

POLITECNICO DI TORINO
ESAMI DI STATO PER L'ABILITAZIONE
ALLA PROFESSIONE DI INGEGNERE
II SESSIONE - ANNO 1998
Ramo: INFORMATICA Tema n. 2

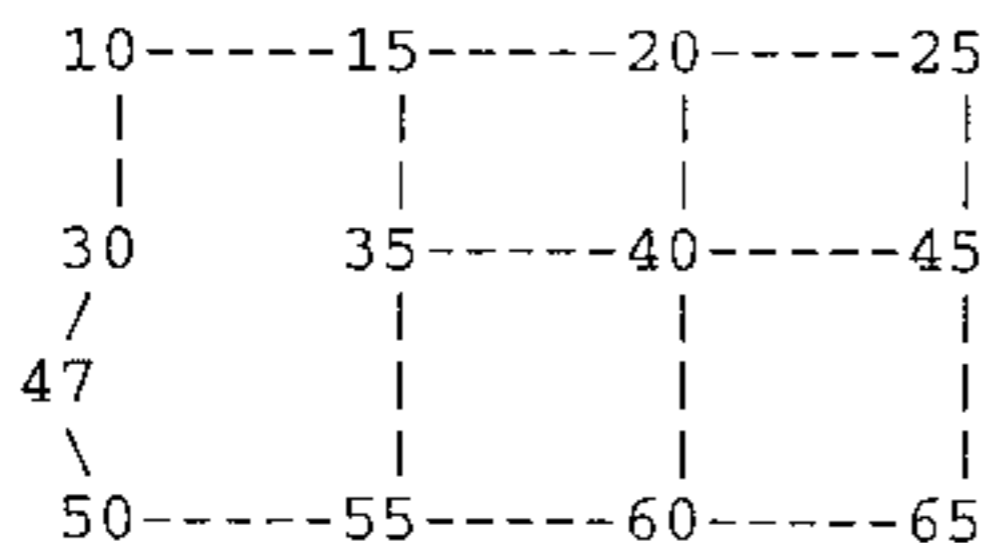
Risolvere i seguenti *due* problemi realizzando dei programmi in un linguaggio di alto livello (PASCAL o C).

Le soluzioni devono essere corredate da una buona documentazione, in cui si evidenziano sinteticamente ma efficacemente le base dati e gli algoritmi adottati.

Problema 1

Per evitare un problema che si presenta nelle reti a commutazione di pacchetti, che consiste nel fatto che i messaggi potrebbero restare in circolo per un tempo troppo lungo, in una condizione di "loop", ciascun messaggio contiene un campo detto Time To Live (TTL). Questo campo contiene il numero di nodi (stazioni, computer, ecc.) che possono ritrasmettere il messaggio, avviandolo verso la destinazione, prima che il messaggio venga drasticamente scaricato. Ogni volta che una stazione riceve un messaggio, decrementa di 1 unità il campo TTL. Se la destinazione del messaggio è la stazione corrente, allora il campo TTL è ignorato. Invece, se il messaggio deve essere mandato avanti nella rete e il campo TTL dopo il decremento contiene zero, allora il messaggio viene scaricato.

Sia data la descrizione di un certo numero di reti. Occorre determinare per ciascuna rete il numero di nodi che non sono raggiungibili fissato un nodo iniziale e un valore iniziale per il campo TTL. Si consideri ad esempio la seguente rete:



Se un messaggio con un campo TTL a 2 è stato inviato dal nodo 35, esso potrà raggiungere i nodi 15, 10, 55, 50, 40, 20 e 60, ma non potrà raggiungere i nodi 30, 47, 25, 45 e 65, poiché il suo campo TTL sarà posto a zero all'arrivo del messaggio ai nodi 10, 20, 50, e 60. Se si incrementa il valore iniziale del messaggio ponendolo pari a 3, partendo dal nodo 35 il messaggio potrebbe raggiungere tutti i nodi eccetto il nodo 45.

In un file di tipo testo c'è la descrizione di alcune reti. Ciascuna descrizione di rete inizia con un intero NC che specifica il numero di connessioni tra i nodi della rete. Dopo questo valore vi sono NC coppie di interi positivi. Queste coppie identificano i nodi che sono connessi da linee di comunicazione. Non vi è più di una linea (diretta) di comunicazione tra ogni coppia di nodi, e nessuna rete contiene più di 30 nodi. Di seguito a ciascuna configurazione di rete vi sono richieste multiple di quanti nodi non sono raggiungibili dato un nodo iniziale e il valore iniziale del campo TTL. Queste richieste sono date come coppie di interi, il primo dei quali identifica il nodo di partenza e il secondo indica il valore iniziale del campo TTL. Le richieste terminano con una coppia di zeri.

Per ciascuna richiesta il programma deve visualizzare sul videoterminale il numero progressivo del test (a partire da 1), il numero di nodi non raggiungibili, il nodo di partenza, il valore di inizializzazione del campo TTL. Gli esempi di input e output illustrano i formati di ingresso e di uscita.

File in input:

```
16
10 15 15 20 20 25 10 30 30 47 47 50 25 45 45 65
15 35 35 55 20 40 50 55 35 40 55 60 40 60 60 65
35 2 35 3 0 0
```

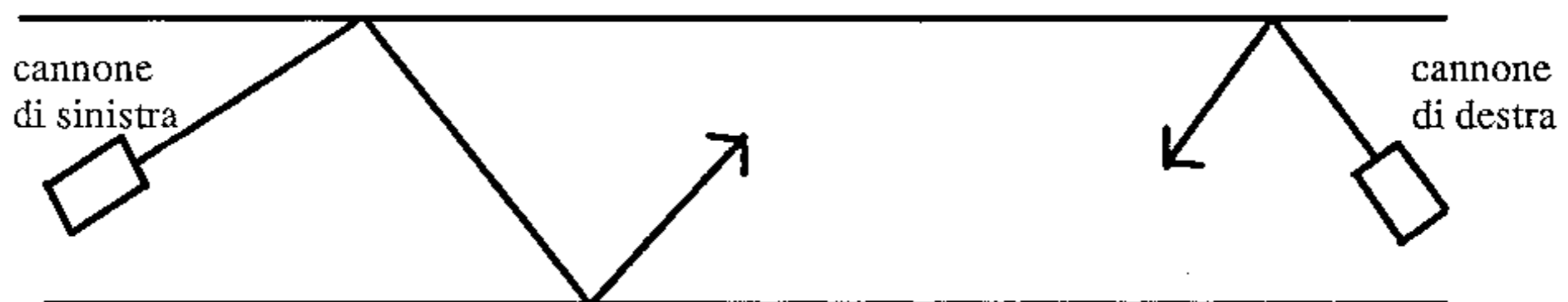
```
14
1 2 2 7 1 3 3 4 3 5 5 10 5 11
4 6 7 6 7 8 7 9 8 9 8 6 6 11
1 1 1 2 3 2 3 3 0 0
```

File in output:

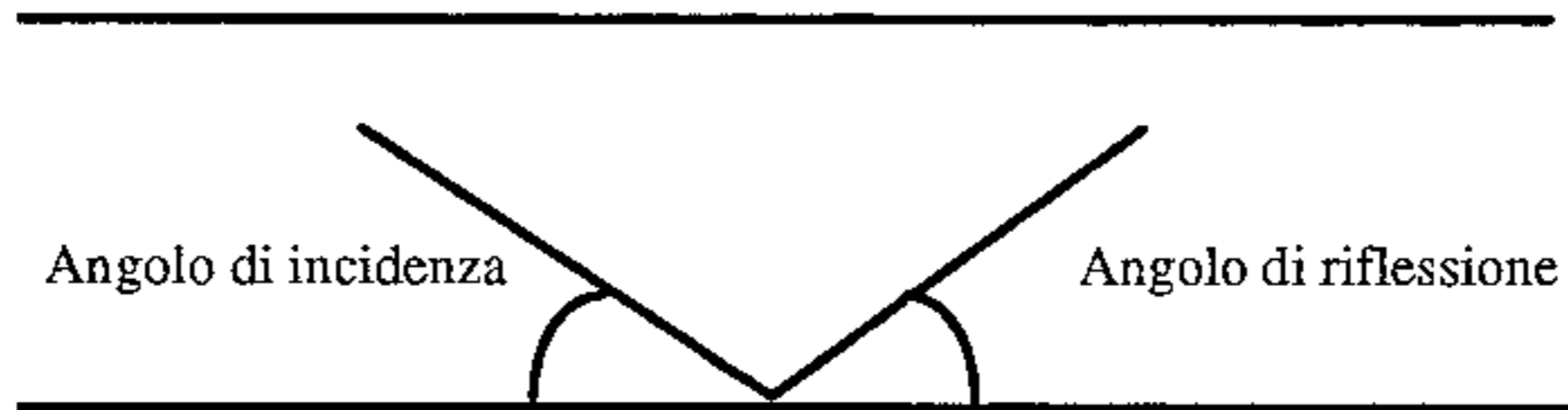
```
Test 1: 5 nodi non raggiungibili dal nodo 35 con TTL = 2
Test 2: 1 nodi non raggiungibili dal nodo 35 con TTL = 3
Test 3: 8 nodi non raggiungibili dal nodo 1 con TTL = 1
Test 4: 5 nodi non raggiungibili dal nodo 1 con TTL = 2
Test 5: 3 nodi non raggiungibili dal nodo 3 con TTL = 2
Test 6: 1 nodi non raggiungibili dal nodo 3 con TTL = 3
```

Problema 2

Una casa costruttrice di robot deve determinare sotto quali condizioni una coppia di robot di scansione (scanner robot) collidono. I robot sono lanciati simultaneamente da due "cannoni" che sono montati agli estremi di un corridoio (striscia) orizzontale. Essi viaggiano in linea retta finché non urtano con una parete del corridoio oppure non si trovano nello stesso "punto" (più precisamente lo stesso "spot") approssimativamente nello stesso istante.



Quando un robot urta contro una parete, rimbalza senza perdita di velocità e in linea retta, così che l'angolo di incidenza è uguale all'angolo di riflessione.



Se i robot sono nello stesso "spot" approssimativamente nello stesso istante, essi "collidono".

Realizzare un programma per determinare se i robot collidono e in quale posizione. Per semplificare la modellizzazione dell'evento fisico, si possono assumere le seguenti ipotesi:

- 1) La striscia orizzontale (corridoio) è bidimensionale e si estende da sinistra a destra. Le pareti sono linee rette.
- 2) Ciascun robot è un punto massa. In pratica, la circonferenza di ciascun robot è 0.
- 3) Un robot mantiene la sua velocità iniziale di lancio finché non collide con l'altro robot oppure raggiunge la posizione del cannone che ha lanciato l'altro robot.
- 4) Vi sono due cannoni, uno montato all'estremo sinistro e l'altro all'estremo destro della striscia. L'angolo iniziale del cannone di sinistra è tra -85° e $+85^\circ$. L'angolo iniziale del cannone di destra è tra 95° e 180° ovvero tra -95° e -180° (tutti gli angoli sono misurati in senso antiorario a partire dall'asse x positivo).
- 5) I robot collidono quando passano nello stesso posto con 0.5 secondi l'uno dall'altro.
- 6) La striscia orizzontale è alta 10 unità. Per ciascun punto (x, y) nella striscia, è $0 \leq y \leq 10$.
- 7) La velocità dei robot è positiva.
- 8) Le modalità con cui sono lanciati e si muovono i robot sono da ritenere provvisori, e in altre prove potrebbero cambiare. In tempi successivi potrebbe essere utile assumere leggi del moto espresse mediante funzioni analitiche complesse o tabelle, ed il comportamento potrebbe essere descritto mediante un algoritmo. Pertanto è richiesto di individuare la soluzione del problema facendo ricorso alla *simulazione*.

In un file di tipo testo vi sono i dati di alcune coppie di robot. Le linee sono da considerare a due a due. La prima linea fornisce le informazioni del lancio iniziale del robot che parte da sinistra, la seconda le informazioni relative al robot che parte da destra. Ciascuna linea contiene le seguenti 4 informazioni:

<coordinata_x> <coordinata_y> <angolo in gradi> <velocità>

Queste informazioni sono date come numeri reali e sono separate da almeno uno spazio (blank).

L'output del programma deve essere effettuato su un file di tipo testo e deve consistere di tante righe quanti sono i problemi proposti nel file di input. Ogni riga deve dichiarare se, nel caso considerato, i robot collidono o meno e, nella prima ipotesi, le coordinate del punto di collisione. Tutti i reali in output devono essere stampati con due cifre nella parte frazionaria.

Esempio di file in input:

```
0 4 0 3.3
40 5 125 5
1 6 -5 10
5 2 95 20
2 5 45 5
42 5 -135 5
0 6 20 3
0 5 180 4
```

Esempio di file in output:

```
Problema #1: i robot non collidono
Problema #2: i robot collidono in (4.68, 5.68)
Problema #3: i robot collidono in (22.00, 5.00)
Problema #4: i robot non collidono
```