APPENDIX 8

domains

stringlist = string*
integerlist = integer*

database

length(integer)
timeallow(integer)
anwer(symbol, integer, symbol, string)
usename(string)
rolno(string)
branch(string)
marks(integer);

predicates

append(stringlist, stringlist, stringlist)
checkword(stringlist, stringlist)
getfulllist(stringlist)
answerlist(symbol, integer, symbol, integer, stringlist)
firstscreen
moretime
outtime(integer)
checktime(integer)
check(integer)
show(symbol, symbol, stringlist)
para(integer, symbol, stringlist)
disp(integer, symbol, symbol)
paraist(integerlist)
membere(integer, integerlist)
membere(string, integerlist)
p(integer)
s(integer)
scrollall
prediction
scrtting
showwhat
shownot
add(integer)
paranad
"prompt(symbol)"
score
evaluate(symbol, integer, symbol, string)
specialscore(string)
showmarks

goal

scrollall, firstscreen, assert(length(0)), assert(timeallow(30)), assert(marks(0))
prediction, scrollall,
write("We have tried to know about your ability to predict before"); ni
write("reading a passage. Now you are going to scan the passage"); ni
write("Please Read Carefully"); ni, scanning, showwhat, shownot
score, showmarks
showmarks:- retract(marks(Marks)), scrollail,
write("Your total score is "). Marks).

score: - answer(PS,N,A,Ans), getbacktrack(G), evaluate(PS,N,A,Ans), 
cutbacktrack(G), fail.
score
specialscore(Ans): - getfulllist(List), upper_lower(Ans,A), member(A,List)
evaluate(p,3,Ans): - !, specialscore(Ans), add(1)
evaluate(PS,N,A,Ans): - answerlist(PS,N,A,List), 
upper_lower(Ans AA), member(AA,List), add(M)

add(Marks): - retract(marks(Old)) New - Old + Marks, assertz(marks(New))
answerlist(p,1,4,['computersystem', 'reservation', 'booking', 'management', 'organisation', 'improvement'])
answerlist(p,2,2,['financial control', 'inventory control', 'advance reservation', 'real time systems'],
airlines booking, seat reservation, automatic printing, information service
computers display, safe landing, safe take-off, management information system
reports about performance, creating database, utilisation, cargo transport
expenses record, income and expenditure, batch processing, instant
feedback
'quick processing', 'optimisation', 'resource utilisation'])
answerlist(p,4,a,1,['after'])
answerlist(p,4,b,1,['after'])
answerlist(p,4,c,1,['before'])
answerlist(p,4,d,1,['after'])
answerlist(p,4,e,1,['before'])
answerlist(p,4,f,1,['before'])
answerlist(s,2,a,1,['computers operations', 'computers'])
answerlist(s,2,b,1,['need for a real time system', 'real time system'])
answerlist(s,2,c,1,['instant seat reservation', 'seat reservation', 'reservation', 'seat reservation'])
answerlist(s,2,d,1,['flight information system', 'information system', 'information', 'updated information'])
answerlist(s,2,e,1,['safety', 'safe landing', 'safe take-off'])
answerlist(s,2,f,1,['flight information system', 'information system', 'information', 'updated information'])
answerlist(s,3,a,1,['financial control', 'inventory control', 'advance reservation', 'real time systems'],
airlines booking, seat reservation, automatic printing, information service
computers display, safe landing, safe take-off, management information system
reports about performance, creating database, utilisation, cargo transport
expenses record, income and expenditure, batch processing, instant
feedback
'quick processing', 'optimisation', 'resource utilisation'])

firstscreen

write("This Program involves two basic reading skills - PREDICTING and").
write("SCANNING The passage you are about to read has been taken from").
prediction:- p(1), scrollall, p(2), scrollall, p(3), scrollall, p(4), scrollall.

scanning:- s(2), pararead, s(1), scrollall, s(3), scrollall.

pararead:- scrollall, retract(timeallow(30)), write("READ THE FOLLOWING IN ABOUT TWO MINUTES"), nl, nl.
write("Paragraph 1"), nl, nl, disp(1, limited.para).
write("Paragraph 2"), nl, nl, disp(2, limited.para).
write("Paragraph 3"), nl, nl, disp(3, limited.para).
write("Paragraph 4"), nl, nl, disp(4, limited.para).
write("Paragraph 5"), nl, nl, disp(5, limited.para).
write("Paragraph 6"), nl, nl, disp(6, limited.para).

prompt(A):- write("Do you wish to read the passage again? (yes/no) "). readln(B). upper_kwer(B,A).
prompt(A):- pararead(A), s(1). p(1) :- write("The title of the passage you are about to read is COMPUTERISATION") nl.
write("IN INDIA. THE STORY OF INDIAN AIRLINES."). nl.
write("Can you guess what the passage is about? Give your answer."). nl.
write("within three words."). nl, nl.
write("Your first word: "). readln(P1A). assertz(answer(p.1.a.P1A)). nl.
write("Your second word: "). readln(P1B). assertz(answer(p.1.a.P1B)). nl.
write("Your third word: "). readln(P1C). assertz(answer(p.1.a.P1C)). nl.
write("Your fourth guess: "). readln(P1D). assertz(answer(p.1.a.P1D)). nl.
write("Your fifth guess: "). readln(P1E). assertz(answer(p.1.a.P1E)). nl.
write("Your sixth guess: "). readln(P1F). assertz(answer(p.1.a.P1F)). nl.
write("Your eighth guess: "). readln(P1H). assertz(answer(p.1.a.P1H)). nl.
write("Your ninth guess: "). readln(P1I). assertz(answer(p.1.a.P1I)). nl.
write("Your tenth guess: "). readln(P1J). assertz(answer(p.1.a.P1J)). nl.
p(2) :- write("List ten operations necessary to this organisation"). nl.
write("e.g., maintain its planes"). nl, nl.
write("Your first guess: "). readln(P2A). assertz(answer(p.2.a.P2A)). nl.
write("Your third guess: "). readln(P2C). assertz(answer(p.2.a.P2C)). nl.
write("Your fourth guess: "). readln(P2D). assertz(answer(p.2.a,P2D)). nl.
write("Your fifth guess: "). readln(P2E). assertz(answer(p.2.a,P2E)). nl.
write("Your sixth guess: "). readln(P2F). assertz(answer(p.2.a.P2F)). nl.
write("Your seventh guess: "). readln(P2G). assertz(answer(p.2.a.P2G)). nl.
write("Your eighth guess: "). readln(P2H). assertz(answer(p.2.a.P2H)). nl.
write("Your ninth guess: "). readln(P2I). assertz(answer(p.2.a.P2I)). nl.
write("Your tenth guess: "). readln(P2J). assertz(answer(p.2.a.P2J)). nl.
p(3) :- write("List ten words likely to be found in the text of the passage"). nl, nl.
write("Your first guess: "). readln(P3A). assertz(answer(p.3.a,P3A)). nl.
write("Your second guess: "). readln(P3B). assertz(answer(p.3.a,P3B)). nl.
write("Your third guess: "). readln(P3C). assertz(answer(p.3.a,P3C)). nl.
write("Your fourth guess: "). readln(P3D). assertz(answer(p.3.a,P3D)). nl.
write("Your fifth guess: "). readln(P3E). assertz(answer(p.3.a,P3E)). nl.
write("Your sixth guess: "). readln(P3F). assertz(answer(p.3.a.P3F)). nl.
write("Your seventh guess: "). readln(P3G). assertz(answer(p.3.a.P3G)). nl.
write("Your eighth guess: "). readln(P3H). assertz(answer(p.3.a,P3H)). nl.
write("Your ninth guess: "). readln(P3I). assertz(answer(p.3.a.P3I)). nl.
write("Your tenth guess: "). readln(P3J). assertz(answer(p.3.a,P3J)). nl.
p(4) :- write("Given below is a sentence from the passage."). nl, nl.
write("Today seat reservation in IA can be done instantly with"). nl.
write("the aid of a computer at the four international airports"). nl.
Another facility that has been made possible by computerisation is the Flight Information System. Tickets are printed automatically in the major cities. Answer before or after... The first one had been a tape based system but this one was a disk-based system. Answer before or after... The system in IA has become efficient; the management can now have up-to-the-minute reports. Answer before or after... After computerisation, the management information system in IA has become efficient; the management can now have up-to-the-minute reports. The first one had been a tape based system but this one was a disk-based system. Answer before or after... The system in IA has become efficient; the management can now have up-to-the-minute reports. Answer before or after...

Pick out the relevant paragraph that answers the questions asked... How do passengers receive information about departure and arrival times of flights? How does the computer help in ensuring the safety of airline flights? What is a realtime system? Why did a real time system become necessary? What changes occurred in the computerisation in Indian Airlines in the 1970s and early 80s? What is a domestic station?
Now that you have scanned all the paragraphs, 

```
write("*). nl. 
write("Your first guess: "). readln(S3A). assertz(answer(s.3.a,S3A), nl.
write("Your second guess: "). readln(S3B). assertz(answer(s.3.b,S3B)), nl.
write("Your third guess: "). readln(S3C). assertz(answer(s.3.c,S3C)), nl.
write("Your fourth guess: "). readln(S3D). assertz(answer(s.3.d,S3D)), nl.
write("Your fifth guess: "). readln(S3E). assertz(answer(s.3.e,S3E)), nl.
write("Your sixth guess "). readln(S3F). assertz(answer(s.3.f,S3F)), nl.
write("Your seventh guess "). readln(S3G). assertz(answer(s.3.g,S3G)), nl.
write("Your eighth guess "). readln(S3H). assertz(answer(s.3.h,S3H)), nl.
write("Your ninth guess "). readln(S3I). assertz(answer(s.3.i,S3I)), nl.
write("Your tenth guess "). readln(S3J). assertz(answer(s.3.j,S3J)), nl.
```

```
showwhat - scrollall.
```

```
write("Operations predicted by you Operations listed by you "). nl.
write("------before reading the text after scanning the text "). nl.
write("------

answer(p.2.a,P2A). answer(s.3.a,S3A),
write("*). P2A). nl.
answer(p.2.b,P2B). answer(s.3.b,S3B),
write("*). P2B). nl.
answer(p.2.c,P2C). answer(s.3.c,S3C),
write("*). P2C). nl.
answer(p.2.d,P2D). answer(s.3.d,S3D),
write("*). P2D). nl.
answer(p.2.e,P2E). answer(s.3.e,S3E),
write("*). P2E). nl.
answer(p.2.f,P2F). answer(s.3.f,S3F),
write("*). P2F). nl.
answer(p.2.g,P2G). answer(s.3.g,S3G),
write("*). P2G). nl.
answer(p.2.h,P2H). answer(s.3.h,S3H),
write("*). P2H). nl.
answer(p.2.i,P2I). answer(s.3.i,S3I),
write("*). P2I). nl.
answer(p.2.j,P2J). answer(s.3.j,S3J),
write("*). P2J). nl.
```

```
readcharf ( ).
```

```
shownot - scrollall.
```

```
The words you predicted The words you predicted "). nl.
write("*). nl.
write("------before appearing in the text that don't appear in the text "). nl.
write("------

findallfA. answer(p.2 A). Anslst).
```

```
getfullist(FullList) para(1 P1) para(2 P2). para(3 P3)
append(P1 P2 P12. append(P12 P3 P123).
append(P123 P4 P1234). append(P1234 P5 P12345)
append(P12345 P6 FullList)
```

```
append([L L L] [L L L] [L L L]).
append([L L L] L2 [L L L]).
```

```
checkword([].
```

```
checkword([HIT].List) - upper lower(H HH), member(H HH,List), nl.
write(H). checkword(T,List)
para(1."More";"than";"twenty";"years";"ago";"Indian";"Airlines","decided",;"to";"computerise";"some";"of";"their";"operations",".These","were";"to";"be";"begin";"with";"financial";"control";"and";"inventory",";"control",".";"Five";"years";"later",".";"in";"1971",".";"one",";"more",";"IBM",";"1401",";"system",";"was",";"acquired",".";"The",";"first",";"one",";"had",";"been",";"a",";"tape-based",";"system",";"but",";"this",";"one",";"was",";"a",";"disk-based",";"system",";"Besides",";"the",";"earlier",";"operations",";"advance",";"reservation",";"of",";"seats",";"for",";"passenger",";"flights",";"was",";"now",";"brought",";"within",";"the",";"scope",";"of",";"computerisation")

para(2."With","the","increase","in","the","volume","of","traffic",";"the",";"need","for","a",";"real",";"time",";"system",";"was",";"felt",";"this",";"is",";"a",";"system",";"in",";"which",";"data",";"is",";"processed",";"immediately",";"and",";"the",";"result",";"is",";"available",";"instantly",";"to",";"be",";"used",";"as",";"feedback",";"information",";"in",";"airlines",";"booking",";"each",";"booking",";"is",";"processed",";"as",";"soon",";"as",";"it",";"is",";"made",";"")


para[5] "Meteorological", "information", "is", "of", "vital", "importance", "to", "a", "flight", "the", "crew", "will", "find", "such", "information", "most", "useful", "since", "flights", "may", "have", "to", "be", "delayed", "or", "cancelled", "or", "diverted", "if", "the", "weather", "does", "not", "permit", "safe", "landing", "or", "take-off", "The", "computer", "provides", "up-to-the-minute", "information", "on", "the", "conditions", "in", "the", "airports" "and" "the", "information", "can", "be", "displayed", "on", "the", "screen", "at", "any", "airport", "from", "where", "a", "plane", "is", "to", "take-off".