

Informatica – Prova di laboratorio, 29 giugno 2023

MATRICOLA: COGNOME: NOME:

CARRAZZA ☐

MEREGHETTI ☐

TAMASCELLI ☐

QUBIT

Preparazione. In questa prova implementeremo un algoritmo per la ricostruzione di misure rilevate da un *quantum bit* (qubit) realizzato in laboratorio. Diversamente da un bit classico, che può assumere unicamente i valori 0 o 1, il *qubit* ha un comportamento probabilistico e quindi può essere configurato per assumere valori che corrispondono alla sovrapposizione degli *stati base* $|0\rangle$ e $|1\rangle$: stati della forma $\alpha|0\rangle + \beta|1\rangle$, con $\alpha, \beta \in \mathbb{R}$ e $\alpha^2 + \beta^2 = 1$ sono stati legittimi. Idealmente, se il *qubit* si trova nello stato $\alpha|0\rangle + \beta|1\rangle$ una misura dello stesso restituirà il valore 0 con probabilità α^2 e il valore 1 con probabilità β^2 . Una qualsiasi realizzazione sperimentale di un qubit è tuttavia soggetta a rumore (spesso indicato come *decoerenza*) che induce perturbazioni dello stato. La decoerenza può dipendere da molti fattori tra cui, ad esempio, la temperatura e essenzialmente “sporca” lo stato del qubit in modo tale che quando vengono ripetute N misure di uno stesso qubit sempre preparato inizialmente nello stato $\alpha|0\rangle + \beta|1\rangle$ le frequenze dei valori 0 (N_0/N) e 1 (N_1/N) misurate possono discostarsi da quelle “ideali”, ovvero α^2 e β^2 rispettivamente. In questo progetto dovremo determinare lo stato di un qubit che può essere preparato in uno dei tre stati: $|0\rangle$ ($\alpha = 1, \beta = 0$), $|1\rangle$ ($\alpha = 0, \beta = 1$) e $|+\rangle$ ($\alpha = 1/\sqrt{2}, \beta = 1/\sqrt{2}$). L'algoritmo che dovrete progettare dovrà, in particolare:

1. Caricare le misure rilevate dagli strumenti in laboratorio.
2. Eliminare i dati non affidabili.
3. Determinare lo stato preparato.

Acquisizione dati. Ogni *misura* è caratterizzata dai seguenti 3 valori:

(t , mis , $temperatura$)

dove: t indica l'istante, in secondi, in cui è stata effettuata la misurazione, mis il valore della misura (0 o 1), mentre la *temperatura* del qubit è in Kelvin. Inoltre, lo stato del qubit viene misurato per un minuto, a partire dal tempo di riferimento $t = 0$. Le misure effettuate nel tempo $[60 * (k - 1), 60 * k - 1]$ si riferiscono quindi allo stesso stato e formano un *set di misure*. Per questo esempio $k = 1, 2, 3$.

Misure degli stati. Nel nostro setup, per ogni configurazione del *qubit* il *set di misure* può riportare le seguenti frequenze di misure di 0 (N_0/N) e di 1 (N_1/N) che tengono conto della possibile presenza di decoerenza:

Stato	Etichetta	N_0/N	N_1/N
$ 0\rangle$	'0'	[0.95, 1.]	[0., 0.05]
$ 1\rangle$	'1'	[0., 0.05]	[0.95, 1.]
$ +\rangle$	'+'	[0.45, 0.55]	[0.45, 0.55]

Esempio: Poniamo di aver ottenuto in un set di misure di $N = 1000$ rilevazioni $N_0 = 480$ volte il valore 0 e $N_1 = 520$ volte il valore 1. Allora concluderemo che il qubit misurato era preparato nello stato $|+\rangle$.

Specifiche del progetto, leggete attentamente \Rightarrow

SPECIFICHE DEL PROGETTO

La specifica dei parametri della simulazione da implementare è fornita nel file `dati.dat` nella cartella `/home/comune/20230629_Dati/` sulla macchina `tolab.fisica.unimi.it`. Il file contiene, riga per riga, un numero imprecisato di *misure* acquisite dagli strumenti in laboratorio. Ciascuna misura è descritta dall'istante t , in secondi, in cui è effettuata la misura, il valore della misura (0 o 1) e la temperatura del qubit (in Kelvin). Dunque ogni riga del file `dati.dat` contiene *tre* dati di cui i primi due di tipo `int` mentre il terzo è di tipo `float`.

1. Caricare tutte le misure descritte nel file `dati.dat` in un array di `misura` allocato dinamicamente e definito dalla struttura:

```
struct misura {
    int    t;      // istante della misura in secondi
    int    mis;    // 0 o 1 rilevato dagli strumenti
    float  temp;   // temperatura del qubit in Kelvin
    char   stato;  // nome della configurazione
};
```

Il campo `stato` verrà riempito in seguito. Stampare quindi a video:

- (i) il numero di misure lette,
 - (ii) la descrizione completa delle prime e delle ultime tre misure.
2. Le misure sono state salvate in ordine casuale e contengono elementi non affidabili dovuti al surriscaldamento del setup sperimentale:
 - (i) eliminare dall'array di `misura` gli eventi con temperatura superiore a 30K, e stampare a video il numero di misure rimaste,
 - (ii) stampare a video le frequenze complessive (ovvero su tutti i dati rimasti) dei valori 0 e 1.
 3. Considerando SOLO i dati sopravvissuti alla selezione effettuata al punto (i) precedente:
 - (i) ordinare l'array di `misura` in ordine crescente in `t`, stampare a video le prime e ultime tre misure dopo l'ordinamento,
 - (ii) per ogni *set di misure*, definito come nella spiegazione, determinare lo stato del qubit usando i criteri della tabella (vedi foglio precedente). Aggiornare il campo `stato` di ogni misura con le etichette corrispondenti (vedi tabella). Stampare a video la descrizione completa delle prime e ultime tre misure riferite al *primo* stato $|+\rangle$ misurato (quindi quello che compare per primo nel vettore di `misura`).

ATTENZIONE! Tutti i risultati devono essere stampati a video e anche registrati su un file `results.out` corredati da *opportune diciture* che consentano di capire il significato di quanto stampato/registrato.

Istruzioni per la consegna del progetto, *comando di copia di file e cartelle* \Rightarrow

ISTRUZIONI PER LA CONSEGNA DEL PROGETTO

Il vostro software deve essere predisposto in una cartella denominata `cognome_matricola` che deve essere copiata in `/home/comune/20230629_Risultati` sulla macchina `tolab.fisica.unimi.it`

Nella cartella `cognome_matricola` devono essere inclusi:

- un `makefile` che tramite i comandi `make compila` e `make esegui` consenta rispettivamente di compilare e di eseguire il programma,
- il file `dati.dat` dei dati di input del progetto,
- il file `results.out` prodotto dal programma,
- tutti e soli i file `.C` `.cpp` `.cxx` `.cc` `.h` `.hpp` utili alla soluzione del problema.

Valutazione del progetto. La valutazione terrà conto sia della qualità dei risultati sia della struttura e dell'organizzazione del codice; per chiarire, sono graditi uso di funzioni e compilazione separata, mentre non è gradito un `main` onnicomprensivo. *I progetti che non compilano o che entrano in loop dopo il lancio verranno immediatamente classificati come insufficienti.*

ISTRUZIONI PER LA COPIA DI FILE E CARTELLE

- Per copiare nella vostra home directory il file di dati di input al progetto, lanciate dalla vostra home directory il comando

```
cp /home/comune/20230629_Dati/dati.dat .
```

Attenzione a non dimenticare il “.” alla fine del comando stesso.

- Per copiare la cartella contenente il vostro svolgimento nella cartella di consegna, usate il comando `cp -r` (attenzione a non dimenticare l'opzione `-r`) con opportuna sorgente (nome della cartella col vostro svolgimento) seguita da opportuna destinazione (nome della cartella ove copiare, cioè `/home/comune/20230629_Risultati/`).

Lanciate il comando dalla cartella contenente la cartella col vostro svolgimento.

ATTENZIONE!

Durante l'intero svolgimento della prova, i docenti
NON
forniranno ulteriori chiarimenti o indicazioni sull'uso del comando `cp`