Q1] Given content - free grammar is:

S → NP VP	DET → a / the
NP → DP	NN → car / bike
DP → DET NN	VP → is VRB
	VRB → running / stopping

Shift Reduce Parser.

Stack	Input	Action
$	a car is running $	shift
$ a	car is running $	Reduce (Det → a)
$ Det	car is running $	shift
$ Det car	is running $	Reduce (NN - car)
$ Det NN	is running $	Reduce (DP - DETNN)
$ DP	is running $	shift
$ DP is	running $	shift.
$ DP is running.	$	Reduce (VP → DP)
$ NP is running	$	Reduce (VRB → running)
$ NP is VRB	$	Reduce (VP → is VRB)
$ NP VP	$	Reduce (S → NP VP)
$ S		ACCEPT

END

classmate

* The Parser successfully accepts the input string "a car is running" which means that it can be generated by the given grammar.

PYTHON CODE

```
import nltk

grammar = nltk.CFG.fromString("""
    S → NP VP
    NP → PP
    PP → DET NN
    Det → 'a' | 'the'
    NN → 'car' | 'bike'
    VP → 'is' VRB
    VRB → 'running' | 'stopping'
""")

# Generate all possible sentences

sentences = []
for length in range(1, 6):
    for tree in grammar.generate(n = 1000, depth = length):
        sentence = ' '.join(tree.leaves())
        sentence.append(sentence)
print(sentences)
```

Q2] Lets Calculate probability of playing Golf.

$$P(No) = 4/11 \quad ; \quad P(Yes) = 7/11$$

We need $X = \{$ Temperature = ~~Normal~~ Hot,
Humidity = Normal,
Windy = True $\}$

Now calculate, $P\left(\frac{\text{Temp} = \text{Hot}}{\text{Yes}}\right) = 1/7$

$$P\left(\frac{\text{Humidity} = \text{Normal}}{\text{Yes}}\right) = 4/7$$

$$P\left(\frac{\text{Windy} = \text{bool}}{\text{Yes}}\right) = 2/7$$

$$P\left(\frac{\text{Temp} = \text{Hot}}{\text{No}}\right) = 2/4$$

$$P\left(\frac{\text{Humidity} = \text{Normal}}{\text{No}}\right) = 1/4$$

$$P\left(\frac{\text{Windy} = \text{False}}{\text{No}}\right) = 2/4$$

Final Probabilities:

$$P\left(\frac{Yes}{x}\right) = P\left(\frac{Hot}{Yes}\right) \times P\left(\frac{Normal}{Yes}\right) \times P\left(\frac{True}{Yes}\right)$$

$$= 4/7 \times 2/7 \times 1/7$$

$$= \underline{\underline{8/7^3}} \bullet$$

$$P\left(\frac{No}{x}\right) = P\left(\frac{Hot}{No}\right) \times P\left(\frac{Normal}{No}\right) \times P\left(\frac{True}{No}\right)$$

$$= 2/4 \times 1/4 \times 2/4$$

$$= \underline{\underline{1/4}}$$

$$P\left(\frac{Yes}{x}\right) = \frac{8/7^3}{8/7^3 + 1/4} = \frac{0.023}{0.023 + 0.25} = \frac{0.023}{0.273} = \underline{\underline{0.084}}$$

$$P\left(\frac{No}{x}\right) = \frac{1/4}{1/4 + 8/7^3} = \frac{0.25}{0.273} = \underline{\underline{0.9157}}$$

∴ For the given conditions X,

the game, GOLF is "NOT PLAYED"

Q3] CFG :

S → NP VP
NP → Det NN
Det → a / an / the
NN → child / adult
VP → AUX VRB
AUX → is / was
VRB → crying / sleeping.

Stack	Input	Action
$	a child is crying $	shift
$ a	child is crying $	Reduce (Det → a)
$ Det	child is crying $	shift
$ Det child	is crying $	Reduce (NN → child)
$ Det NN	is crying $	Reduce (NP → Det NN)
$ NP	is crying $	shift
$ NP is	crying $	Reduce (AUX → is)
$ NP AUX	crying $	shift
$ NP AUX crying	$	Reduce (VRB → crying)
$ NP AUX VRB	$	Reduce (VP → AUX VRB)
$ NP VP	$	Reduce (S → NP VP)
$ S		ACCEPT

⟹ Parser successfully accepts "a child is crying"

Code - PYTHON

```
import nltk

grammar = nltk.CFG.fromString("""
    S --> NP VP
    NP --> Det NN
    Det --> 'a' | 'an' | 'the'
    NN --> 'child' | 'adult'
    VP --> AUX VRB
    AUX --> 'is' | 'was'
    VRB --> 'crying' | 'sleeping'

sr_parser = nltk.ShiftReduceParser(grammar)

sentence = "a child is crying"
tokens = nltk.word_tokenize(sentence)

try:
    for tree in sr_parse.parse(tokens):
        print(tree)
    print("Sentence is grammatically correct")

except ValueError:
    print("Sentence is grammatically incorrect")
```

classmate

Q4] $P(No) = 4/10$

$P(Yes) = 6/10$

Now, calculate probability of likelyhood of evidence · Given ~~child, young, low~~.

$X = \{$ type of family = single parent
age group = young
income_status = high. $\}$.

$P(\text{Single Parent}/Yes) = 1/6$

$P(\text{Single Parent}/No) = 1/4$

$P(\text{Young}/Yes) = 2/6$

$P(\text{Young}/No) = 1/4$

$P(\text{Low}/Yes) = 1/6$; $P(\text{High}/Yes) = 2/2$

$P(\text{Low}/No) = 4/4$; $P(\text{High}/No) = 0$

$$P(\text{Yes}/X) = P\left(\frac{\text{Single Parent}}{\text{Yes}}\right) * P\left(\frac{\text{Young}}{\text{Yes}}\right) * P\left(\frac{\text{High}}{\text{Yes}}\right)$$

$$= 1/6 \times 2/6 \times 1 = 2/36 = 1/18$$

$$P(\text{No}/X) = P\left(\frac{\text{Single Parent}}{\text{No}}\right) * P\left(\frac{\text{Young}}{\text{No}}\right) * P\left(\frac{\text{High}}{\text{No}}\right)$$

$$= 1/4 \times 1/4 \times 0$$

$$= 0$$

∴ Final Probabilities are : ~~1/3 / 1/3 + 0~~

$$P(\text{Yes}/X) = \frac{1/18}{1/18 + 0} = 1$$

$$P(\text{No}/X) = \frac{0}{1/18 + 0} = 0$$

∴ For the given conditions X they "WILL BUY A CAR"

Q5]

```
S(tokens):
        NP(tokens)
        VP(tokens)
```

```
NP(tokens):
        DP(tokens)
```

```
DP(tokens):
        Det(tokens)
        NN(tokens)
```

```
Det(tokens):
        if tokens[0] == 'a' or tokens[0] == 'the':
            tokens.pop(0)

        else:
            raise ValueError("Error, got" + tokens[0])
```

```
NN(tokens):
        if tokens[0] == 'car' or tokens[0] == 'bike':
            tokens.pop(0)

        else:
            raise ValueError("Error, got" + tokens[0])
```

CLASSMATE

```
def VBR (tokens):
    if tokens[0] == 'running' or tokens[0] == 'stopping'
        tokens.pop (0)
    else:
        raise ValueError ("Expected running or stopping,
                          but got " + tokens[0])
```