Esame 20240109

Esercizio 3

(1) **Esercizio 3 v1**

ESSAY | marked out of 10 | penalty 0 | File picker

Il *gioco delle sedie* è un gioco musicale per bambini che avviene in questo modo: inizialmente, un numero N di bambini ($N > 0$) sono disposti in un cerchio, ognuno seduto su una sedia. A questo punto, viene fatta partire una musica e i bambini, in senso orario, iniziano a correre intorno alle sedie. Mentre la musica continua a suonare, un adulto, che sorveglia il gioco, rimuove una sedia. Dopo qualche altro giro, la musica si ferma improvvisamente ed ogni bambino deve cercare di sedersi sulla sedia più vicina. Il bambino che non trova una sedia sulla quale sedersi, è eliminato. Le sedie vengono quindi sistemate nuovamente a cerchio e la musica riparte. Il gioco finisce non appena rimane un unico bambino, che viene nominato vincitore.

Il file `esercizio3.cpp`, insieme ai file `lista.cc` e `lista.h` implementano parzialmente il gioco delle sedie tramite una lista **singolarmente concatenata circolare** in cui sono inseriti i nomi dei bambini. Tuttavia, anzichè rimuovere una sedia, in questa implementazione il partecipante che deve uscire dal gioco riceverà un colore e continuerà a girare, senza che venga rimossa una sedia. Ogni bambino della lista ha associato inizialmente un colore nero, per indicare che non gli è stato ancora assegnato un colore.

a) Si scriva la dichiarazione e la definizione della funzione `cerca()` che riceve come parametri un puntatore L a un qualsiasi nodo della lista, un valore v da cercare nella lista, e restituisca un puntatore al nodo di lista contenente v se v è presente nella lista, `NULL` altrimenti.

b) Si scriva la dichiarazione e la definizione della funzione `coloraPartecipante()`, che riceve come parametri:

- un puntatore L a un qualsiasi nodo della lista,
- un valore $k \in N$, che rappresenta il numero di secondi trascorsi dopo il quale la musica viene interrotta,
- un indice i tale che $i \in \{1, \ldots, N\}$,

La funzione scorre la lista di $k \times$ `POS_AL_SECONDO` posizioni, dove k è la durata della canzone, e `POS_AL_SECONDO` è una costante definita pari a 2, pari a quante sedie al secondo vengono visitate durante il gioco dai bambini. Una volta fatto questo, la funzione assegna un colore scelto *casualmente* al partecipante che si trova in **quella posizione più il valore dell'indice** i passato come parametro. Tuttavia, il colore scelto dovrà essere diverso rispetto a quelli dei partecipanti nelle posizioni precedente e successiva. Se la posizione scelta è già stata colorata allora la lista viene scorsa **in avanti** fino a trovare una posizione non colorata. La funzione restituisce un puntatore al nodo corrispondente al partecipante che è stato eliminato, o `NULL` se non è stato eliminato alcun partecipante. Ricordarsi di gestire i casi limite, incluso quello di una lista vuota. I partecipanti eliminati vengono comunque considerati nelle posizioni che devono essere scorse inizialmente.

È permesso aggiungere e modificare funzioni ausiliarie, ma non modificare il resto del programma. Non è permesso allocare nuove liste, ma è possibile manipolare la lista già presente con le funzioni già fornite. È permesso l'uso della libreria `cstring`, che è già inclusa nel programma.

Di seguito è riportato un esempio di esecuzione.

```
marco> a.out
Seed: 1703945587
Ci sono 6 bambini nella lista.
Roberto(-1) Marco(-1) Francesco(-1) Emily(-1) Emma(-1) Liam(-1)
------------------------------------------------------------
Giro numero 1
La canzone durera' per 47 secondi.
Fermo il bambino alla sedia numero 2.
Colore: 0
Scelto il colore 0 per il partecipante Marco
Marco ha ricevuto il colore 0
Roberto(-1) Marco(0) Francesco(-1) Emily(-1) Emma(-1) Liam(-1)
------------------------------------------------------------
Giro numero 2
La canzone durera' per 20 secondi.
Fermo il bambino alla sedia numero 4.
Colore: 0
Scelto il colore 3 per il partecipante Francesco
Francesco ha ricevuto il colore 3
Roberto(-1) Marco(0) Francesco(3) Emily(-1) Emma(-1) Liam(-1)
------------------------------------------------------------
[ ... output troncato ... ]
------------------------------------------------------------
Giro numero 6
La canzone durera' per 20 secondi.
Fermo il bambino alla sedia numero 0.
Colore: 2
Scelto il colore 2 per il partecipante Roberto
Roberto ha ricevuto il colore 2
Roberto(2) Marco(0) Francesco(3) Emily(2) Emma(3) Liam(0)
------------------------------------------------------------
Giro numero 7
La canzone durera' per 42 secondi.
Fermo il bambino alla sedia numero 0.
Colore: 3
Tutti i bambini hanno un colore. Ho finito.
Roberto(2) Marco(0) Francesco(3) Emily(2) Emma(3) Liam(0)
------------------------------------------------------------
Tutti i bambini hanno un colore. Ho finito.
```

Note:

- Il programma è randomizzato, e **non devono essere** fatte assunzioni sul numero di partecipanti, sui valori passati alle funzioni, e sui colori assegnati ai partecipanti: devono funzionare per qualsiasi input compatibile con la specifica, ed è **vietato** colorare i partecipanti manualmente.
- Scaricare il file `esercizio3.cpp`, modificarlo per inserire le dichiarazioni e le definizioni delle funzioni `coloraPartecipante` e `cerca`, e **caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstring`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.

- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

lista.cpp

lista.h

esercizio3.cpp

Information for graders:

(2) **Esercizio 3 v2**

ESSAY | marked out of 10 | penalty 0 | File picker

Il *gioco delle sedie* è un gioco musicale per bambini che avviene in questo modo: inizialmente, un numero N di bambini ($N > 0$) sono disposti in un cerchio, ognuno seduto su una sedia. A questo punto, viene fatta partire una musica e i bambini, in senso orario, iniziano a correre intorno alle sedie. Mentre la musica continua a suonare, un adulto, che sorveglia il gioco, rimuove una sedia. Dopo qualche altro giro, la musica si ferma improvvisamente ed ogni bambino deve cercare di sedersi sulla sedia più vicina. Il bambino che non trova una sedia sulla quale sedersi, è eliminato. Le sedie vengono quindi sistemate nuovamente a cerchio e la musica riparte. Il gioco finisce non appena rimane un unico bambino, che viene nominato vincitore.

Il file `esercizio3.cpp`, insieme ai file `lista.cc` e `lista.h` implementano parzialmente il gioco delle sedie tramite una lista **singolarmente concatenata circolare** in cui sono inseriti i nomi dei bambini. Tuttavia, anzichè rimuovere una sedia, in questa implementazione il partecipante che deve uscire dal gioco riceverà un colore e continuerà a girare, senza che venga rimossa una sedia. Ogni bambino della lista ha associato inizialmente un colore nero, per indicare che non gli è stato ancora assegnato un colore.

a) Si scriva la dichiarazione e la definizione della funzione `cerca()` che riceve come parametri un puntatore L a un qualsiasi nodo della lista, un valore v da cercare nella lista, e restituisca un puntatore al nodo di lista contenente v se v è presente nella lista, `NULL` altrimenti.

b) Si scriva la dichiarazione e la definizione della funzione `coloraPartecipante()`, che riceve come parametri:

- un puntatore L a un qualsiasi nodo della lista,
- un valore $k \in N$, che rappresenta il numero di secondi trascorsi dopo il quale la musica viene interrotta,
- un indice i tale che $i \in \{1, \ldots, N\}$,

La funzione scorre la lista di $k \times$ `POS_AL_SECONDO` posizioni, dove k è la durata della canzone, e `POS_AL_SECONDO` è una costante definita pari a 3, pari a quante sedie al secondo vengono visitate durante il gioco dai bambini. Una volta fatto questo, la funzione assegna un colore scelto *casualmente* al partecipante che si trova in **quella posizione più il valore dell'indice** i passato come parametro. Tuttavia, il colore scelto dovrà essere diverso rispetto a quelli dei partecipanti nelle posizioni precedente e successiva. Se la posizione scelta è già stata colorata allora la lista viene scorsa **in avanti** fino a trovare una posizione non colorata. La funzione restituisce un puntatore al nodo corrispondente al partecipante che è stato eliminato, o `NULL` se non è stato eliminato alcun partecipante. Ricordarsi di gestire i casi limite, incluso quello di una lista vuota. I partecipanti eliminati vengono comunque considerati nelle posizioni che devono essere scorse inizialmente.

È permesso aggiungere e modificare funzioni ausiliarie, ma non modificare il resto del programma. Non è permesso allocare nuove liste, ma è possibile manipolare la lista già presente con le funzioni già fornite. È permesso l'uso della libreria `cstring`, che è già inclusa nel programma.

Di seguito è riportato un esempio di esecuzione.

```
marco> ./a.out
Seed: 1703945587
Ci sono 6 bambini nella lista.
Luca(-1) Anna(-1) Giovanni(-1) Luigi(-1) Licia(-1) Letizia(-1)
------------------------------------------------------------
```

```
Giro numero 1
La canzone durera' per 47 secondi.
Fermo il bambino alla sedia numero 2.
Colore: 0
Scelto il colore 0 per il partecipante Letizia
Letizia ha ricevuto il colore 0
Luca(-1) Anna(-1) Giovanni(-1) Luigi(-1) Licia(-1) Letizia(0)
------------------------------------------------------------
Giro numero 2
La canzone durera' per 20 secondi.
Fermo il bambino alla sedia numero 4.
Colore: 0
Scelto il colore 2 per il partecipante Licia
Licia ha ricevuto il colore 2
Luca(-1) Anna(-1) Giovanni(-1) Luigi(-1) Licia(2) Letizia(0)
------------------------------------------------------------
[ ... output troncato ... ]
------------------------------------------------------------
Giro numero 6
La canzone durera' per 20 secondi.
Fermo il bambino alla sedia numero 0.
Colore: 1
Scelto il colore 1 per il partecipante Luca
Luca ha ricevuto il colore 1
Luca(1) Anna(0) Giovanni(2) Luigi(1) Licia(2) Letizia(0)
------------------------------------------------------------
Giro numero 7
La canzone durera' per 42 secondi.
Fermo il bambino alla sedia numero 0.
Colore: 2
Tutti i bambini hanno un colore. Ho finito.
Luca(1) Anna(0) Giovanni(2) Luigi(1) Licia(2) Letizia(0)
------------------------------------------------------------
Tutti i bambini hanno un colore. Ho finito.
```

Note:

- Il programma è randomizzato, e **non devono essere** fatte assunzioni sul numero di partecipanti, sui valori passati alle funzioni, e sui colori assegnati ai partecipanti: devono funzionare per qualsiasi input compatibile con la specifica, ed è **vietato** colorare i partecipanti manualmente.
- Scaricare il file `esercizio3.cpp`, modificarlo per inserire le dichiarazioni e le definizioni delle funzioni `coloraPartecipante` e `cerca`, e **caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstring`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

lista.cpp

lista.h

esercizio3.cpp

Information for graders:

(3) **Esercizio 3 v3**

ESSAY | marked out of 10 | penalty 0 | File picker

Il *gioco delle sedie* è un gioco musicale per bambini che avviene in questo modo: inizialmente, un numero N di bambini ($N > 0$) sono disposti in un cerchio, ognuno seduto su una sedia. A questo punto, viene fatta partire una musica e i bambini, in senso orario, iniziano a correre intorno alle sedie. Mentre la musica continua a suonare, un adulto, che sorveglia il gioco, rimuove una sedia. Dopo qualche altro giro, la musica si ferma improvvisamente ed ogni bambino deve cercare di sedersi sulla sedia più vicina. Il bambino che non trova una sedia sulla quale sedersi, è eliminato. Le sedie vengono quindi sistemate nuovamente a cerchio e la musica riparte. Il gioco finisce non appena rimane un unico bambino, che viene nominato vincitore.

Il file `esercizio3.cpp`, insieme ai file `lista.cc` e `lista.h` implementano parzialmente il gioco delle sedie tramite una lista **singolarmente concatenata circolare** in cui sono inseriti i nomi dei bambini. Tuttavia, anzichè rimuovere una sedia, in questa implementazione il partecipante che deve uscire dal gioco riceverà un colore e continuerà a girare, senza che venga rimossa una sedia. Ogni bambino della lista ha associato inizialmente un colore nero, per indicare che non gli è stato ancora assegnato un colore.

a) Si scriva la dichiarazione e la definizione della funzione `cerca()` che riceve come parametri un puntatore L a un qualsiasi nodo della lista, un valore v da cercare nella lista, e restituisca un puntatore al nodo di lista contenente v se v è presente nella lista, `NULL` altrimenti.

b) Si scriva la dichiarazione e la definizione della funzione `coloraPartecipante()`, che riceve come parametri:

- un puntatore L a un qualsiasi nodo della lista,
- un valore $k \in N$, che rappresenta il numero di secondi trascorsi dopo il quale la musica viene interrotta,
- un indice i tale che $i \in \{1, \ldots, N\}$,

La funzione scorre la lista di $k\times$ `POS_AL_SECONDO` posizioni, dove k è la durata della canzone, e `POS_AL_SECONDO` è una costante definita pari a 3, pari a quante sedie al secondo vengono visitate durante il gioco dai bambini. Una volta fatto questo, la funzione assegna un colore scelto *casualmente* al partecipante che si trova in **quella posizione più il valore dell'indice** i passato come parametro. Tuttavia, il colore scelto dovrà essere diverso rispetto a quelli dei partecipanti nelle posizioni precedente e successiva. Se la posizione scelta è già stata colorata allora la lista viene scorsa **all'indietro** fino a trovare una posizione non colorata. La funzione restituisce un puntatore al nodo corrispondente al partecipante che è stato eliminato, o `NULL` se non è stato eliminato alcun partecipante. Ricordarsi di gestire i casi limite, incluso quello di una lista vuota. I partecipanti eliminati vengono comunque considerati nelle posizioni che devono essere scorse inizialmente.

È permesso aggiungere e modificare funzioni ausiliarie, ma non modificare il resto del programma. Non è permesso allocare nuove liste, ma è possibile manipolare la lista già presente con le funzioni già fornite. È permesso l'uso della libreria `cstring`, che è già inclusa nel programma.

Di seguito è riportato un esempio di esecuzione.

```
marco> ./a.out
Seed: 1703945587
Ci sono 7 bambini nella lista.
```

```
Sara(-1) Simone(-1) Niccolò(-1) Elia(-1) Grace(-1) Alberto(-1) Julia(-1)
------------------------------------------------------------
Giro numero 1
La canzone durera' per 46 secondi.
Fermo il bambino alla sedia numero 3.
Colore: 1
Scelto il colore 1 per il partecipante Simone
Simone ha ricevuto il colore 1
Sara(-1) Simone(1) Niccolò(-1) Elia(-1) Grace(-1) Alberto(-1) Julia(-1)
------------------------------------------------------------
Giro numero 2
La canzone durera' per 19 secondi.
Fermo il bambino alla sedia numero 1.
Colore: 3
Scelto il colore 3 per il partecipante Niccolò
Niccolò ha ricevuto il colore 3
Sara(-1) Simone(1) Niccolò(3) Elia(-1) Grace(-1) Alberto(-1) Julia(-1)
------------------------------------------------------------
[ ... output troncato ... ]
------------------------------------------------------------
Giro numero 7
La canzone durera' per 42 secondi.
Fermo il bambino alla sedia numero 4.
Colore: 3
Scelto il colore 3 per il partecipante Alberto
Alberto ha ricevuto il colore 3
Sara(3) Simone(1) Niccolò(3) Elia(1) Grace(0) Alberto(3) Julia(1)
------------------------------------------------------------
Giro numero 8
La canzone durera' per 54 secondi.
Fermo il bambino alla sedia numero 1.
Colore: 1
Tutti i bambini hanno un colore. Ho finito.
Sara(3) Simone(1) Niccolò(3) Elia(1) Grace(0) Alberto(3) Julia(1)
------------------------------------------------------------
Tutti i bambini hanno un colore. Ho finito.
```

Note:

- Il programma è randomizzato, e **non devono essere** fatte assunzioni sul numero di partecipanti, sui valori passati alle funzioni, e sui colori assegnati ai partecipanti: devono funzionare per qualsiasi input compatibile con la specifica, ed è **vietato** colorare i partecipanti manualmente.
- Scaricare il file `esercizio3.cpp`, modificarlo per inserire le dichiarazioni e le definizioni delle funzioni `coloraPartecipante` e `cerca`, e **caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstring`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

lista.cpp

lista.h

esercizio3.cpp

Information for graders:

(4) **Esercizio 3 v4**

ESSAY | marked out of 10 | penalty 0 | File picker

Il *gioco delle sedie* è un gioco musicale per bambini che avviene in questo modo: inizialmente, un numero N di bambini ($N > 0$) sono disposti in un cerchio, ognuno seduto su una sedia. A questo punto, viene fatta partire una musica e i bambini, in senso orario, iniziano a correre intorno alle sedie. Mentre la musica continua a suonare, un adulto, che sorveglia il gioco, rimuove una sedia. Dopo qualche altro giro, la musica si ferma improvvisamente ed ogni bambino deve cercare di sedersi sulla sedia più vicina. Il bambino che non trova una sedia sulla quale sedersi, è eliminato. Le sedie vengono quindi sistemate nuovamente a cerchio e la musica riparte. Il gioco finisce non appena rimane un unico bambino, che viene nominato vincitore.

Il file `esercizio3.cpp`, insieme ai file `lista.cc` e `lista.h` implementano parzialmente il gioco delle sedie tramite una lista **singolarmente concatenata circolare** in cui sono inseriti i nomi dei bambini. Tuttavia, anzichè rimuovere una sedia, in questa implementazione il partecipante che deve uscire dal gioco riceverà un colore e continuerà a girare, senza che venga rimossa una sedia. Ogni bambino della lista ha associato inizialmente un colore nero, per indicare che non gli è stato ancora assegnato un colore.

a) Si scriva la dichiarazione e la definizione della funzione `cerca()` che riceve come parametri un puntatore L a un qualsiasi nodo della lista, un valore v da cercare nella lista, e restituisca un puntatore al nodo di lista contenente v se v è presente nella lista, `NULL` altrimenti.

b) Si scriva la dichiarazione e la definizione della funzione `coloraPartecipante()`, che riceve come parametri:

- un puntatore L a un qualsiasi nodo della lista,
- un valore $k \in N$, che rappresenta il numero di secondi trascorsi dopo il quale la musica viene interrotta,
- un indice i tale che $i \in \{1, \ldots, N\}$,

La funzione scorre la lista di $k\times$ `POS_AL_SECONDO` posizioni, dove k è la durata della canzone, e `POS_AL_SECONDO` è una costante definita pari a 2, pari a quante sedie al secondo vengono visitate durante il gioco dai bambini. Una volta fatto questo, la funzione assegna un colore scelto *casualmente* al partecipante che si trova in **quella posizione più il valore dell'indice** i passato come parametro. Tuttavia, il colore scelto dovrà essere diverso rispetto a quelli dei partecipanti nelle posizioni precedente e successiva. Se la posizione scelta è già stata colorata allora la lista viene scorsa **all'indietro** fino a trovare una posizione non colorata. La funzione restituisce un puntatore al nodo corrispondente al partecipante che è stato eliminato, o `NULL` se non è stato eliminato alcun partecipante. Ricordarsi di gestire i casi limite, incluso quello di una lista vuota. I partecipanti eliminati vengono comunque considerati nelle posizioni che devono essere scorse inizialmente.

È permesso aggiungere e modificare funzioni ausiliarie, ma non modificare il resto del programma. Non è permesso allocare nuove liste, ma è possibile manipolare la lista già presente con le funzioni già fornite. È permesso l'uso della libreria `cstring`, che è già inclusa nel programma.

Di seguito è riportato un esempio di esecuzione.

```
marco> ./a.out
Seed: 1704400514
Ci sono 5 bambini nella lista.
```

```
Eleanor(-1) Emanuela(-1) Stella(-1) Filippo(-1) Nora(-1)
------------------------------------------------------------
Giro numero 1
La canzone durera' per 18 secondi.
Fermo il bambino alla sedia numero 1.
Colore: 1
Scelto il colore 1 per il partecipante Stella
Stella ha ricevuto il colore 1
Eleanor(-1) Emanuela(-1) Stella(1) Filippo(-1) Nora(-1)
------------------------------------------------------------
Giro numero 2
La canzone durera' per 31 secondi.
Fermo il bambino alla sedia numero 3.
Colore: 3
Scelto il colore 3 per il partecipante Emanuela
Emanuela ha ricevuto il colore 3
Eleanor(-1) Emanuela(3) Stella(1) Filippo(-1) Nora(-1)
------------------------------------------------------------
[ ... output troncato ... ]
------------------------------------------------------------
Giro numero 5
La canzone durera' per 17 secondi.
Fermo il bambino alla sedia numero 4.
Colore: 1
Scelto il colore 1 per il partecipante Nora
Nora ha ricevuto il colore 1
Eleanor(2) Emanuela(3) Stella(1) Filippo(2) Nora(1)
------------------------------------------------------------
Giro numero 6
La canzone durera' per 52 secondi.
Fermo il bambino alla sedia numero 0.
Colore: 3
Tutti i bambini hanno un colore. Ho finito.
Eleanor(2) Emanuela(3) Stella(1) Filippo(2) Nora(1)
------------------------------------------------------------
Tutti i bambini hanno un colore. Ho finito.
```

Note:

- Il programma è randomizzato, e **non devono essere** fatte assunzioni sul numero di partecipanti, sui valori passati alle funzioni, e sui colori assegnati ai partecipanti: devono funzionare per qualsiasi input compatibile con la specifica, ed è **vietato** colorare i partecipanti manualmente.
- Scaricare il file `esercizio3.cpp`, modificarlo per inserire le dichiarazioni e le definizioni delle funzioni `coloraPartecipante` e `cerca`, e **caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstring`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

lista.cpp

lista.h

esercizio3.cpp

Information for graders:

Total of marks: 40