

NON SPEGNERE IL PC A FINE ESAME

Corso di Sistemi Operativi e Reti

Corso di Sistemi Operativi

Prova scritta di LUGLIO 2017

ISTRUZIONI

1. **Rinomina** la cartella chiamata "CognomeNomeMatricola" che hai trovato sul Desktop e in cui hai trovato questa traccia, sostituendo "Cognome" "Nome" e "Matricola" con i tuoi dati personali;
2. **Carica** tutto il materiale didattico che vorrai usare sul Desktop; puoi farlo solo nei primi 5 minuti della prova;
3. **Svolgi** il compito; lascia tutto il sorgente che hai prodotto nella cartella di cui al punto 1;
4. **Se vuoi che ti venga inviato il tuo elaborato** a fine esame, lascia un file di testo chiamato *email.txt* sul Desktop, contenente il tuo indirizzo e-mail.
5. **Quando hai finito** lascia la postazione facendo logout,

senza spegnere il PC.

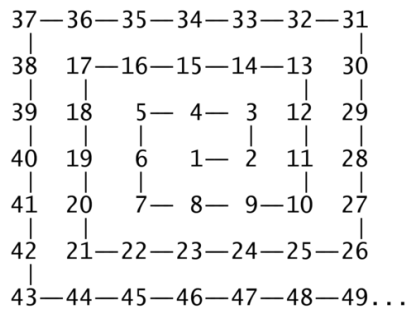
SALVA SPESSO il tuo lavoro

PER GLI STUDENTI DI SISTEMI OPERATIVI: si può sostenere solo uno dei due esercizi se si è già superata in un appello precedente la corrispondente prova. Il tempo a disposizione in questo caso è di 2 ORE.

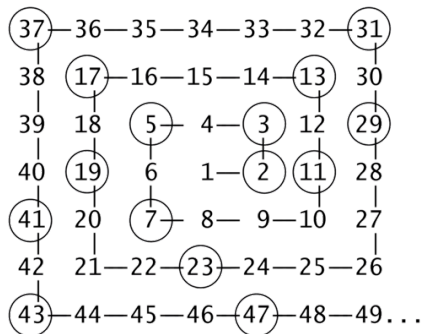
NON SPEGNERE IL PC A FINE ESAME

ESERCIZIO 1 (Programmazione multithread. Punti: 0-20)

La **spirale di Ulam** è un oggetto matematico composto da una serie di valori booleani disposti a spirale. Le caselle della spirale sono numerate: il centro della spirale corrisponde al numero intero 1 e i valori via via esterni corrispondono ai numeri interi crescenti, secondo lo schema:



Quando riempita, la spirale di Ulam contiene *true* nelle posizioni corrispondenti a un numero primo, e *false* altrove, come nello schema:



Si progetti una classe contenente il metodo pubblico:

```
void printUlam(int N, int NThreads)
```

che calcola la spirale di Ulam per i primi N numeri interi e stampa la spirale a video. Il metodo deve avvalersi di NThreads opportunamente organizzati dallo studente per parallelizzare quanto più efficientemente le operazioni di calcolo e stampa.

NON SPEGNERE IL PC A FINE ESAME

Il formato della stampa richiesto deve riportare un asterisco per le posizioni corrispondenti a numeri primi, e un punto per le posizioni corrispondenti a numeri non primi. Ad esempio la spirale di Ulam usata nell'esempio deve essere visualizzata così:

```
* . . . . . *
. * . . . * .
. . * . * . *
. * . . * * .
* . * . . . .
. . . * . . .
* . . . * . .
```

Codice che può essere usato per testare se un certo numero è primo:

```
private boolean eprimo(final int n) {
    if (n < 2)
        return false;
    if (n == 2)
        return true;
    if (n % 2 == 0)
        return false;
    for (int i = 3; i <= Math.sqrt(n); i+=2)
        if (n % i == 0)
            return false;
    return true;
}
```

NON SPEGNERE IL PC A FINE ESAME

NON SPEGNERE IL PC A FINE ESAME

CI SONO DEI PUNTI AMBIGUI NELLA TRACCIA? **COMPLETA TU**

È parte integrante di questo esercizio completare le specifiche date nei punti non esplicitamente definiti, introducendo o estendendo tutte le strutture dati laddove si ritenga necessario, e risolvendo eventuali ambiguità.

POSSO CAMBIARE IL PROTOTIPO DEI METODI RICHIESTI? **NO**

*Non è consentito modificare il prototipo dei metodi se questo è stato fornito. Potete aggiungere qualsivoglia campo e metodo di servizio, e qualsivoglia classe ausiliaria, ma **NON** variare l'interfaccia dei metodi pubblici già specificati.*

CHE LINGUAGGIO DEVO USARE? **JAVA 7 O SUCCESSIVO**

Il linguaggio da utilizzare per l'implementazione è Java. È consentito usare qualsiasi funzione di libreria di Java 7 o successivi.

MA IL MAIN() LO DEVO SCRIVERE? E I THREAD DI PROVA? **SOLO PER FARE IL TUO DEBUG**

Non è esplicitamente richiesto di scrivere un `main()` o di implementare esplicitamente del codice di prova, anche se lo si suggerisce per testare il proprio codice prima della consegna. Il codice accessorio non verrà preso in considerazione per la valutazione dell'elaborato.

NON SPEGNERE IL PC A FINE ESAME

ESERCIZIO 2 (Linguaggi di scripting. Punti 0-10)

Il comando bash `lsdf` (list open files) consente di conoscere informazioni sui file aperti dai processi in esecuzione.

Esempio 1

Un possibile output del comando `lsdf` è:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
bash	1	aaaa	cwd	DIR	136,8	1024	2	/
init	2	root	cwd	DIR	136,8	1024	2	/
bash	3	root	txt	REG	136,8	49016	1655	/sbin/init
bash	5	aaaa	txt	REG	16,7	1916	1655	/var/tmp
bash	5	aaaa	txt	DIR	31,7	2811	1655	/var/aa
init	7	aaaa	txt	REG	136,8	51084	3185	/lib/libuutil.so.1
vi	2013	root	3u	REG	136,8	0	8501	/var/tmp/ExXDaOra
...								

In questo esercizio bisogna realizzare uno script Perl, salvandolo nel file `lsdf_script.pl`, che prenda un parametro opzionale (vedi prossimo paragrafo) da riga di comando. Tale parametro rappresenta il nome di un utente (che chiameremo `USER_NAME`). Se il parametro da riga di comando (`USER_NAME`) non è specificato, il suo valore sarà dato dall'output del comando bash `whoami` (che consente di mostrare l'username dell'utente corrente).

Lo script Perl dovrà identificare il nome (`NAME`) e il tipo (`TYPE`) dei file aperti da processi che hanno come `COMMAND` il valore "bash" e come `USER` il valore di `USER_NAME`, e dovrà visualizzarli su Standard Output secondo il formato descritto in seguito.

Nota: l'output deve essere raggruppato per `PID`, come descritto in Formato Output

Formato Output - (DA RISPETTARE RIGOROSAMENTE)

L'output deve elencare i `PID` seguiti dal carattere di nuova linea '\n', e subito dopo elencare i file associati a tale `PID`.

Per ogni file aperto trovato, associato a uno specifico `PID`, deve comparire il carattere ':', seguito dal `TYPE`, seguito dal carattere ':', seguito dal `NAME`, seguito dal carattere di nuova linea '\n'.

Esempio 2

Si assuma che l'output del comando `lsOf` sia quello dell'**Esempio 1**

Se lo script Perl viene eseguito con `aaaa` come valore di `USER_NAME`, cioè così

```
./lsOf_script.pl aaaa
```

L'output dello script Perl dovrà essere:

```
1
:DIR:/
5
:REG:/var/tmp
:DIR:/var/aa
```

Nota se i processi eseguiti dall'utente `USER_NAME` non hanno file aperti (non appaiono nell'output di `lsOf`) o il parametro da riga di comando non corrisponde ad un utente valido (non appare nell'output di `lsOf`) non bisognerà visualizzare nulla su Standard Output.