def main (nltk):

nltk.download ()

print("Type in your AI computer question:\n")

xa = input()

xb = (xa.lower())

xc = (xb.split())

print(xc)

stop\_words = []

a = 0

a = ("1) What is Artificial Intelligence?")

aa = 0

print()

aa = ("Artificial Intelligence is an area of \ncomputer science that emphasizes the \ncreation of an intelligent machine \nthat works and reacts like humans.\n")

word\_list = aa

filtered\_list = ["artificial", "intelligence", "ai", "intelligent", "computer", "human", "humans"]

xi = 0

for xi in xc:

if xi in filtered\_list:

print(aa)

print()

b = 0

b = ("2) What is an artificial intelligence Neural Networks?\n")

bb = 0

bb = ("Artificial intelligence Neural Networks can model \nmathematically the way biological brain \nworks, allowing the machine to think and \nlearn the same way the humans do- making them\n capable of recognizing things like speech, \nobjects and animals like we do.")

word\_list1 = bb

filtered\_list1 = ["artificial", "intelligence", "ai", "neural networks", "neural network", "mathematically", "brain", "think", "humans", "recognize", "recognizing", "speech"]

for xi in xc:

if xi in filtered\_list1:

print(bb)

print()

c = 0

c = ("3) What are the various areas where AI (Artificial Intelligence) can be used?\n")

cc = 0

cc = ("Artificial Intelligence can be used \nin many areas like Computing, Speech \nrecognition, Bio-informatics, Humanoid \nrobot, Computer software, Space and \nAeronautics’s etc.\n")

word\_list = cc

filtered\_list2 = ["artificial", "intelligence", "ai", "computing", "computers", "speech recognition", "bio-informatics", "robots", "computers", "computer", "software", "space", "aeoronautics"]

for xi in xc:

if xi in filtered\_list2:

print(cc)

print()

d = 0

d = ("4) Which is not commonly used programming language for AI?\n")

dd = 0

dd = ("Perl language is not commonly used \nprogramming language for AI.")

word\_list = dd

filtered\_list3 = ["perl", "language", "programming", "ai"]

for xi in xc:

if xi in filtered\_list3:

print(dd)

print()

e = 0

e = ("5) What is Prolog in AI?")

ee = 0

ee = ("In AI, Prolog is a programming language \nbased on logic.")

print()

word\_list = ee

filtered\_list4 = ["programing", "language", "prologue", "logic"]

for xi in xc:

if xi in filtered\_list4:

print(ee)

print()

f = 0

f = ("6) Give an explanation on the difference between strong AI and weak AI?\n")

ff = 0

ff = ("Strong AI makes strong claims that computers can be \nmade to think on a level equal to humans \nwhile weak AI simply predicts that some \nfeatures that are resembling to human \nintelligence can be incorporated to computer \nto make it more useful tools.")

word\_list = ff

filtered\_list5 = ["strong", "ai", "computers", "think", "weak", "humans", "predicts", "incorporated", "intellegence"]

for xi in xc:

if xi in filtered\_list5:

print(ff)

print()

g = 0

g = ("7) Mention the difference between statistical AI and Classical AI ?\n")

gg = 0

gg = ("Statistical AI is more concerned with \n“inductive” thought like given a set of pattern, induce the trend etc. While, classical AI, on the other hand, is more concerned with “ deductive” thought given as a set of constraints, deduce a conclusion etc.")

word\_list = gg

filtered\_list6 = ["statistics", "statistical", "inductive", "pattern", "classical", "ai", "deductive", "conclusion"]

for xi in xc:

if xi in filtered\_list6:

print(gg)

print()

h = 0

h = ("8) What is alternate, artificial, compound and natural key?")

hh = 0

hh = ("Alternate Key: Excluding primary keys all candidate keys\nare known as Alternate Keys.\nArtificial Key: If no obvious key either\n stands alone or \ncompound is available, then the last \nresort is to, simply create a key, \nby assigning a number to each record \nor occurrence. This is known as artificial key.\nCompound Key: When there is no single data element that\nuniquely defines the occurrence within\n a construct, then integrating multiple elements to create a \nunique identifier for the construct is known as Compound Key.\nNatural Key: Natural key is one of the data element \nthat is stored within a construct, \nand which is utilized as the primary key.")

word\_list = hh

filtered\_list7 = ["alternate", "primary key", "primary keys", "artificial", "key", "keys"]

for xi in xc:

if xi in filtered\_list7:

print(hh)

print()

i = 0

i = ("9) What does a production rule consist of?")

ii = 0

ii = ("The production rule comprises of a set of rule and a \nsequence of steps.")

word\_list = ii

filtered\_list8 = ["production rule", "set", "sequence"]

for xi in xc:

if xi in filtered\_list8:

print(ii)

print()

j = 0

j = ("10) Which search method takes less memory?")

jj = 0

jj = ("The depth first search method takes \nless memory.")

word\_list = jj

filtered\_list9 = ["depth first search", "memory"]

for xi in xc:

if xi in filtered\_list9:

print(jj)

print()

k = 0

k = ("11) Which is the best way to go for Game playing problem?")

kk = 0

kk = ("Heuristic approach is the best way to go for game playing \nproblem, as it will use the technique based on \nintelligent guesswork. For example, \nChess between humans and computers as it will use brute force computation, looking at hundreds of thousands of positions.")

word\_list = kk

filtered\_list10 = ["heuristic", "game playing", "chess", "brute", "force", "humans", "computers"]

for xi in xc:

if xi in filtered\_list10:

print(kk)

l = 0

l = ("12) A\* algorithm is based on which search method?")

ll = 0

ll = ("A\* algorithm is based on best first search method, as it \ngives an idea of optimization and quick choose of path, and all characteristics lie in A\* algorithm.")

word\_list = ll

filtered\_list11 = ["algorithm", "first", "search", "method", "optimisation", "quick", "choose", "path", "characteristics"]

for xi in xc:

if xi in filtered\_list11:

print(ll)

print()

m = 0

m = ("13) What does a hybrid Bayesian network contain?")

mm = 0

mm = ("A hybrid Bayesian network contains both a discrete and \ncontinuous variables.")

word\_list = mm

filtered\_list12 = ["hybrid", "Bayesian", "network", "discrete", "continuous", "variable", "variables"]

for xi in xc:

if xi in filtered\_list12:

print(mm)

n = 0

n = ("14) What is agent in artificial intelligence?")

nn = 0

nn = ("Anything perceives its environment by sensors and acts \nupon an environment by effectors are known as Agent. Agent includes Robots, Programs, and Humans etc.")

word\_list = nn

filtered\_list13 = ["percives", "environment", "sensors", "effectors", "agent", "robots", "robot", "program", "programs", "humans"]

for xi in xc:

if xi in filtered\_list13:

print(nn)

o = 0

o = ("15) What does Partial order or planning involve?")

oo = 0

oo = ("In partial order planning , rather than \nsearching over possible situation it \ninvolves searching over the space of \npossible plans. The idea is to \nconstruct a plan piece by piece.")

word\_list = oo

filtered\_list14 = ["partial", "order", "planning", "searching", "plan", "piece"]

for xi in xc:

if xi in filtered\_list14:

print(oo)

p = 0

p = ("16) What are the two different kinds of steps that we can take in constructing a plan?")

pp = 0

pp = ("The two different kinds of steps \nthat we can take in constructing a plan:\n1.Add an operator (action). \n2.Add an ordering constraint between \noperators.")

word\_list = pp

filtered\_list15 = ["operator", "operators", "action", "constraint", "steps"]

for xi in xc:

if xi in filtered\_list15:

print(pp)

q = 0

q = ("17) Which property is considered as not a desirable property of a logical rule-based system?")

qq = 0

qq = ("“Attachment” is considered as not a desirable \nproperty of a logical rule-based system.")

word\_list = qq

filtered\_list16 = ["attachment", "logic", "logical", "rule-based"]

for xi in xc:

if xi in filtered\_list16:

print(qq)

r = 0

r = ("18) What is Neural Network in Artificial Intelligence?")

rr = 0

rr = ("In artificial intelligence, neural network is an emulation of a biological neural system, which receives the data, process the data and gives the output based on the algorithm and empirical data.\n")

word\_list = rr

filtered\_list17 = ["artificial", "intelligence", "ai", "biological", "neural", "system", "data", "process", "output", "algorithm", "empirical"]

for xi in xc:

if xi in filtered\_list17:

print(rr)

s = 0

s = ("19) When an algorithm is considered completed?")

ss = 0

ss = ("An algorithm is said completed when it \nterminates with a solution when one exists.")

word\_list = ss

filtered\_list18 = ["algorithm", "completed", "terminates", "solution", "exists"]

for xi in xc:

if xi in filtered\_list18:

print(ss)

t = 0

t = ("20) What is a heuristic function?")

tt = 0

tt = ("A heuristic function ranks alternatives, \nin search algorithms, at each branching step based on the available information to decide which branch to follow.")

word\_list = tt

filtered\_list19 = ["heuristic", "function", "ranks", "ranking", "alternatives", "algorithms", "branching", "branch", "information", "follow"]

for xi in xc:

if xi in filtered\_list19:

print(tt)

u = 0

u = ("21) What is the function of the third component of the planning system?\n")

uu = 0

uu = ("In a planning system, the function of the third \ncomponent is to detect when a solution to \nproblem has been found.")

word\_list = uu

filtered\_list20 = ["system", "fuction", "component", "planning"]

for xi in xc:

if xi in filtered\_list20:

print(uu)

print()

v = 0

v = ("22) What is “Generality” in AI ?")

vv = 0

vv = ("Generality is the measure of ease with which the method can be adapted to different domains of application.\n")

word\_list = vv

filtered\_list21 = ["generality", "measure", "ease", "domains", "domain"]

for xi in xc:

if xi in filtered\_list21:

print(vv)

print()

w = 0

w = ("23) What is a top-down parser?")

ww = 0

ww = ("A top-down parser begins by \nhypothesizing a sentence and \nsuccessively predicting lower \nlevel constituents until individual \npre-terminal symbols are written.")

word\_list = ww

filtered\_list22 = ["top-down", "top-down", "parser", "hypothesis", "sentence", "predicting", "predict", "low", "lower", "level", "individual", "pre-terminal", "symbols"]

for xi in xc:

if xi in filtered\_list22:

print(ww)

print()

x = 0

x == ("24) Mention the difference between breadth first search and best first search in artificial intelligence?")

xx = 0

xx = ("These are the two strategies which are \nquite similar. In best first search, we expand the nodes in accordance with the evaluation function. While, in breadth first search a node is expanded in accordance to the cost function of the parent node.")

word\_list = xx

filtered\_list23 = ["strategy", "strategies", "best first search", "nodes", "evaluaton", "function", "breadth first search", "cost", "parent"]

for xi in xc:

if xi in filtered\_list23:

print(xx)

print()

y = 0

y = ("25) What are frames and scripts in “Artificial Intelligence”?")

yy = 0

yy = ("Frames are a variant of semantic \nnetworks which is one of the popular \nways of presenting non-procedural knowledge \nin an expert system. A frame which is an \nartificial data structure is used to \ndivide knowledge into substructure by \nrepresenting “stereotyped situations'. Scripts are \nsimilar to frames, except the values that \nfill the slots must be ordered. \nScripts are used in natural language \nunderstanding systems to organize a \nknowledge base in terms of the situation \nthat the system should understand.")

word\_list = yy

filtered\_list24 = ["frames", "frame", "variant", "networks", "network", "non-procedural", "knowledge", "system", "artificial", "data", "stereotyped systems", "script", "scripts", "values", "slots", "natural", "language"]

for xi in xc:

if xi in filtered\_list24:

print(yy)

print()

z = 0

z = ("26) What is FOPL stands for and explain its role in Artificial Intelligence?")

zz = 0

zz = ("FOPL stands for First Order Predicate \nLogic, Predicate Logic provides\na language to express assertions \nabout certain “World”. An inference \nsystem to deductive apparatus whereby we \nmay draw conclusions from such assertion. \nA semantic based on set theory")

word\_list = zz

filtered\_list25 = ["FOPL", "first order predicate logic", "language", "inference system", "deductive", "apparatus", "set theory"]

for xi in xc:

if xi in filtered\_list25:

print(zz)

print()

ak = 0

ak = ("27) What does the language of FOPL consists of?")

aaa = 0

aaa = ("The language of FOPL consists of\nA set of constant symbols. A set \nof variables. A set of predicate symbols/\n. A set of function symbols.\nThe logical connective. The Universal \nQuantifier and Existential Qualifier.\nA special binary relation of equality")

word\_list = aaa

filtered\_list26 = ["language", "FOPL", "symbols", "variables", "function", "set", "logical connective", "Universal Qualifier", "Extential Qualifier"]

for xi in xc:

if xi in filtered\_list26:

print(aaa)

print()

bk = 0

bk = ("28) For online search in ‘Artificial Intelligence’ which search agent operates by interleaving computation and action?")

bbb = 0

bbb = ("In online search, it will first take action \nand then observes the environment.")

word\_list = bbb

filtered\_list27 = ["online search", "action", "environment"]

for xi in xc:

if xi in filtered\_list27:

print(bbb)

print()

ck = 0

ck = ("29) Which search algorithm will use a limited amount of memory in online search?")

ccc = 0

ccc = ("RBFE and SMA\* will solve any kind of \nproblem that A\* can’t by using a limited amount of memory.")

word\_list = ccc

filtered\_list28 = ["RBFE", "SMA", "solve", "problem", "memory"]

for xi in xc:

if xi in filtered\_list28:

print(ccc)

print()

dk = 0

dk = ("30) In ‘Artificial Intelligence’ where you can use the Bayes rule?")

ddd = 0

ddd = ("In Artificial Intelligence to answer the \nprobabilistic queries conditioned on one \npiece of evidence, Bayes rule can be used.")

word\_list = ddd

filtered\_list29 = ["Artificial Intelligence", "answer", "probalistic", "queries", "query", "evidence", "Bayes rule"]

for xi in xc:

if xi in filtered\_list29:

print(ddd)

print()

ek = 0

ek = ("31) For building a Bayes model how many terms are required?")

eee = 0

eee = ("For building a Bayes model in AI, three terms \nare required; they are one conditional probability and two unconditional probability.")

word\_list = eee

filtered\_list30 = ["Bayes", "ai", "probability", "conditional", "unconditional"]

for xi in xc:

if xi in filtered\_list30:

print(eee)

print()

fk = 0

fk = ("32) While creating Bayesian Network what is the consequence between a node and its predecessors?")

fff = 0

fff = ("While creating Bayesian Network, the \nconsequence between a node and its \npredecessors is that a node can be \nconditionally independent of its \npredecessors.")

word\_list = fff

filtered\_list31 = ["bayesian", "node", "independant"]

for xi in xc:

if xi in filtered\_list31:

print(fff)

print()

gk = 0

gk = ("33) To answer any query how the Bayesian network can be used?")

ggg = 0

ggg = ("If a Bayesian Network is a representative \nof the joint distribution, then by \nsumming all the relevant joint entries, \nit can solve any query.")

word\_list = ggg

filtered\_list32 = ["bayesian", "joint", "distribution"]

for xi in xc:

if xi in filtered\_list32:

print(ggg)

print()

hk = 0

hk = ("34) What combines inductive methods with the power of first order representations?")

hhh = 0

hhh = ("Inductive logic programming combines inductive \nmethods with the power of first order \nrepresentations.")

word\_list = hhh

filtered\_list33 = ["inductive", "logic", "programming", "methods", "power", "first", "order"]

for xi in xc:

if xi in filtered\_list33:

print(hhh)

print()

ik = 0

ik = ("35) In Inductive Logic Programming what needed to be satisfied?")

iii = 0

iii = ("The objective of an Inductive Logic Programming\n is to come up with a set of sentences \nfor the hypothesis such that the entailment \nconstraint is satisfied.")

word\_list = iii

filtered\_list34 = ["logic", "set of sentences", "hypothosis" ]

for xi in xc:

if xi in filtered\_list34:

print(iii)

print()

jk = 0

jk = ("36) In top-down inductive learning methods how many literals are available? What are they?")

jjj = 0

jjj = ("There are three literals available in top-down \ninductive learning methods they are \npredicates Equality and Inequality \nArithmetic Literals")

word\_list = jjj

filtered\_list35 = ["top-down inductive", "equality", "inequality", "arithmetic"]

for xi in xc:

if xi in filtered\_list35:

print(jjj)

print()

lk = 0

lk = ("37) Which algorithm inverts a complete resolution strategy?")

lll = 0

lll = ("‘Inverse Resolution’ inverts a complete resolution, \nas it is a complete algorithm for learning first order theories.")

word\_list = lll

filtered\_list36 = ["inverse resolution", "inverts", "invert", "complete resolution", "algorithm", "learning", "first order", "theories", "theory"]

for xi in xc:

if xi in xc:

if xi in filtered\_list36:

print(lll)

print()

mk = 0

mk = ("38) In speech recognition what kind of signal is used?")

mmm = 0

mmm = ("In speech recognition, Acoustic signal is \nused to identify a sequence of words.")

word\_list = mmm

filtered\_list37 = ["speech recognition", "acoustics", "acoustic signal", "sequence", "words"]

for xi in xc:

if xi in filtered\_list37:

print(mmm)

print()

nk = 0

nk = ("39) In speech recognition which model gives the probability of each word following each word?")

nnn = 0

nnn = ("Biagram model gives the probability of \neach word following each other word in \nspeech recognition.")

word\_list = nnn

filtered\_list38 = ["biagram model", "probability", "word", "speech recognition"]

for xi in xc:

if xi in filtered\_list38:

print(nnn)

print()

ok = 0

ok = ("40) Which algorithm is used for solving temporal probabilistic reasoning?")

ooo = 0

ooo = ("To solve temporal probabilistic reasoning,\n HMM (Hidden Markov Model) is used, independent of transition and sensor model.")

word\_list = ooo

filtered\_list39 = ["temporal", "hmm", "markov", "sensor"]

for xi in xc:

if xi in filtered\_list39:

print(ooo)

print()

pk = 0

pk = ("41) What is Hidden Markov Model (HMMs) is used?")

ppp = 0

ppp = ("Hidden Markov Models are a ubiquitous tool \nfor modelling time series data or to \nmodel sequence behaviour. They are used \nin almost all current speech recognition systems.")

word\_list = ppp

filtered\_list40 = ["markov", "tool", "modelling", "model", "series", "data", "sequence", "sequences", "speech recognition"]

for xi in xc:

if xi in filtered\_list40:

print(ppp)

print()

qk = 0

qk = ("42) In Hidden Markov Model, how does the state of the process is described?")

qqq = 0

qqq = ("The state of the process in HMM’s model is \ndescribed by a ‘Single Discrete Random Variable’.")

word\_list = qqq

filtered\_list41 = ["hmm", "hmm's", "random"]

for xi in xc:

if xi in filtered\_list41:

print(qqq)

print()

rk = 0

rk = ("43) In HMM’s, what are the possible values of the variable?")

rrr = 0

rrr = ("‘Possible States of the World’ is the possible values \nof the variable in HMM’s.")

word\_list = rrr

filtered\_list42 = ["possible values", "variable"]

for xi in xc:

if xi in filtered\_list42:

print(rrr)

print()

sk = 0

sk = ("44) In HMM, where does the additional variable get added?")

sss = 0

sss = ("While staying within the HMM network, the \nadditional state variables can be added to a temporal model.")

word\_list = sss

filtered\_list43 = ["hmm", "network", "additional", "state", "variable", "variables", "temporary", "model"]

for xi in xc:

if xi in filtered\_list43:

print(sss)

print()

tk = 0

tk = ("45) In Artificial Intelligence, what are semantic analyses used for?")

ttt = 0

ttt = ("In Artificial Intelligence, to extract the \nmeaning from the group of sentences semantic analysis is used.")

word\_list = ttt

filtered\_list44 = ["artificial intelligence", "ai", "sentences"]

for xi in xc:

if xi in filtered\_list44:

print(ttt)

print()

uk = 0

uk = ("46) What is meant by compositional semantics?")

uuu = 0

uuu = ("The process of determining the meaning of \nP\*Q from P,Q and\* is known as Compositional Semantics.")

word\_list = uuu

filtered\_list45 = ["process", "P\*Q", "P,Q", "compositional"]

for xi in xc:

if xi in filtered\_list45:

print(uuu)

print()

vk = 0

vk = ("47) How logical inference can be solved in Propositional Logic?")

vvv = 0

vvv = ("In Propositional Logic, Logical Inference \nalgorithm can be solved by using \nLogical Equivalence/Validity/Satisfying ability")

word\_list = vvv

filtered\_list46 = ["logic", "logical", "algorithm", "equivalence", "solved"]

for xi in xc:

if xi in filtered\_list46:

print(vvv)

print()

wk = 0

wk = ("48) Which process makes different logical expression/n looks identical?")

www = 0

www = ("‘Unification’ process makes different logical \nexpressions identical. Lifted inferences \nrequire finding substitute which can make \na different expression looks identical. \nThis process is called unification.")

word\_list = www

filtered\_list47 = ["unification", "logical", "logic", "identical", "different", "expression"]

for xi in xc:

if xi in filtered\_list47:

print(www)

print()

yk = 0

yk = ("49) Which algorithm in ‘Unification and Lifting’ takes two sentences and returns a unifier?")

yyy = 0

yyy = ("In ‘Unification and Lifting’ the algorithm \nthat takes two sentences and returns a unifier \nis ‘Unify’ algorithm.")

word\_list = yyy

filtered\_list48 = ["unification", "unifier", "unify", "algorithm", "sentences"]

for xi in xc:

if xi in filtered\_list48:

print(yyy)

print()

zk = 0

zk = ("50) Which is the most straight forward approach for planning algorithm?")

zzz = 0

zzz = ("State space search is the most straight \nforward approach for planning algorithm because it \ntakes account of everything for finding a solution.")

word\_list = zzz

filtered\_list49 = ["state space search", "search","planning", "algorithm", "solution"]

for xi in xc:

if xi in filtered\_list49:

print(zzz)

print()

print()