Informatica – Prova di laboratorio, 20 luglio 2023

MATRICOLA: ..... COGNOME: ..... NOME: .....

Carrazza ☐ Mereghetti ☐ Tamascelli ☐

Particelle

Preparazione. In questa prova implementeremo un algoritmo per la ricostruzione di misure di particelle acquisite al CERN. Ogni particella è rappresentata dal quadrimpulso di una particella in moto:

$$\vec{p} = (E, p_x, p_y, p_z) \tag{1}$$

dove E indica l'energia della particella mentre p_i, $i \in x, y, z$ sono le componenti del vettore impulso della particella in moto.

L'algoritmo che dovrete progettare dovrà, in particolare:

1. Caricare le misure rilevate dagli strumenti in laboratorio.
2. Determinare la frazione di particelle positive, negative e neutre.
3. Determinare la rapidità delle particelle misurate.

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

$$(\vec{p},\ \texttt{carica},\ \texttt{eta})$$

dove: $\vec{p}$ indica il quadrimpulso in Eq. (1), `carica` il valore della carica elettrica della particella, mentre l'`eta` è la rapidità della particella definita come:

$$\text{eta} = \frac{1}{2} \log \left(\frac{E + p_z}{E - p_z} \right). \tag{2}$$

Specifiche del progetto, leggete attentamente ⇒

Specifiche del progetto

Le misure effettuate al CERN sono fornite nel file `dati.dat` nella cartella `/home/comune/20230720_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 da quadrimpulsi `p=(E,px,py,pz)`, cioè energia e componenti lungo i tre assi del vettore momento, da un campo `carica` con la carica della particella, e dalla rapidità `eta` che sarà riempita in seguito. Dunque ogni riga del file `dati.dat` contiene *cinque* dati di cui i primi quattro relativi a `p` di tipo `double` e l'ultimo relativo a `carica` di tipo `int`.

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

   ```
   struct misura {
      double p[4]; // energia-momento (E, px, py, pz)
      int carica;  // carica elettrica
      double eta;  // rapidita'
   };
   ```

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

   (i) il numero di misure lette,

   (ii) il numero di misure con carica positiva, negativa e neutra,

   (iii) la percentuale di misure positive, negative e neutre rispetto al numero totale di particelle

2. Implementare una funzione che calcola la rapiditià `eta` per ogni misura usando la formula in equazione (2):

   (i) Calcolare la rapidità e aggiornare il relativo campo `eta` per ogni misura.

   (ii) Stampare a video il campo `eta` delle prime cinque misure.

3. A partire dall'array di misure caricate al Punto 1 e aggiornata al Punto 2 ordinare l'array di misure in ordine di `eta` crescente e:

   (i) stampare a video la descrizione completa delle prime tre e delle ultime tre misure.

   (ii) calcolare e stampare a video la *media*, la *deviazione standard*, il *minimo* e il *massimo* valore di `eta` per tutte le misure, per le misure positive, per le misure negative e per le misure neutre. Ricordiamo che la deviazione standard σ di una sequenza $x_1, \ldots, x_n$ si calcola come $\sigma = \sqrt{(\sum_{i=1}^{n}(x_i - \mu)^2)/n}$, ove μ indica la media della sequenza.

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* ⇒

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/20230720_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` omnicomprensivo. *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/20230720_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/20230720_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`
.