

Esame 20230830

Esercizio 3

(1) Esercizio 3 v1

ESSAY marked out of 10 penalty 0 File picker

Un'astronave deve visitare n pianeti disposti a cerchio, muovendosi sempre in senso orario, ma potendo scegliere liberamente il pianeta di partenza. Ogni pianeta è identificato da un numero casuale a 4 cifre, e ospita una stazione di rifornimento che può erogare uno specifico numero di litri di carburante. Ogni coppia di pianeti adiacenti è distante uno specifico numero di anni luce. Dato un pianeta di partenza, l'astronave si rifornisce di tutto il carburante disponibile, e poi si muove al pianeta adiacente, consumando 1 litro di carburante per ogni anno luce percorso. L'astronave ripete le due operazioni — cioè rifornimento di carburante al pianeta corrente e viaggio al successivo pianeta adiacente — fino a che non completa il giro e torna al pianeta di partenza. Il serbatoio dell'astronave può contenere un numero infinito di litri di carburante, e all'inizio del viaggio — prima del rifornimento al pianeta di partenza — contiene già 1 litro di carburante.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `trovaIlPianetaDiPartenza` che prende come parametro formale un intero n che rappresenta il numero dei pianeti. Viene poi fornita una coda che contiene tutti i pianeti (già ordinati in senso orario), dove ogni pianeta è una struttura con 3 valori interi: identificativo, carburante disponibile e distanza dal pianeta successivo. La funzione `trovaIlPianetaDiPartenza` deve ritornare il numero identificativo del pianeta da cui l'astronave può partire per poter visitare tutti i pianeti — cioè compiere un giro completo — senza mai finire il carburante. Se non esiste nessun pianeta che soddisfa questa condizione, la funzione `trovaIlPianetaDiPartenza` deve ritornare il valore -1 ;

Importante: la funzione `trovaIlPianetaDiPartenza` **deve essere ricorsiva** e **NON deve contenere iteratori espliciti** (`for`, `while`, `do-while`). Sono **solo** consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non** contengano iterazioni esplicite (`for`, `while`, `do-while`).

Questi sono quattro diversi esempi di esecuzione (con relativa spiegazione):

```
computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "6008", carburante disponibile 4, distanza dal prossimo pianeta 1
Pianeta "9418", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "4271", carburante disponibile 3, distanza dal prossimo pianeta 3
Pianeta "1623", carburante disponibile 4, distanza dal prossimo pianeta 4
Pianeta "8173", carburante disponibile 5, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 6008
Rifornimento al pianeta "6008", carburante attuale 5
Viaggio al pianeta "9418", carburante rimasto 4
Rifornimento al pianeta "9418", carburante attuale 7
Viaggio al pianeta "4271", carburante rimasto 6
Rifornimento al pianeta "4271", carburante attuale 9
Viaggio al pianeta "1623", carburante rimasto 6
Rifornimento al pianeta "1623", carburante attuale 10
Viaggio al pianeta "8173", carburante rimasto 6
Rifornimento al pianeta "8173", carburante attuale 11
Viaggio al pianeta "6008", carburante rimasto 9
```

```
computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4098", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "4072", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "7195", carburante disponibile 5, distanza dal prossimo pianeta 3
```

```

Pianeta "8716", carburante disponibile 5, distanza dal prossimo pianeta 1
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 7195
Rifornimento al pianeta "7195", carburante attuale 6
Viaggio al pianeta "8716", carburante rimasto 3
Rifornimento al pianeta "8716", carburante attuale 8
Viaggio al pianeta "4098", carburante rimasto 7
Rifornimento al pianeta "4098", carburante attuale 9
Viaggio al pianeta "4072", carburante rimasto 4
Rifornimento al pianeta "4072", carburante attuale 6
Viaggio al pianeta "7195", carburante rimasto 1

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4886", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "5793", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "7492", carburante disponibile 5, distanza dal prossimo pianeta 2
Pianeta "4362", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "3059", carburante disponibile 4, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 4886
Rifornimento al pianeta "4886", carburante attuale 4
Viaggio al pianeta "5793", carburante rimasto 3
Rifornimento al pianeta "5793", carburante attuale 4
Viaggio al pianeta "7492", carburante rimasto 2
Rifornimento al pianeta "7492", carburante attuale 7
Viaggio al pianeta "4362", carburante rimasto 5
Rifornimento al pianeta "4362", carburante attuale 8
Viaggio al pianeta "3059", carburante rimasto 7
Rifornimento al pianeta "3059", carburante attuale 11
Viaggio al pianeta "4886", carburante rimasto 9

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "3761", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "8771", carburante disponibile 2, distanza dal prossimo pianeta 4
Pianeta "5713", carburante disponibile 1, distanza dal prossimo pianeta 1
Pianeta "5554", carburante disponibile 1, distanza dal prossimo pianeta 4
Non e' possibile completare il percorso tra i pianeti

```

Suggerimenti:

- Una possibile soluzione è descritta in pseudocodice nel file `esercizio3.cc`.

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `trovaIlPianetaDiPartenza` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Scaricare anche i file `coda.cc` e `coda.h` i quali implementano le funzionalità di una coda. Usare questi file nella risoluzione dell'esercizio;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.

`coda.h`

`coda.cc`

[esercizio3.cc](#)

Information for graders:

(2) Esercizio 3 v2

ESSAY marked out of 10 penalty 0 File picker

Un'astronave deve visitare n pianeti disposti a cerchio, muovendosi sempre in senso orario, ma potendo scegliere liberamente il pianeta di partenza. Ogni pianeta è identificato da un numero a 4 cifre, e ospita una stazione di rifornimento che può erogare uno specifico numero di litri di carburante. Ogni coppia di pianeti adiacenti è distante uno specifico numero di anni luce. Dato un pianeta di partenza, l'astronave si rifornisce di tutto il carburante disponibile, e poi si muove al pianeta adiacente, consumando 1 litro di carburante per ogni anno luce percorso. L'astronave ripete le due operazioni — cioè rifornimento di carburante al pianeta corrente e viaggio al successivo pianeta adiacente — fino a che non completa il giro e torna al pianeta di partenza. Il serbatoio dell'astronave può contenere un numero infinito di litri di carburante, e all'inizio del viaggio — prima del rifornimento al pianeta di partenza — contiene già 2 litri di carburante.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `trovaIlPianetaDiPartenza` che prende come parametro formale un intero n che rappresenta il numero dei pianeti. Viene poi fornita una coda che contiene tutti i pianeti (già ordinati in senso orario), dove ogni pianeta è una struttura con 3 valori interi: identificativo, carburante disponibile e distanza dal pianeta successivo. La funzione `trovaIlPianetaDiPartenza` deve ritornare il numero identificativo del pianeta da cui l'astronave può partire per poter visitare tutti i pianeti — cioè compiere un giro completo — senza mai finire il carburante. Se non esiste nessun pianeta che soddisfa questa condizione, la funzione `trovaIlPianetaDiPartenza` deve ritornare il valore -1 ;

Importante: la funzione `trovaIlPianetaDiPartenza` **deve essere ricorsiva** e **NON deve contenere iteratori espliciti** (`for`, `while`, `do-while`). Sono **solo** consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non** contengano iterazioni esplicite (`for`, `while`, `do-while`).

Questi sono quattro diversi esempi di esecuzione (con relativa spiegazione):

```
computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "6008", carburante disponibile 4, distanza dal prossimo pianeta 1
Pianeta "9418", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "4271", carburante disponibile 3, distanza dal prossimo pianeta 3
Pianeta "1623", carburante disponibile 4, distanza dal prossimo pianeta 4
Pianeta "8173", carburante disponibile 5, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 6008
Rifornimento al pianeta "6008", carburante attuale 6
Viaggio al pianeta "9418", carburante rimasto 5
Rifornimento al pianeta "9418", carburante attuale 8
Viaggio al pianeta "4271", carburante rimasto 7
Rifornimento al pianeta "4271", carburante attuale 10
Viaggio al pianeta "1623", carburante rimasto 7
Rifornimento al pianeta "1623", carburante attuale 11
Viaggio al pianeta "8173", carburante rimasto 7
Rifornimento al pianeta "8173", carburante attuale 12
Viaggio al pianeta "6008", carburante rimasto 10
```

```
computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4098", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "4072", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "7195", carburante disponibile 5, distanza dal prossimo pianeta 3
Pianeta "8716", carburante disponibile 5, distanza dal prossimo pianeta 1
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 7195
Rifornimento al pianeta "7195", carburante attuale 7
Viaggio al pianeta "8716", carburante rimasto 4
```

```

Rifornimento al pianeta "8716", carburante attuale 9
Viaggio al pianeta "4098", carburante rimasto 8
Rifornimento al pianeta "4098", carburante attuale 10
Viaggio al pianeta "4072", carburante rimasto 5
Rifornimento al pianeta "4072", carburante attuale 7
Viaggio al pianeta "7195", carburante rimasto 2

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4886", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "5793", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "7492", carburante disponibile 5, distanza dal prossimo pianeta 2
Pianeta "4362", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "3059", carburante disponibile 4, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 4886
Rifornimento al pianeta "4886", carburante attuale 5
Viaggio al pianeta "5793", carburante rimasto 4
Rifornimento al pianeta "5793", carburante attuale 5
Viaggio al pianeta "7492", carburante rimasto 3
Rifornimento al pianeta "7492", carburante attuale 8
Viaggio al pianeta "4362", carburante rimasto 6
Rifornimento al pianeta "4362", carburante attuale 9
Viaggio al pianeta "3059", carburante rimasto 8
Rifornimento al pianeta "3059", carburante attuale 12
Viaggio al pianeta "4886", carburante rimasto 10

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "3761", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "8771", carburante disponibile 2, distanza dal prossimo pianeta 4
Pianeta "5713", carburante disponibile 1, distanza dal prossimo pianeta 1
Pianeta "5554", carburante disponibile 1, distanza dal prossimo pianeta 4
Non e' possibile completare il percorso tra i pianeti

```

Suggerimenti:

- Una possibile soluzione è descritta in pseudocodice nel file `esercizio3.cc`.

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `trovaIlPianetaDiPartenza` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Scaricare anche i file `coda.cc` e `coda.h` i quali implementano le funzionalità di una coda. Usare questi file nella risoluzione dell'esercizio;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.

`coda.h`

`coda.cc`

`esercizio3.cc`

Information for graders:

(3) Esercizio 3 v3

ESSAY marked out of 10 penalty 0 File picker

Un'astronave deve visitare n pianeti disposti a cerchio, muovendosi sempre in senso orario, ma potendo scegliere liberamente il pianeta di partenza. Ogni pianeta è identificato da un numero a 4 cifre, e ospita una stazione di rifornimento che può erogare uno specifico numero di litri di carburante. Ogni coppia di pianeti adiacenti è distante uno specifico numero di anni luce. Dato un pianeta di partenza, l'astronave si rifornisce di tutto il carburante disponibile, e poi si muove al pianeta adiacente, consumando 1 litro di carburante per ogni anno luce percorso. L'astronave ripete le due operazioni — cioè rifornimento di carburante al pianeta corrente e viaggio al successivo pianeta adiacente — fino a che non completa il giro e torna al pianeta di partenza. Il serbatoio dell'astronave può contenere un numero infinito di litri di carburante, e all'inizio del viaggio — prima del rifornimento al pianeta di partenza — contiene già 3 litri di carburante.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `trovaIlPianetaDiPartenza` che prende come parametro formale un intero n che rappresenta il numero dei pianeti. Viene poi fornita una coda che contiene tutti i pianeti (già ordinati in senso orario), dove ogni pianeta è una struttura con 3 valori interi: identificativo, carburante disponibile e distanza dal pianeta successivo. La funzione `trovaIlPianetaDiPartenza` deve ritornare il numero identificativo del pianeta da cui l'astronave può partire per poter visitare tutti i pianeti — cioè compiere un giro completo — senza mai finire il carburante. Se non esiste nessun pianeta che soddisfa questa condizione, la funzione `trovaIlPianetaDiPartenza` deve ritornare il valore -1 ;

Importante: la funzione `trovaIlPianetaDiPartenza` **deve essere ricorsiva** e **NON deve contenere iteratori espliciti** (`for`, `while`, `do-while`). Sono **solo** consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non** contengano iterazioni esplicite (`for`, `while`, `do-while`).

Questi sono quattro diversi esempi di esecuzione (con relativa spiegazione):

```
computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "6008", carburante disponibile 4, distanza dal prossimo pianeta 1
Pianeta "9418", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "4271", carburante disponibile 3, distanza dal prossimo pianeta 3
Pianeta "1623", carburante disponibile 4, distanza dal prossimo pianeta 4
Pianeta "8173", carburante disponibile 5, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 6008
Rifornimento al pianeta "6008", carburante attuale 7
Viaggio al pianeta "9418", carburante rimasto 6
Rifornimento al pianeta "9418", carburante attuale 9
Viaggio al pianeta "4271", carburante rimasto 8
Rifornimento al pianeta "4271", carburante attuale 11
Viaggio al pianeta "1623", carburante rimasto 8
Rifornimento al pianeta "1623", carburante attuale 12
Viaggio al pianeta "8173", carburante rimasto 8
Rifornimento al pianeta "8173", carburante attuale 13
Viaggio al pianeta "6008", carburante rimasto 11
```

```
computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4098", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "4072", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "7195", carburante disponibile 5, distanza dal prossimo pianeta 3
Pianeta "8716", carburante disponibile 5, distanza dal prossimo pianeta 1
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 4072
Rifornimento al pianeta "4072", carburante attuale 5
Viaggio al pianeta "7195", carburante rimasto 0
```

```

Rifornimento al pianeta "7195", carburante attuale 5
Viaggio al pianeta "8716", carburante rimasto 2
Rifornimento al pianeta "8716", carburante attuale 7
Viaggio al pianeta "4098", carburante rimasto 6
Rifornimento al pianeta "4098", carburante attuale 8
Viaggio al pianeta "4072", carburante rimasto 3

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4886", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "5793", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "7492", carburante disponibile 5, distanza dal prossimo pianeta 2
Pianeta "4362", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "3059", carburante disponibile 4, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 4886
Rifornimento al pianeta "4886", carburante attuale 6
Viaggio al pianeta "5793", carburante rimasto 5
Rifornimento al pianeta "5793", carburante attuale 6
Viaggio al pianeta "7492", carburante rimasto 4
Rifornimento al pianeta "7492", carburante attuale 9
Viaggio al pianeta "4362", carburante rimasto 7
Rifornimento al pianeta "4362", carburante attuale 10
Viaggio al pianeta "3059", carburante rimasto 9
Rifornimento al pianeta "3059", carburante attuale 13
Viaggio al pianeta "4886", carburante rimasto 11

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "3761", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "8771", carburante disponibile 2, distanza dal prossimo pianeta 4
Pianeta "5713", carburante disponibile 1, distanza dal prossimo pianeta 1
Pianeta "5554", carburante disponibile 1, distanza dal prossimo pianeta 4
Non e' possibile completare il percorso tra i pianeti

```

Suggerimenti:

- Una possibile soluzione è descritta in pseudocodice nel file `esercizio3.cc`.

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `trovaIlPianetaDiPartenza` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Scaricare anche i file `coda.cc` e `coda.h` i quali implementano le funzionalità di una coda. Usare questi file nella risoluzione dell'esercizio;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.

`coda.h`

`coda.cc`

`esercizio3.cc`

Information for graders:

(4) Esercizio 3 v4

ESSAY marked out of 10 penalty 0 File picker

Un'astronave deve visitare n pianeti disposti a cerchio, muovendosi sempre in senso orario, ma potendo scegliere liberamente il pianeta di partenza. Ogni pianeta è identificato da un numero a 4 cifre, e ospita una stazione di rifornimento che può erogare uno specifico numero di litri di carburante. Ogni coppia di pianeti adiacenti è distante uno specifico numero di anni luce. Dato un pianeta di partenza, l'astronave si rifornisce di tutto il carburante disponibile, e poi si muove al pianeta adiacente, consumando 1 litro di carburante per ogni anno luce percorso. L'astronave ripete le due operazioni — cioè rifornimento di carburante al pianeta corrente e viaggio al successivo pianeta adiacente — fino a che non completa il giro e torna al pianeta di partenza. Il serbatoio dell'astronave può contenere un numero infinito di litri di carburante, e all'inizio del viaggio — prima del rifornimento al pianeta di partenza — contiene già 4 litri di carburante.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `trovaIlPianetaDiPartenza` che prende come parametro formale un intero n che rappresenta il numero dei pianeti. Viene poi fornita una coda che contiene tutti i pianeti (già ordinati in senso orario), dove ogni pianeta è una struttura con 3 valori interi: identificativo, carburante disponibile e distanza dal pianeta successivo. La funzione `trovaIlPianetaDiPartenza` deve ritornare il numero identificativo del pianeta da cui l'astronave può partire per poter visitare tutti i pianeti — cioè compiere un giro completo — senza mai finire il carburante. Se non esiste nessun pianeta che soddisfa questa condizione, la funzione `trovaIlPianetaDiPartenza` deve ritornare il valore -1 ;

Importante: la funzione `trovaIlPianetaDiPartenza` **deve essere ricorsiva** e **NON deve contenere iteratori espliciti** (`for`, `while`, `do-while`). Sono **solo** consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non** contengano iterazioni esplicite (`for`, `while`, `do-while`).

Questi sono quattro diversi esempi di esecuzione (con relativa spiegazione):

```
computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "6008", carburante disponibile 4, distanza dal prossimo pianeta 1
Pianeta "9418", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "4271", carburante disponibile 3, distanza dal prossimo pianeta 3
Pianeta "1623", carburante disponibile 4, distanza dal prossimo pianeta 4
Pianeta "8173", carburante disponibile 5, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 6008
Rifornimento al pianeta "6008", carburante attuale 8
Viaggio al pianeta "9418", carburante rimasto 7
Rifornimento al pianeta "9418", carburante attuale 10
Viaggio al pianeta "4271", carburante rimasto 9
Rifornimento al pianeta "4271", carburante attuale 12
Viaggio al pianeta "1623", carburante rimasto 9
Rifornimento al pianeta "1623", carburante attuale 13
Viaggio al pianeta "8173", carburante rimasto 9
Rifornimento al pianeta "8173", carburante attuale 14
Viaggio al pianeta "6008", carburante rimasto 12

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4098", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "4072", carburante disponibile 2, distanza dal prossimo pianeta 5
Pianeta "7195", carburante disponibile 5, distanza dal prossimo pianeta 3
Pianeta "8716", carburante disponibile 5, distanza dal prossimo pianeta 1
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 4072
Rifornimento al pianeta "4072", carburante attuale 6
Viaggio al pianeta "7195", carburante rimasto 1
```



```

Rifornimento al pianeta "7195", carburante attuale 6
Viaggio al pianeta "8716", carburante rimasto 3
Rifornimento al pianeta "8716", carburante attuale 8
Viaggio al pianeta "4098", carburante rimasto 7
Rifornimento al pianeta "4098", carburante attuale 9
Viaggio al pianeta "4072", carburante rimasto 4

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "4886", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "5793", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "7492", carburante disponibile 5, distanza dal prossimo pianeta 2
Pianeta "4362", carburante disponibile 3, distanza dal prossimo pianeta 1
Pianeta "3059", carburante disponibile 4, distanza dal prossimo pianeta 2
Invochiamo la funzione trovaIlPianetaDiPartenza...
Il pianeta di partenza e' il pianeta con identificativo 4886
Rifornimento al pianeta "4886", carburante attuale 7
Viaggio al pianeta "5793", carburante rimasto 6
Rifornimento al pianeta "5793", carburante attuale 7
Viaggio al pianeta "7492", carburante rimasto 5
Rifornimento al pianeta "7492", carburante attuale 10
Viaggio al pianeta "4362", carburante rimasto 8
Rifornimento al pianeta "4362", carburante attuale 11
Viaggio al pianeta "3059", carburante rimasto 10
Rifornimento al pianeta "3059", carburante attuale 14
Viaggio al pianeta "4886", carburante rimasto 12

computer > ./a.out
Stampa dei pianeti in senso orario
Pianeta "3761", carburante disponibile 1, distanza dal prossimo pianeta 2
Pianeta "8771", carburante disponibile 2, distanza dal prossimo pianeta 4
Pianeta "5713", carburante disponibile 1, distanza dal prossimo pianeta 1
Pianeta "5554", carburante disponibile 1, distanza dal prossimo pianeta 4
Non e' possibile completare il percorso tra i pianeti

```

Suggerimenti:

- Una possibile soluzione è descritta in pseudocodice nel file `esercizio3.cc`.

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `trovaIlPianetaDiPartenza` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Scaricare anche i file `coda.cc` e `coda.h` i quali implementano le funzionalità di una coda. Usare questi file nella risoluzione dell'esercizio;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.

`coda.h`

`coda.cc`

`esercizio3.cc`

Information for graders:

Total of marks: 40