V. *** S-TERMS PROBLEM (Selected for the 2nd round)

An s-term is a sequence of S's and parentheses defined recursively as follows:

S is an s-term, and if M and N are s-terms, (MN) is also an s-term.

Example of an s-term:

(((SS)(SS))S)(SS))

The right parentheses provide no new information, so they can be omitted, i.e. (MN instead of (MN), so that the previous s-term becomes:

((((SS(SSS(SS

1. Write a procedure 'gensterm' to generate s-terms: your procedure should generate n (n=length=number of S's) textfiles containing all s-terms of length 1,...,n respectively. S-terms should be separated by ";". The end of the last s-term in each file should be marked with ".".

Write a program that accepts an integer n(<=10), uses the above procedure and displays all generated s-terms on the screen.

Consider a calculus with s-terms. The only algebraic rule (s-rule) that can be used is the following: any subterm of an s-term that has the form (((SA)B)C) (where A,B and C are s-terms) can be rewritten as: ((AC)(BC)) i.e.

Context1(((SA)B)C)Context2->Context1((AC)(BC))Context2

The application of this rule on an s-term is called 'reduction' of the s-term.

There are different ways (strategies) of choosing a subterm to apply the s-rule. The succesive application of the s-rule on an s-term until no more applications of the s-rule are possible is called 'normalization' of the s-term.

Example of a reduction chain:

((((SS)(SS))S)(SS)) -> (((SS)((SS)S))(SS)) ->
((S(SS))(((SS)S)(SS))) -> ((S(SS))((S(SS))(S(SS))))

2. Choose an appropriate data structure for representing s-terms which can facilitate the application of the s-rule. Write two procedures 'readterm' and 'printterm' that transform s-terms to (and from) your representation from (and to) the form generated by 'gensterm'. Your program should be able to demonstrate these transformations.
3. Write a procedure 'reduce' to perform one reduction step according to the s-rule on a specified subterm of an s-term given in yout representation. Your program should be able to demonstrate this.
4. Write a procedure 'normalize': given an s-term, it should repeatedly choose a subterm to apply the s-rule, until no further reducations are possible or until the number of reduction steps exceeds some maximum, e.g. 30. Your program should be able to demonstrate this.
5. Finally, incorporate all of the above in a program that:

a) requests a length n from the user,

```
  b) uses s-terms of length n, generated by 'gensterm',
  c) transforms s-terms into your representation,
  d) normalizes (if possible),
  e) outputs the resulting (normalized) s-terms,
  f) outputs the number of reduction steps performed for
each s-term, or a 'not normalized' message in case of
unsuccessful normalization in 30 steps, and
  g) outputs the number of not normalized vs. the total
number of s-terms for the given length n.

EVALUATION:
  (1)     20 points
  (2)     25 points
  (3)     15 points
  (4)     20 points
  (5)     10 points
  Jury    10 points (clearity, elegance, style)

-----------------------------------------------------
```

VI. MAXIMUM GANG

A Police captain knows well all the outlaw persons of his city, as well as, every possible collaboration among them. He would like to find the maximum gang of the city.

In this case, a gang is a subset of outlaw persons so that any person in it collaborates with all others in the subset. Maximum gang means that there does not exist another gang with greater cardinality.

Create a program capable of carrying out the following tasks:

(A) Accept the police captain's data with a total number of outlaw persons less than 41. Use as input data file an ASCII one with the following structure:

```
a(1,1)
a(2,1)a(2,2)
a(3,1)a(3,2)a(3,3)
........................
a(n,1)a(n,2)a(n,3)..........a(n,n)
```

where a(i,j)=1, if person i is collaborating with person j or i=j, and a(i,j)=0, otherwise. For instance (in the case of 6 persons):

```
1
01
101
1011
01101
101111
```

In this example an output can be the following one:

```
A Maximum Gang is :
1  3  4  6
cardinality = 4
```

(B) Extend the input part of the program, so that to generate data in a random way under a given propability $0 < d < 1$ of collaboration of outlaw persons.

(C) Using random data or input file, find the maximum gang of the city. Use the same output format as in the example above (see task A).

EVALUATION

(A)	5 points
(B)	15 points
(C)	50 points
Total Machine Time	20 points
Jury	10 points

VII. MEDICAL VISITOR'S REGISTER

A medical visitor would like to have a program which would schedule all his medical appointments and help him meet as many doctors as possible in one day in order to advertise the medical products of the company he represents.

You are asked to write the program which would do this task for him. The input file consists of lines, each of which contains the name of a doctor, the beginning and the end of the interval when the doctor is ready to meet the medical visitor, as well as the name of his medical institute.

The rules for scheduling the appointments are as follows:

1. An appointment has at least 70 minutes duration; furthermore the medical visitor needs at least 30 minutes between any two appointments to travel to the next medical institute.

2. All the times in the input file belong to the same day, and the medical visitor does not want to meet any doctor twice on the same day.

3. Also, two consequtive appointments cannot be held at the same institute.

4. Among several doctors the medical visitor always prefers the one whom he can meet earlier.

5. If there is more than one doctor whom the medical visitor would be able to meet at the same time (either because by the time he finishes from the previous appointment there are doctors already waiting or because their proposed appointment would begin later but at the same time during the day) he prefers the one who has less remaining time for the given day (but of course this time at the beginning of the appointment must be enough for the required 70 minutes).

6. To visit as many doctors as possible, the medical visitor always (even during an appointment) keeps in mind the next appointment's starting time, therefore if it is necessary he can terminate the on-going appointment after the minimal 70 minutes expire, take his 30 minutes for traveling and begin the next appointment according to rule 5.

7. If he is not in a hurry for a new appointment (according to rule 6), the medical visitor stays with a doctor as long as he can.

Input file: consists of lines each of which contains a name (with the English alphabet), a blank space , a

time (in hh.mm form) indicating the possible beginning of an appointment, a dash ('-'), a time again indicating the last possible moment of the appointment, a blank space, and a name (with the English alphabet) of the medical institute.
The input file does not contain empty lines, nor is the end of it marked with special characters.
A possible input file is as follows:

```
Bob 16.00-17.25 Cross
John 09.30-11.50 Health
Charles 11.00-20.00 Chest
Don 08.00-13.20 Cross
Norman 22.00-23.05 Brain
Jerry 10.00-17.00 Health
Charles 09.20-10.40 Orthopedic
Evelyn 19.15-20.40 Orthopedic
Peter 09.35-11.55 Brain
Don 18.00-20.00 Eye
```

Output file: should contain a table in which all the possible appointments appear in their proposed chronological order. The appointments which are ruled out by any of the rules above should not appear in the table. The table should consist of lines containing the realizable beginning and end time of the possible appointments, their place, and the proposing doctor's name. The exact spacing of the output table is not important, but should look similar to the one below. The output for the previous input file is as follows:

```
08.00-09.10    Cross          Don
09.40-10.50    Health         John
11.20-12.30    Chest          Charles
13.00-15.30    Health         Jerry
16.00-17.25    Cross          Bob
19.15-20.40    Orthopedic     Evelyn
```

8. Solve the above problem for two medical visitors A and B preserving the same scheduling rules and the additional restriction that they are not allowed to visit the same doctor. Each time an appointment is available, it is taken by that medical visitor who has been free for the longest period. If both have been free for the same time period, Mr. A takes the appointment. The output consists of two appointment lists.

Evaluation:

One medical visitor	45 points
Two medical visitor	45 points
Jury	10 points (elegance, clarity, progr. style)