AI Assignment 1.

Abhishek Rajgaria
2017276

This is the example of advice generated by the prolog system.

Sample 1.

```
----------Welcome to Ardia career advisory system---------
Please enter your branch:- csam.

Please enter your CGPA:- |    9.

*choose elective from: [graphtheory, artificialIntelligence, machineLearning, compiler, naturalLanguage, programming,security, networking, system, quantumComputing, a
lgorithm, finance, theater, vlsi, radarsystem, signal, communication, humanities, politics, entrepreneurship, statistics]*
*enter maximum of 3 electives*
Any electives you had taken in your B.tech.:-|    graphtheory.
Any more electives you had taken?|    finance.
Any more electives you had taken?|    machineLearning.

*choose domain from: [ml, ai, robotics, networking, entrepreneur, programming, security, maths, nlp, quantum, java, system, radar, 5g: fiveg, algos, vlsi, design, eco
nomics, filmmaking, ngorelated]*
*enter maximum of 2 projects domain*
Any project done in your B.tech.:-|    ai.
Any more project you had done?|    java.

*types of BTP:- [thesis, engineering, entrepreneurship]*
*If BTP not done then write na*
specify the type of BTP you had done:- |    thesis.

*Please choose your interest from:*
*[academic, entreprenureship, job, sports, finance, theater, civil]*
What is your interest: |    job.


Oh I see you are a csam student, that is wonderful!
How much you like csam(in scale of [0,1,2])|    1.

It is good that you have done a BTP!
Please spcify the domain your BTP
from the following list
*[ai, maths, software, hardware, design, economics]*|    ai.

Rate your expierence of BTP on sacle of [0,1,2]:|    2.

Advised career: data_analyst score 23.0
**********
scores for all the careers are *(it can also be negative)*

higher_studies: -61
data_analyst: 23.0
software_eng: 15.0
hardware_eng: 5.5
ux_design: 4.75
entrepreneur: -91.5
sports_man: -8
finance: -4
theater: -8
civil_services: -8
```

Sample 2:

```
----------Welcome to Ardia career advisory system---------
Please enter your branch:- ece.

Please enter your CGPA:- |    8.

*choose elective from: [graphtheory, artificialIntelligence, machineLearning, compiler, naturalLanguage, programming,security, networking, system, quantumComputing, a
lgorithm, finance, theater, vlsi, radarsystem, signal, communication, humanities, politics, entrepreneurship, statistics]*
*enter maximum of 3 electives*
Any electives you had taken in your B.tech.:-|    system.
Any more electives you had taken?|    algorithm.
Any more electives you had taken?|    compiler.

*choose domain from: [ml, ai, robotics, networking, entrepreneur, programming, security, maths, nlp, quantum, java, system, radar, 5g: fiveg, algos, vlsi, design, eco
nomics, filmmaking, ngorelated]*
*enter maximum of 2 projects domain*
Any project done in your B.tech.:-|    ai.
Any more project you had done?|    java.

*types of BTP:- [thesis, engineering, entrepreneurship]*
*If BTP not done then write na*
specify the type of BTP you had done:- |    engineering.

*Please choose your interest from:*
*[academic, entreprenureship, job, sports, finance, theater, civil]*
What is your interest: |    job.


Oh I see you are a ece student, that is wonderful!
How much you like ece(in scale of [0,1,2])|    1.

It is good that you have done a BTP!
Please spcify the domain your BTP
from the following list
*[ai, maths, software, hardware, design, economics]*|    software.

Rate your expierence of BTP on sacle of [0,1,2]:|    2.

Advised career: software_eng score 25.5
*******************************
scores for all the careers are *(it can also be negative)*

higher_studies: -77
data_analyst: 9.5
software_eng: 25.5
hardware_eng: 7.75
ux_design: 4.5
entrepreneur: -91.5
sports_man: -8
finance: -8
theater: -8
civil_services: -8
true
```

Aim:
I was trying to generate a machine learning type algorithm, so in order to achieve it I have tried to tune the parameters by myself.

Total number of career that can be advised is 10, namely:

1. Higher_studies
2. Data_analyst.
3. Software_eng.
4. Hardware_eng.
5. Ux_design.
6. Entrepreneurship.
7. Sports_man.

8. Finance.
9. Theater.
10. Civil_services.

I have evaluated the career based on 6 main parameters:

1. Branch
2. Cgpa
3. Electives
4. Projects
5. BTP
6. interest.

There are some more questions asked during the run time of the program, like “How much you like your branch?” etc.

Regarding the concepts of the Prolog used:

- List is used.
- Bact tracking.
- Recursion.
- Cut.
- Knowledge_base.
- more(as per usability).

/*----------Code----------*/

```
suggest() :-retractall(branch(_)),retractall(cgpa(_)),
retractall(electives(_)),retractall(projects(_)),retractall(btp(_)),retractall(interest(_)),retractall(hobbies(_)),retractall(a
pti(_)),retractall(hrs(_)),assert(hrs(0)),retractall(da(_)),assert(da(0)),retractall(soft(_)),assert(soft(0)),retractall(hrd(_)
),assert(hrd(0)),retractall(deg(_)),assert(deg(0)),retractall(en(_)),assert(en(0)),retractall(spm(_)),assert(spm(0)),retr
actall(ff(_)),assert(ff(0)),retractall(th(_)),assert(th(0)),retractall(cv(_)),assert(cv(0)),retractall(se(_)),assert(se(0)),retr
actall(sys(_)),assert(sys(0)),retractall(mth(_)),assert(mth(0)),
nl,
write('----------Welcome to Ardia career advisory system---------'),nl,
write('Please enter your branch:- '),read(B),assert(branch(B)),nl,
write('Please enter your CGPA:- '),read(C),assert(cgpa(C)),nl,
write('*choose elective from: [graphtheory, artificialIntelligence, machineLearning, compiler, naturalLanguage,
programming,security, networking, system, quantumComputing, algorithm, finance, theater, vlsi, radarsystem,
signal, communication, humanities, politics, entrepreneurship, statistics]*'),nl,
write('*enter maximum of 3 electives*'),nl,
write('Any electives you had taken in your B.tech.:-'),read(E),elective(E,3),nl,
```

```
write('*choose domain from: [ml, ai, robotics, networking, entrepreneur, programming, security, maths, nlp,
quantum, java, system, radar, 5g: fiveg, algos, vlsi, design, economics, filmmaking, ngorelated]*'),nl,
write('*enter maximum of 2 projects domain*'),nl,
write('Any project done in your B.tech.:-'),read(P),project(P,2),nl,
write('*types of BTP:- [thesis, engineering, entrepreneurship]*'),nl,
write('*If BTP not done then write na*'),nl,
write('specify the type of BTP you had done:- '),read(B1),assert(btp(B1)),nl,
write('*Please choose your interest from:*'),nl,
write('*[academic, entreprenureship, job, sports, finance, theater, civil]*'),nl,
write('What is your interest: '),read(I),assert(interest(I)),nl,/*
write('*choose hobbies from: [dancing, singing, drawing, reading, sports, movies, maths]*'),nl,
write('*choose maximum of any two hobbies*'),nl,
write('What is your hobby? '),read(H),hobby(H,2),nl,
write('*if you have not attempted the Aptitude test enter na*'),nl,
write('What is you Aptitude score? '),read(A),assert(apti(A)),*/nl.


elective(E,C):- not(E==no),C>1,assert(electives(E)),write('Any more electives you had taken?'),read(E1),C1 is
C-1,elective(E1,C1).
elective(E,_):-(E==no),!.
elective(E,C):-C==1,assert(electives(E)).

project(P,C):- not(P==no),C>1,assert(projects(P)),write('Any more project you had done?'),read(P1),C1 is
C-1,project(P1,C1).
project(P,_):-(P==no),!.
project(P,C):-(C==1),assert(projects(P)).

hobby(H,C):- not(H==no),C>1,assert(hobbies(H)),write('Any more hobby of yours? '),read(H1),C1 is
C-1,hobby(H1,C1).
hobby(H,_):-(H==no),!.
hobby(H,C):-(C==1),assert(hobbies(H)).

/*------------based on branch-----------------------------*/
bob():- retract(branch(Z)),write('Oh I see you are a '),write(Z),write(' student, that is wonderful! '),nl,write('How much
you like '),write(Z),write('(in scale of [0,1,2])'),read(Y),bob_util(Z,Y),assert(branch(Z)).
bob_util(Z,Y):-Z==cse,Y==2,bob_marks(8,2,0,0,0).
bob_util(Z,Y):-Z==cse,Y==1,bob_marks(5,1,0,0,0).
bob_util(Z,Y):-Z==cse,Y==0,bob_marks(2,0,0,0,0).

bob_util(Z,Y):-Z==csam,Y==2,bob_marks(5,5,0,0,0).
bob_util(Z,Y):-Z==csam,Y==1,bob_marks(3,3,0,0,0).
bob_util(Z,Y):-Z==csam,Y==0,bob_marks(1,1,0,0,0).
```

```
bob_util(Z,Y):-Z==ece,Y==2,bob_marks(3,2,5,0,0).
bob_util(Z,Y):-Z==ece,Y==1,bob_marks(2,1,3,0,0).
bob_util(Z,Y):-Z==ece,Y==0,bob_marks(1,0,1,0,0).

bob_util(Z,Y):-Z==csd,Y==2,bob_marks(4,1,0,5,0).
bob_util(Z,Y):-Z==csd,Y==1,bob_marks(3,0,0,3,0).
bob_util(Z,Y):-Z==csd,Y==0,bob_marks(1,0,0,1,0).

bob_util(Z,Y):-Z==css,Y==2,bob_marks(5,1,0,0,4).
bob_util(Z,Y):-Z==css,Y==1,bob_marks(3,0,0,0,3).
bob_util(Z,Y):-Z==css,Y==0,bob_marks(1,0,0,0,1).

bob_marks(A,B,C,D,E):-retract(hrs(Z)),X is (A+B+C)/2+D/2+E/2+Z, assert(hrs(X)),fail.
bob_marks(A,B,_,_,_):-retract(da(Z)),X is A/2+B/2+Z, assert(da(X)),fail.
bob_marks(A,B,_,_,_):-retract(soft(Z)),X is A/2+B/2+Z, assert(soft(X)),fail.
bob_marks(A,B,C,_,_):-retract(hrd(Z)),X is A/4+B/4+C+Z, assert(hrd(X)),fail.
bob_marks(A,_,_,D,_):-retract(deg(Z)),X is A/4+D+Z, assert(deg(X)),fail.
bob_marks(A,B,C,D,E):-retract(en(Z)),X is (A+B+C+D+E)/2+Z, assert(en(X)),fail.
bob_marks(_,_,_,_,E):-retract(cv(Z)),X is E/2+Z, assert(cv(X)),fail.
/*
bob_marks(A,B,_,_,_):-retract(se(Z)),X is A/2+B/2+Z-1, assert(se(X)),fail.
bob_marks(A,_,_,_,_):-retract(sys(Z)),X is A/2+Z, assert(sys(X)),fail.
bob_marks(_,B,_,_,_):-retract(mth(Z)),X is B+Z, assert(mth(X)),fail.*/
bob_marks(_,_,_,_,E):-retract(ff(Z)),X is E/2+Z,assert(ff(X)),fail.
bob_marks(_,_,_,_,_):-true.

/*--------------------based on branch---------------------*/

/*----------------------based on cgpa---------------------*/

boc():-retract(cgpa(Z)),retract(branch(B)),boc_util(Z,B).
boc_util(Z,B):-Z>=9,boc_marks(a,B).
boc_util(Z,B):-Z>=8,boc_marks(b,B).
boc_util(Z,B):-Z>=7,boc_marks(c,B).
boc_util(Z,B):-Z>=6,boc_marks(d,B).
boc_util(Z,B):-Z<6,boc_marks(e,B).

boc_marks(G,_):- G==a,retract(hrs(Z)),C is Z+8,assert(hrs(C)),fail.
boc_marks(G,_):- G==c,retract(hrs(Z)),C is Z-8,assert(hrs(C)),fail.
boc_marks(G,_):- (G==d ; G==e),retract(hrs(Z)),C is Z-50,assert(hrs(C)),fail.

boc_marks(G,B):- G==a,(B==cse ; B==csam),retract(da(Z)),C is Z+3, assert(da(C)),fail.
boc_marks(G,B):- G==c,(B==cse ; B==csam),retract(da(Z)),C is Z-3, assert(da(C)),fail.
```

```
boc_marks(G,B):- G==d,(B==cse ; B==csam),retract(da(Z)),C is Z-7, assert(da(C)),fail.
boc_marks(G,B):- G==e,(B==cse ; B==csam),retract(da(Z)),C is Z-10, assert(da(C)),fail.
boc_marks(G,B):- G==a,(B\==cse ; B\==csam),retract(da(Z)),C is Z+1, assert(da(C)),fail.
boc_marks(G,B):- (G==c;G==d,G==e),(B\==cse ; B\==csam),retract(da(Z)),C is Z-3, assert(da(C)),fail.

boc_marks(G,B):- (G==a),(B==cse ; B==csam ; B==csd ; B==css),retract(soft(Z)),C is Z+4, assert(soft(C)),fail.
boc_marks(G,B):- (G==a),(B==ece),retract(soft(Z)),C is Z+2, assert(soft(C)),fail.
boc_marks(G,B):- (G==c),(B==cse ; B==csam ; B==csd ; B==css),retract(soft(Z)),C is Z-2, assert(soft(C)),fail.
boc_marks(G,B):- (G==d),(B==cse ; B==csam ; B==csd ; B==css),retract(soft(Z)),C is Z-4, assert(soft(C)),fail.
boc_marks(G,B):- (G==d),(B==ece),retract(soft(Z)),C is Z-2, assert(soft(C)),fail.
boc_marks(G,B):- (G==e),(B==cse ; B==csam ; B==csd ; B==css),retract(soft(Z)),C is Z-7, assert(soft(C)),fail.
boc_marks(G,B):- (G==e),(B==ece),retract(soft(Z)),C is Z-4, assert(soft(C)),fail.

boc_marks(G,B):-G==a,B==ece, retract(hrd(Z)),C is Z+4,assert(hrd(C)),fail.
boc_marks(G,B):-G==c,B==ece, retract(hrd(Z)),C is Z-2,assert(hrd(C)),fail.
boc_marks(G,B):-G==d,B==ece, retract(hrd(Z)),C is Z-4,assert(hrd(C)),fail.
boc_marks(G,B):-G==e,B==ece, retract(hrd(Z)),C is Z-7,assert(hrd(C)),fail.

boc_marks(G,B):-G==a,B==csd, retract(deg(Z)),C is Z+4,assert(deg(C)),fail.
boc_marks(G,B):-G==c,B==csd, retract(deg(Z)),C is Z-2,assert(deg(C)),fail.
boc_marks(G,B):-G==d,B==csd, retract(deg(Z)),C is Z-4,assert(deg(C)),fail.
boc_marks(G,B):-G==e,B==csd, retract(deg(Z)),C is Z-7,assert(deg(C)),fail.

boc_marks(G,B):-G==a,B==css, retract(ff(Z)),C is Z+3,assert(deg(C)),fail.
boc_marks(G,B):-G==c,B==css, retract(ff(Z)),C is Z-2,assert(ff(C)),fail.
boc_marks(G,B):-G==d,B==css, retract(ff(Z)),C is Z-3,assert(ff(C)),fail.
boc_marks(G,B):-G==e,B==css, retract(ff(Z)),C is Z-4,assert(ff(C)),fail.

boc_marks(G,B):-G==a,B==css, retract(cv(Z)),C is Z+2,assert(cv(C)),fail.
boc_marks(G,B):-G==c,B==css, retract(cv(Z)),C is Z-1,assert(cv(C)),fail.
boc_marks(G,B):-G==d,B==css, retract(cv(Z)),C is Z-2,assert(cv(C)),fail.
boc_marks(G,B):-G==e,B==css, retract(cv(Z)),C is Z-3,assert(cv(C)),fail.

boc_marks(_,_):-true.

/*--------------------based on cgpa----------------------*/

/*evaluating on the basis of electives taken*/
util_bc(X):- X\==theater, retract(hrs(C)),C1 is C+4,assert(hrs(C1)),fail.

util_bc(X):- (X==ai ; X==maths), retract(da(C)),C1 is C+4,assert(da(C1)),fail.

util_bc(X):- (X==soft), retract(soft(C)),C1 is C+4,assert(soft(C1)),fail.
```

```
util_bc(X):- X==hardware, retract(hrd(C)),C1 is C+4,assert(hrd(C1)),fail.

util_bc(X):- X==design, retract(deg(C)),C1 is C+4,assert(deg(C1)),fail.

util_bc(X):- X==entrep, retract(en(C)),C1 is C+4,assert(en(C1)),fail.

util_bc(X):- X\==entrep, retract(en(C)),C1 is C+1.5,assert(en(C1)),fail.


util_bc(X):- X==finance, retract(ff(C)),C1 is C+4,assert(ff(C1)),fail.

util_bc(X):- X==theater, retract(th(C)),C1 is C+4,assert(th(C1)),fail.

util_bc(X):- X==civil, retract(cv(C)),C1 is C+4,assert(cv(C1)),fail.
/*
util_bc(X):- X==network, retract(se(C)),C1 is C+4,assert(se(C1)),fail.

util_bc(X):- X==system, retract(sys(C)),C1 is C+4,assert(sys(C1)),fail.

util_bc(X):- X==maths, retract(mth(C)),C1 is C+4,assert(mth(C1)),fail.
*/
util_bc(_):-true.

bc_elec(C):-C>0,retract(electives(Z)),courses(X,List),member(Z,List),util_bc(X),C1 is C-1,bc_elec(C1).
bc_elec(_):-true.
/*----------------------based on elective-----------------------*/


/*--------------------based on project-------------------*/

bc_proj(C):-C>0,retract(projects(Z)),proj(X,List),member(Z,List),proj_marks(X),C1 is C-1,bc_proj(C1).
bc_proj(_):-true.

proj_marks(X):-X\==theater, retract(hrs(Z)), C is Z+4, assert(hrs(C)),fail.
proj_marks(X):-(X==ai ; X==maths), retract(da(Z)), C is Z+4, assert(da(C)),fail.
proj_marks(X):-X==soft, retract(soft(Z)), C is Z+4, assert(soft(C)),fail.
proj_marks(X):-X==hardware, retract(hrd(Z)), C is Z+4, assert(hrd(C)),fail.
proj_marks(X):-X==design, retract(deg(Z)), C is Z+4, assert(deg(C)),fail.
proj_marks(X):-X==entrep, retract(en(Z)), C is Z+4, assert(en(C)),fail.
proj_marks(X):-X\==entrep, retract(en(Z)), C is Z+1.5, assert(en(C)),fail.
proj_marks(X):-X==finance, retract(ff(Z)), C is Z+4, assert(ff(C)),fail.
proj_marks(X):-X==theater, retract(th(Z)), C is Z+4, assert(th(C)),fail.
```

```
proj_marks(X):-X==civil, retract(cv(Z)), C is Z+4, assert(cv(C)),fail.
proj_marks(_):-true.
/*--------------------based on project-------------------*/

/*-------------based on BTP--------------------*/
bc_btp():-retract(btp(Z)),Z\==na,nl,
write('It is good that you have done a BTP! '),nl,write('Please spcify the domain your BTP'),nl,write('from the following list'),nl,write('*[ai, maths, software, hardware, design, economics]*'),read(Y),nl,write('Rate your expierence of BTP on sacle of [0,1,2]:'),read(R),btp_util(Z,Y,R).
bc_btp():-true.

btp_util(B,_,R):-B==thesis,retract(hrs(Z)), C is (Z+R*4),assert(hrs(C)).
btp_util(B,Y,R):-B\==na,(Y==ai ; Y==maths),retract(da(Z)),C is (Z+R*2),assert(da(C)).
btp_util(B,Y,R):-B\==na,Y==software,retract(soft(Z)),C is (Z+R*2),assert(soft(C)).
btp_util(B,Y,R):-B\==na,Y==hardware,retract(hrd(Z)),C is (Z+R*2),assert(hrd(C)).
btp_util(B,Y,R):-B\==na,Y==design,retract(deg(Z)),C is (Z+R*2),assert(deg(C)).
btp_util(B,Y,R):-B\==na,Y==economics,retract(ff(Z)),C is (Z+R*2),assert(ff(C)).
btp_util(B,_,R):-B\==na,B==thesis,retract(en(Z)),C is (Z+R*2),assert(en(C)).


btp_util(B,_,R):-B==engineering,retract(hrs(Z)),C is (Z+R*2),assert(hrs(C)).
btp_util(B,_,R):-B==engineering,retract(en(Z)),C is (Z+R*2),assert(en(C)).

btp_util(B,_,R):-B==entrepreneurship,retract(hrs(Z)),C is (Z+R*2),assert(hrs(C)).
btp_util(B,_,R):-B==entrepreneurship,retract(en(Z)),C is (Z+R*4),assert(en(C)).
btp_util(B,_,_):-B==na.
/*------------based on BTP----------------------*/

/*------------based on interest----------------------*/


fut_int():-retract(interest(Z)),eval_int(Z).

ent_eval(Y):- Y==yes,retract(en(Z)),C is Z+50,assert(en(C)),fail.
ent_eval(Y):- Y==no,retract(en(Z)),C is Z+4,assert(en(C)),fail.

eval_int(X):-X==academic,retract(hrs(Z1)),C1 is Z1+4,assert(hrs(C1)),retract(da(Z2)),C2 is Z2+2,assert(da(C2)),retract(soft(Z3)),C3 is Z3+2,assert(soft(C3)),retract(hrd(Z4)),C4 is Z4+2,assert(hrd(C4)),retract(deg(Z5)),C5 is Z5+2,assert(deg(C5)),retract(en(Z6)),C6 is Z6+2,assert(en(C6)),fail.
eval_int(X):-(X\==academic),retract(hrs(Z1)),C1 is Z1-100,assert(hrs(C1)),retract(da(Z2)),C2 is Z2-2,assert(da(C2)),retract(soft(Z3)),C3 is Z3-2,assert(soft(C3)),retract(hrd(Z4)),C4 is Z4-2,assert(hrd(C4)),retract(deg(Z5)),C5 is Z5-2,assert(deg(C5)),retract(en(Z6)),C6 is Z6-2,assert(en(C6)),fail.
```

```
eval_int(X):-X==entrepreneurship,write('Okay, lets now come to your interest part '),nl,write('you have opted for entrepreneurship, that is awesome!'),nl,write(' So you have any idea for startup rightnow and want to work on it: '),read(Y),ent_eval(Y).

eval_int(X):-X\==entrepreneurship,retract(en(Z)),C is Z-100,assert(en(C)),fail.

eval_int(X):-X==job,retract(da(Z1)),C1 is Z1+6,assert(da(C1)),retract(soft(Z2)),C2 is Z2+6,assert(soft(C2)),retract(hrd(Z3)),C3 is Z3+6,assert(hrd(C3)),retract(deg(Z4)),C4 is Z4+6,assert(deg(C4)),fail.

eval_int(X):-X\==job,retract(da(Z1)),C1 is Z1-4,assert(da(C1)),retract(soft(Z2)),C2 is Z2-6,assert(soft(C2)),retract(hrd(Z3)),C3 is Z3-6,assert(hrd(C3)),retract(deg(Z4)),C4 is Z4-6,assert(deg(C4)),fail.

eval_int(X):-X==sports,retract(spm(Z)),C is Z+8,assert(spm(C)),fail.
eval_int(X):-X\==sports,retract(spm(Z)),C is Z-8,assert(spm(C)),fail.

eval_int(X):-X==finance,retract(ff(Z)),C is Z+8,assert(ff(C)),fail.
eval_int(X):-X\==finance,retract(ff(Z)),C is Z-8,assert(ff(C)),fail.

eval_int(X):-X==theater,retract(th(Z)),C is Z+8,assert(th(C)),fail.
eval_int(X):-X\==theater,retract(th(Z)),C is Z-8,assert(th(C)),fail.

eval_int(X):-X==civil,retract(cv(Z)),C is Z+8,assert(cv(C)),fail.
eval_int(X):-X\==civil,retract(cv(Z)),C is Z-8,assert(cv(C)),fail.
eval_int(_):-true.
/*----------based on interest----------*/



program():-suggest(),bc_elec(3),bob(),boc(),bc_proj(2),bc_btp(),fut_int(),retractall(career(_)),retractall(mxm(_)),assert(career(doubt)),assert(mxm(-200)),final(),career(X),mxm(Y),nl,write('Advised career: '),write(X),write(' score '),write(Y),nl,write('**********'),nl,write('scores for all the careers are *(it can also be negative)*'),nl,nl,value().

final():-retract(hrs(Z)),assert(hrs(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(higher_studies)),fail.
final():-retract(da(Z)),assert(da(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(data_analyst)),fail.
final():-retract(soft(Z)),assert(soft(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(software_eng)),fail.
final():-retract(hrd(Z)),assert(hrd(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(hardware_eng)),fail.
final():-retract(deg(Z)),assert(deg(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(ux_design)),fail.
```

```
final():-retract(en(Z)),assert(en(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(entrepreneur)),fail.
final():-retract(spm(Z)),assert(spm(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(sports_man)),fail.
final():-retract(ff(Z)),assert(ff(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(finance)),fail.
final():-retract(th(Z)),assert(th(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(theater)),fail.
final():-retract(cv(Z)),assert(cv(Z)),retract(mxm(Y)),assert(mxm(Y)),Z>Y,retract(mxm(_)),assert(mxm(Z)),retract(career(_)),assert(career(civil_services)),fail.
final():-true.




value():-hrs(A),write('higher_studies: '),write(A),nl,da(B),write('data_analyst:
'),write(B),nl,soft(C),write('software_eng: '),write(C),nl,hrd(D),write('hardware_eng:
'),write(D),nl,deg(E),write('ux_design: '),write(E),nl,en(F),write('entrepreneur:
'),write(F),nl,spm(G),write('sports_man: '),write(G),nl,ff(H),write('finance: '),write(H),nl,th(I),write('theater:
'),write(I),nl,cv(J),write('civil_services: '),write(J).

/*FACTS*/

courses(soft,[programming,algorithm,compiler,system,networking,security]).
/*courses(system,[compiler,system]).
courses(network,[networking,security]).*/
courses(ai,[machineLearning, artificialLearning, naturalLanguage]).
courses(maths,[graphtheory, quantumComputing, algorithm, statisic]).
courses(hardware,[vlsi, signal, communication]).
courses(design,[design]).
courses(finance,[finance]).
courses(theater,[theater]).
courses(civil,[humanties,politics]).
courses(entrep,[entrepreneurship]).


proj(soft,[programming, java, networking, security, system]).
proj(ai,[ml, ai, nlp]).
proj(rb,[robotics]).
proj(maths,[algos,maths,quantum]).
proj(hardware,[radar, vlsi, fiveg]).
proj(design,[design]).
proj(finance,[economics]).
```

proj(theater,[filmmaking]).
proj(civil,[ngorelated]).
proj(entrep,[entrepreneur]).