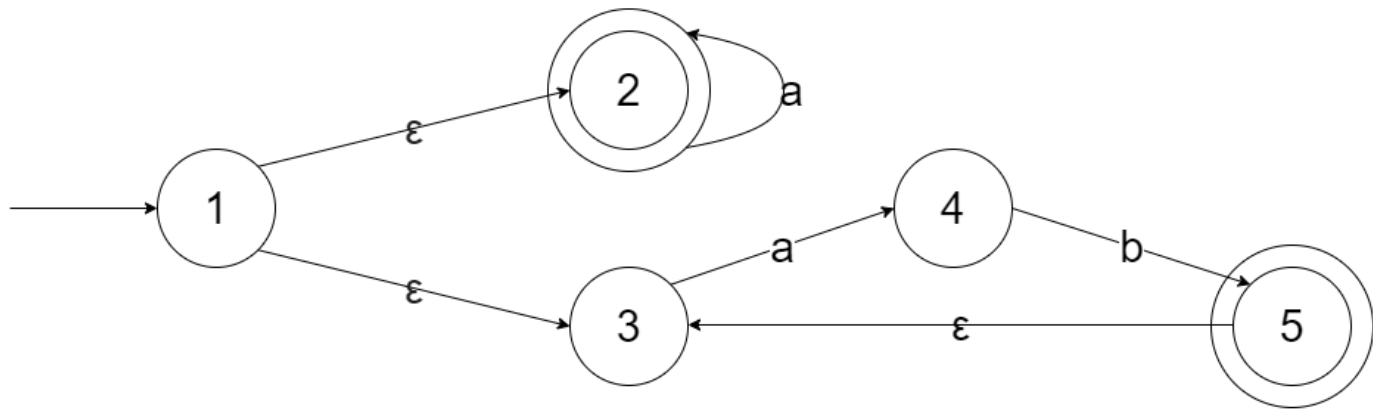
| STATE | a | b |
|-------|---|---|
| S     | T | U |
| T     | T | V |
| U     | T | U |
| V     | T | W |
| W     | X | U |
| X     | Z | V |
| Z     | T | V |

D) Write an algorithm that uses the above table to decide if a string is acceptable by the DFA or not.

```
i=0;
state = 0
while(input[i]){
state=A[state,input[i++]];
}

If(state == X)
    Return accept;
Else
    Return reject;
```

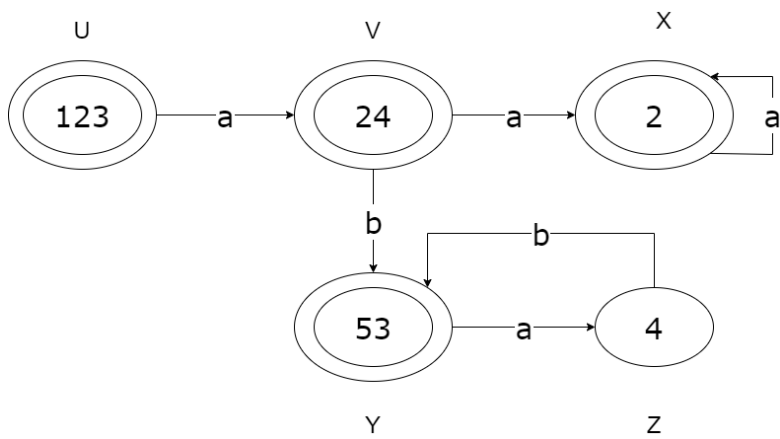
Q4) Consider the following NFA



A) Describe the language recognized by the NFA using regular expression.

$a^* + (ab)^+$

B) Convert the following NFA into a DFA.



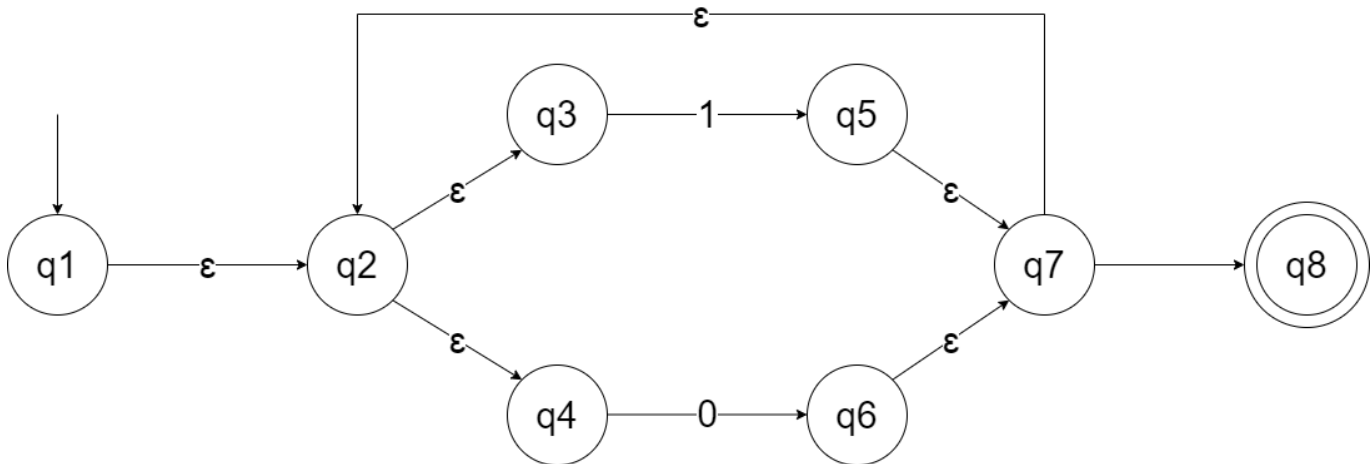
C) Represent the DFA using a table.

| State | A | B |
|-------|---|---|
| U     | V | - |
| V     | X | Y |
| Y     | Z | - |
| X     | X | - |
| Z     | - | Y |

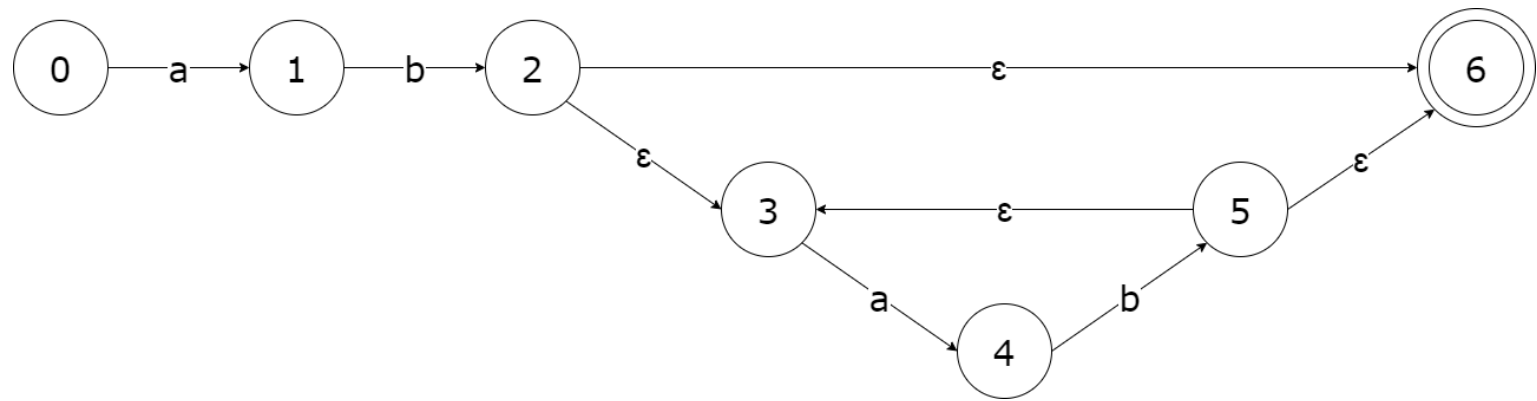
D) Write an algorithm that takes a string over the alphabet and display “accept” if the string is acceptable by the DFA, or “reject” if it is not.

```
i=0;
state=0;
while(input[i]){
    state=A[state,input[i++]];
}
If(state == U || state == V || state == X || state == Y)
    Return “accept”;
Else
    Return “reject”;
```

E) Convert the regular expression  $(0+1)^+$  over the alphabet  $\{0,1\}$  into an NFA.



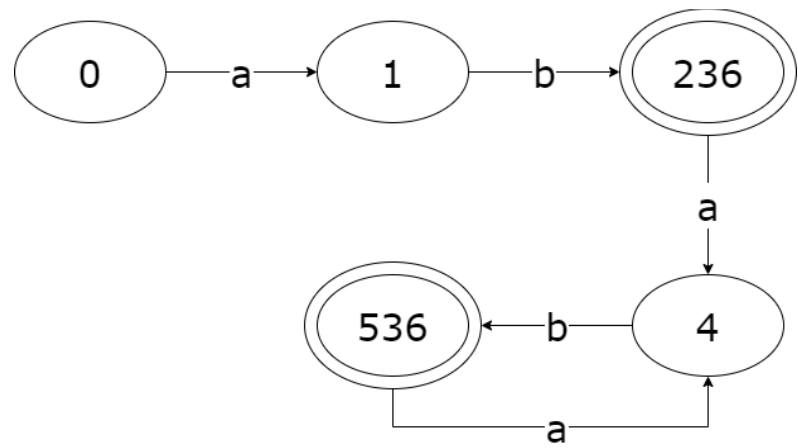
Q5) Consider the following NFA.



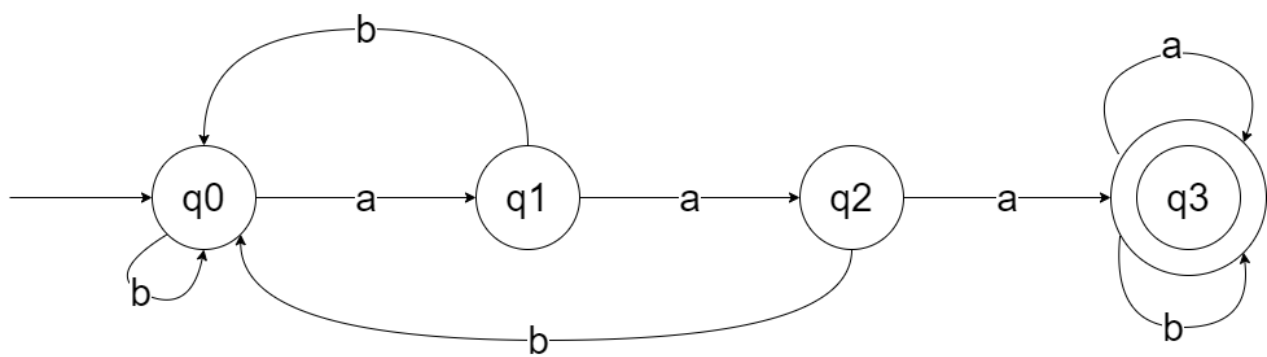
A) Describe the language accepted by the above NFA using regular expression.

(ab)<sup>+</sup>

B) Convert the above NFA into a DFA.



Q6) Consider the following DFA over the alphabet  $\Sigma = \{a,b\}$



A) In your own words describe the language accepted by the above DFA.

Accepts any string as long it has 3 consecutive a's.

B) Write a regular expression that describes the language accepted by the above DFA.

(a+b)<sup>\*</sup>aaa(a+b)<sup>\*</sup>

C) Represent the DFA using a table.

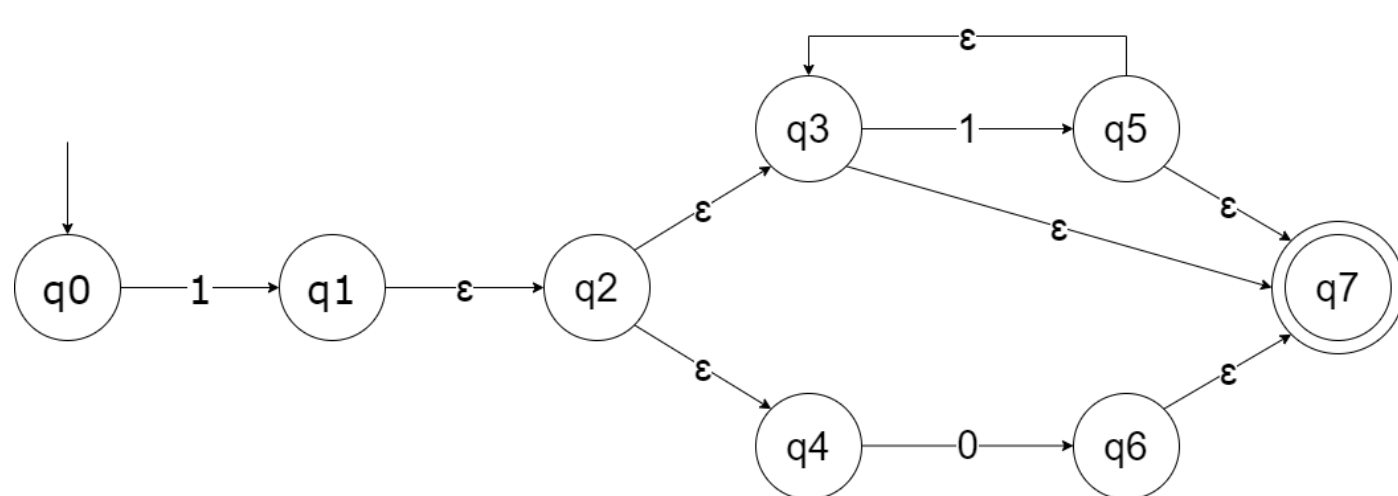
| State | A  | B  |
|-------|----|----|
| q0    | q1 | q0 |
| q1    | q2 | q0 |
| q2    | q3 | q0 |
| q3    | q3 | q3 |

D) Write an algorithm that uses the above table to decide if a string is acceptable by the DFA.

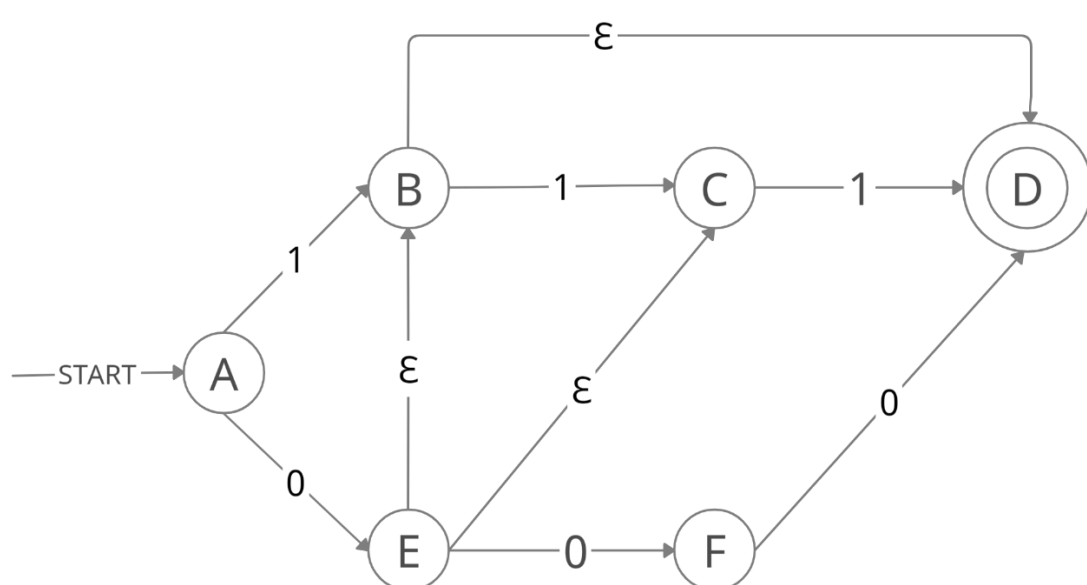
```
i=0;
state=0;
while(input[i]){
    state=A[state,input[i++]];
}
If(state == q3)
    Return "accept";
Else
    Return "reject";
```

Q7)

A) Draw an NFA that recognizes the language denoted by the regular expression  $1(0+1^*)$ .



B) Write a regular expression that describes the language recognized by the NFA.



$\{000+111+011+01+1+0\}$

C) Represent the above NFA using a table.

| state | 0   | 1   | $\epsilon$ |
|-------|-----|-----|------------|
| A     | {E} | {B} | -          |
| B     | -   | {C} | {D}        |
| E     | {F} | -   | {B,C}      |
| C     | -   | {D} | -          |
| F     | {D} | -   | -          |
| D     | -   | -   | -          |

D) Write an algorithm that uses the above table to determine if a given string is legal.

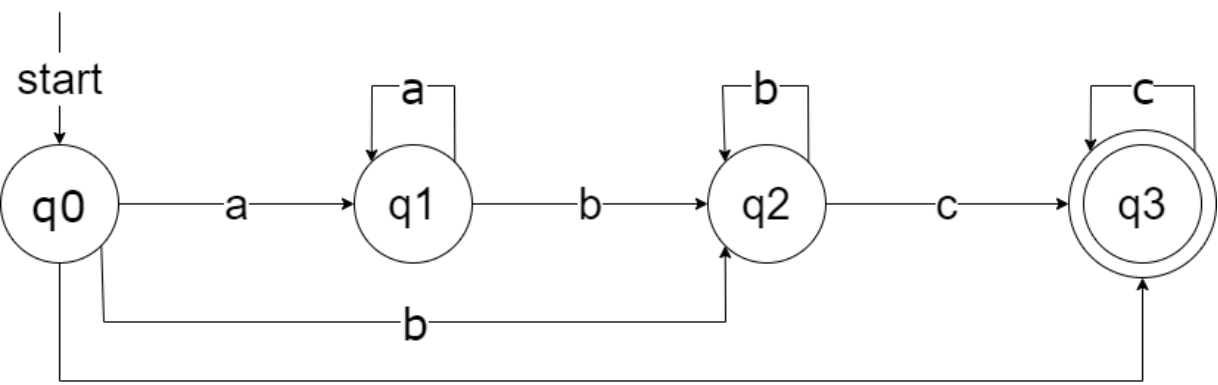
```

i=0;
state = 0
while(input[i]){
    state=A[state, input[ i++ ]];
}

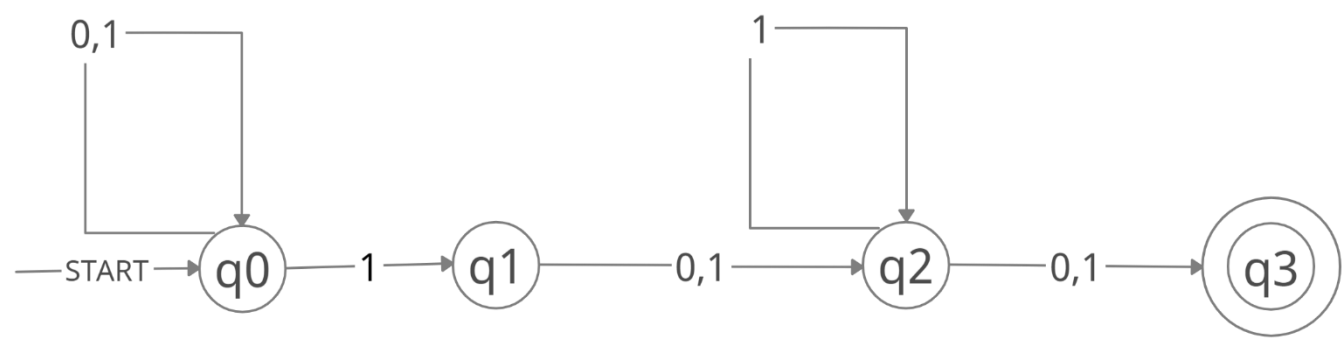
If(state == D)
    Return accept;
Else
    Return reject;
  
```

Q8)

A) Design a DFA that accepts a string if it belongs to the language  $\{a^p b^q c^r \mid p, q, r \geq 0\}$



B) Consider the finite automaton in the following figure.



1. What is the set of reachable states for the input string 0011?

{ q0 , q1, q2 }

2. Is the string 0011 acceptable? Why?

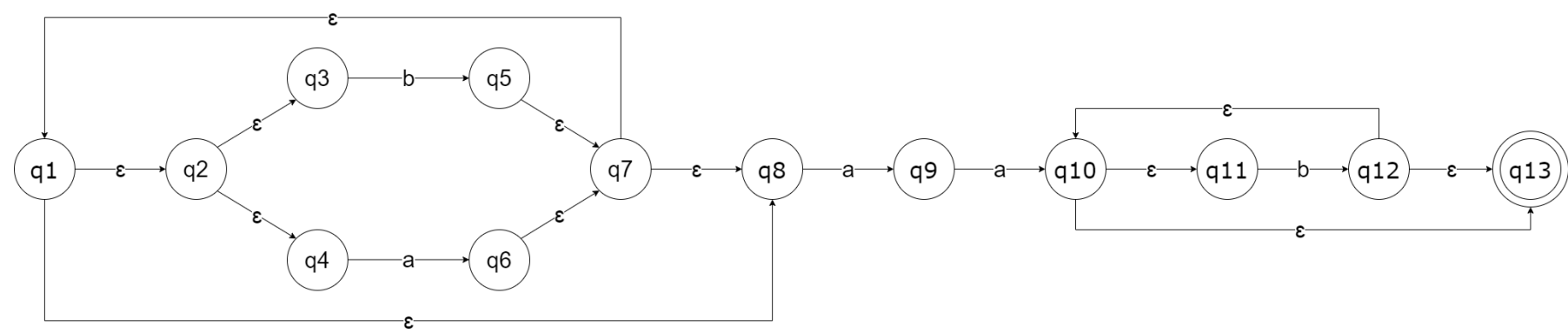
No , it did not reach the final state which is q3.

C) Represent the above finite automaton using a table.

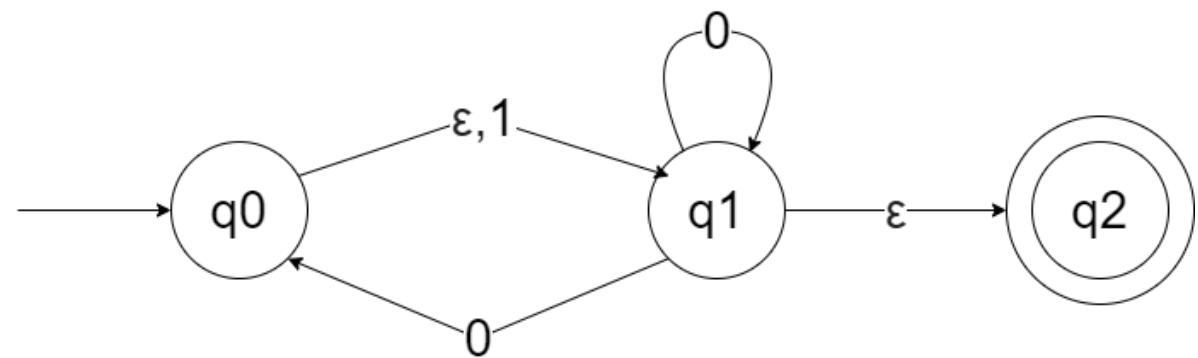
| state | 0    | 1       |
|-------|------|---------|
| q0    | q0   | {q0,q1} |
| q1    | q2   | q2      |
| q2    | q3   | {q3,q2} |
| q3    | ---- | ----    |

Q9)

A) Convert the regular expression below into an NFA.  
 $(a+b)^*aab^*$



B) Consider the following NFA



1. Represent the NFA using a table.

| State | 0      | 1  | $\epsilon$ |
|-------|--------|----|------------|
| q0    | -      | q1 | q1         |
| q1    | q0, q1 | -  | q2         |
| q2    | -      | -  | -          |

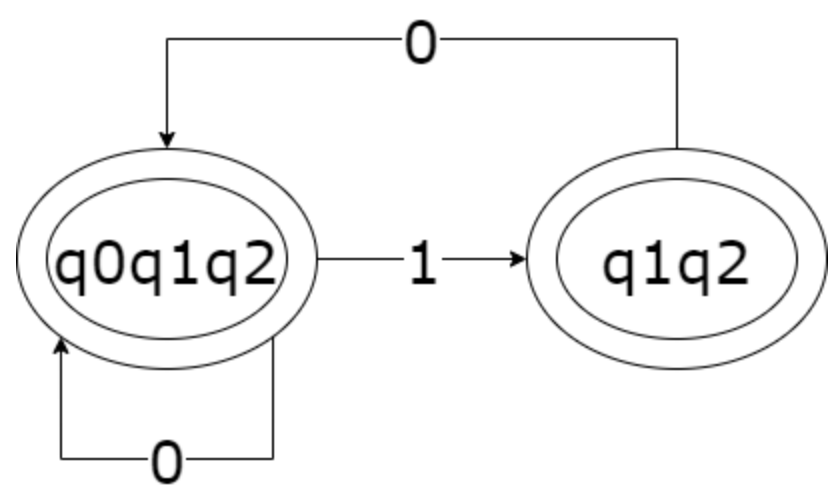
2. What is the shortest string that would be rejected by the above NFA.

11 will be rejected, because q1 has no transition for the second 1.

3. Write an algorithm that accepts a string if it is accepted by the above NFA.

```
i=0;
state=0;
while(input[i]){
    state=A[state,input[i++]];
}
If(state == q2)
    Return "accept";
Else
    Return "reject";
```

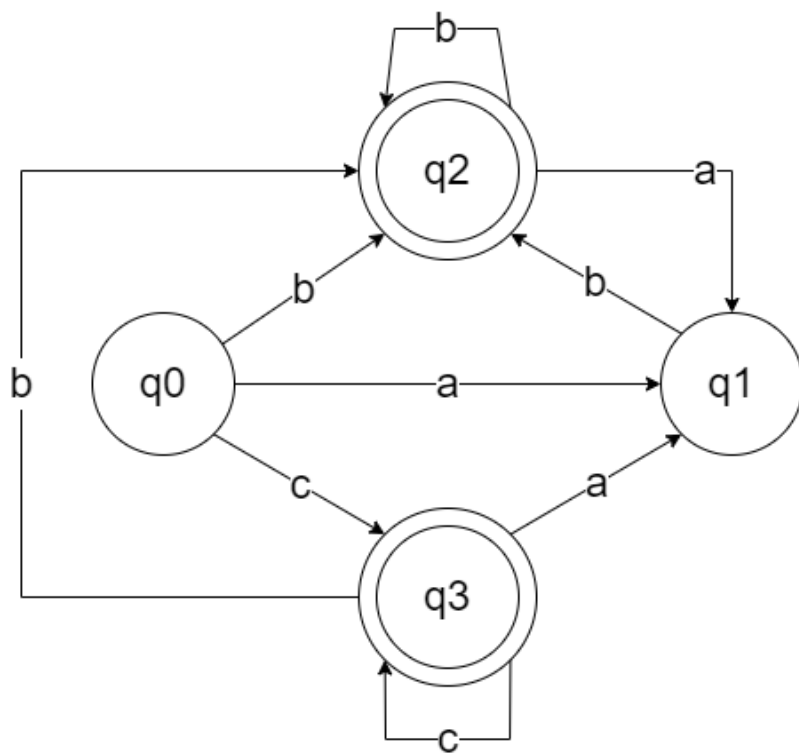
4. Convert the above NFA into a DFA.





Q10) Consider the alphabet  $S=\{a,b,c\}$ .

- A) Design a DFA that accepts a string in which every 'a' is followed by a 'b' but no 'b' is followed by 'c'. for example, the DFA should accept "cabbab" but not "abc".

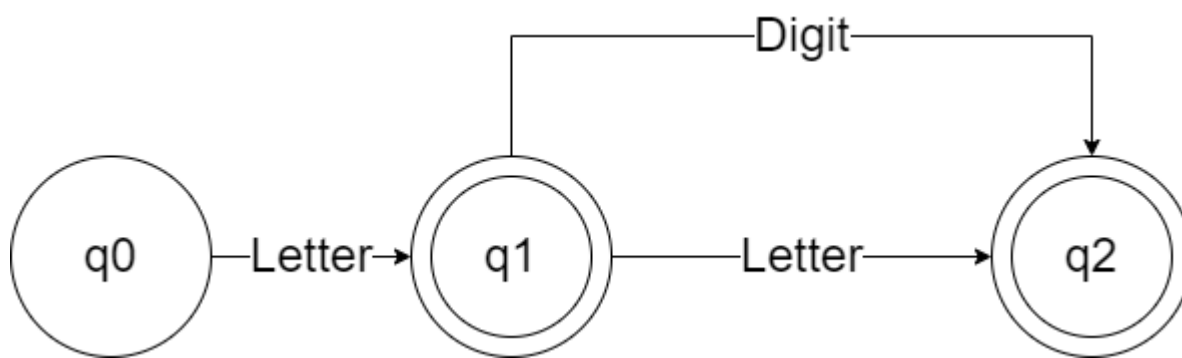


- B) Write a regular expression that describes the language accepted by the above DFA.

$C^*B^*(ab)^*$

Q11)

- A) Design a DFA that accepts a string containing letter and digits where the string consists of at most two characters the first character must be a letter (A-Z), while the second may be a letter or digit (0-9).



- B) Write a regular expression that defines the token class identifier where an identifier can consist of at most two characters the first character must be a letter (A-Z), while the second may be a letter or digit (0-9).

$\text{Letter} + (\text{Letter} + \text{Digit})$

Q12) consider the following CFG.

$E \rightarrow E - \text{int} \mid E / \text{int} \mid \text{int}$

A) Which operation has the highest precedence?

Left most operation

B) What is the association of the “-” operation?

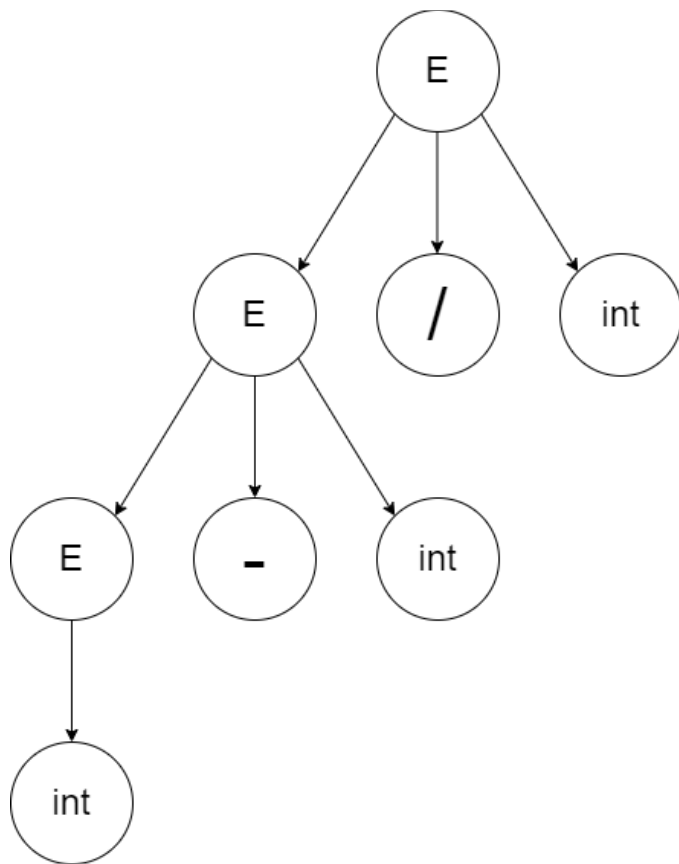
Left to right

C) What is the association of the “/” operation?

Left to right

D) Show that the string int-int/int belongs to the language of the above CFG (is valid)?

$E \rightarrow E / \text{int} \rightarrow E - \text{int} / \text{int} \rightarrow \text{int} - \text{int} / \text{int}$



E) Write a CFG that describes all mathematical expression that can be formed using only \* and + . Make sure that \* has a higher precedence over + and the associativity of both operation is left to right.

$E \rightarrow E^{\wedge} + \text{int} \mid E^{\wedge}$

$E^{\wedge} \rightarrow E^{\wedge} * \text{int} \mid E \mid \text{int}$

Q13) Write a CFG describing a language over the alphabet  $\Sigma=\{a,b\}$ , containing all string with

A) Equal number of a's and b's.

$S \rightarrow aSbS \mid bSaS \mid \epsilon$

B) The number of a's is double the number of b's.

$S \rightarrow aSb \mid bSa \mid a$

Q14)

- A) Assuming that a float number consists of an optional integer part and an optional fraction part, write a regular expression that describes the language of all float numbers, note that the string "14.15", ".15" and "15." Are all legal but the string "." Is not.
- B) Write a CFG that describes the language of float numbers as described above.
- C) Consider a language defined over  $S=[0,1]$  where a string belongs to the language if the number of 1's is twice the number of 0. What would you use to describe this language regular expression or context free grammar? Justify your answer.