

Laboratorio di Calcolo per Fisici, Esame del 01/02/2024 mattina, A.A. 2023/2024

Nome _____ Cognome _____
 Matricola _____ Ritirato/a

Lo scopo di questa esercitazione è di scrivere un programma in C e uno script in python seguendo la traccia riportata di seguito. Si tenga presente che:

1. Per svolgere il compito si hanno a disposizione 3 ore.
2. Si possono usare libri di testo, prontuari e gli appunti ma non è ammesso parlare con nessuno né utilizzare cellulari, tablet o laptop, pena l'annullamento del compito.
3. Il programma va scritto e salvato esclusivamente sul computer del laboratorio, a cui si deve accedere utilizzando come username **studente** e come password **informatica**
4. **Tutti i file vanno salvati in una cartella chiamata ELCG24_NOME_COGNOME nella home directory**, dove NOME e COGNOME indicano rispettivamente il tuo nome e cognome. Ad esempio lo studente *Marco Rossi* deve creare una cartella chiamata ELCG24_MARCO_ROSSI contenente tutti i file specificati nel testo. **Tutto ciò che non si trova all'interno della cartella suddetta non verrà valutato.** In tutti i programmi e script inserisci all'inizio un commento con il tuo nome, cognome e numero di matricola.
5. **Dovete consegnare il presente testo indicando nome, cognome e numero di matricola** (vedi sopra), barrando la casella "Ritirato/a" se ci si vuole ritirare, ovvero se non si vuole che la presente prova venga valutata.
6. **Per consegnare il compito** dovrete eseguire, all'interno della cartella creata in precedenza (come spiegato al punto 4), il seguente comando da terminale: `cp * /media/sf_esame/`

Lo scopo di questa prova è scrivere un programma che effettui la somma di due numeri interi positivi a e b rappresentati in base 2 e generati casualmente in $[0, 127]$. Per rappresentare i numeri a e b in formato binario si dovranno utilizzare due array di NBITS interi (ogni elemento dell'array varrà 0 o 1), dove $\text{NBITS} = 8$ dovrà essere definito con un'opportuna macro. In questa rappresentazione il bit meno significativo sarà nella casella con indice 0 e quello più significativo in quella con indice 7.

Per generare casualmente i numeri a e b in modo che siano compresi tra 0 e 127 si dovranno generare a caso i primi 7 bit meno significativi di ognuno, in modo che le cifre 0 e 1 siano equiprobabili, e impostare a 0 il bit più significativo (cioè quello di indice 7). I due numeri dovranno essere sommati e la loro somma andrà memorizzata in un array di 8 interi chiamato $s2$. Il numero binario così ottenuto andrà convertito in base 10 per ottenere $s10 \in [0, 254]$. Generando N volte i numeri a e b , si dovrà infine creare un istogramma relativo alle occorrenze di $s10$ utilizzando un'array di 255 interi chiamato $isto$. Il numero N va chiesto in input all'utente.

Esempio conversione

indice	3	2	1	0
	a	0	1	0
		1	0	1

in base 10 diventa:

$$1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 = 5$$

Esempio somma

indice	3	2	1	0			
	a	0	1	0	1	+	
		0	1	1	1	=	$5 +$
		1	1	0	0	=	$7 =$
	$s2$	1	1	0	0		12

↑ bit più significativo
↑ bit meno significativo

Figura 1: Esempi a 4 bit: A sinistra la conversione in decimale, a destra un esempio di somma in binario, con la corrispettiva operazione mostrata anche in decimale. In entrambi i casi sono esplicitamente indicati gli indici delle caselle corrispondenti alle cifre mostrate.

► **Prima parte:** Si scriva un programma chiamato `nome_cognome.c` (tutto minuscolo, senza eventuali spazi, accenti o apostrofi) che calcoli la somma di due numeri binari generati casualmente come spiegato in precedenza. In particolare il programma dovrà:

1. Chiedere all'utente di inserire il numero N assicurandosi che sia maggiore di 1.
2. Generare casualmente N volte i numeri a e b rappresentati in forma binaria e memorizzati come degli array di NBITS interi.

3. All'interno del ciclo su N , convertire in base 10 i numeri a , b e la loro somma, e verificare che quest'ultima sia corretta, stampandoli tutti e tre su schermo. **Una volta sicuri del funzionamento del programma, commentate l'istruzione di stampa e quelle relative alla conversione di a e b .**
4. Calcolare l'istogramma delle occorrenze di `s10` (cioè della somma dei numeri generati casualmente) utilizzando un array `isto`.
5. Alla fine delle N iterazioni stampare su un file chiamato `frequenze.dat`, le frequenze con cui si ottengono i possibili valori della somma di a e b . Per ottenere tali frequenze basta dividere le occorrenze memorizzate in `isto` per N . Le frequenze devono venir stampate con 5 cifre significative dopo la virgola.

Nello scrivere il programma si richiede che vengano implementate almeno le seguenti funzioni:

- `inserimento(...)` che chieda all'utente di inserire N , assicurandosi che tale valore sia maggiore di 1.
- `genera_numero(...)` che generi un numero binario di 8 bit, di cui i primi 7 devono venire generati casualmente, mentre quello più significativo deve essere posto uguale a 0. Tale funzione prende come argomento un array di NBITS interi.
- `somma(...)` che somma due numeri binari di NBITS bit. Tale funzione prende come argomenti 3 puntatori a interi (o array), dove i primi due puntatori verranno usati per passare gli array a e b , mentre il terzo puntatore per passare un array di NBITS dove verrà memorizzata la somma di a e b . *Suggerimento:* Per sommare i due numeri binari memorizzati negli array passati a questa funzione si dovrà effettuare la somma di bit corrispondenti (con lo stesso indice nell'array) aggiungendo a tale somma un'eventuale riporto e partendo dai bit meno significativi (quelli con indice 0). Notare che se la somma dei bit corrispondenti e del riporto è maggiore di 1 si dovrà sottrarre 2 a tale somma e riportare un 1 nella somma successiva (analogamente a quello che si fa in base 10). Negli altri casi invece il riporto dovrà essere pari a 0.
- `converti_numero(...)` che converte un numero binario in un numero in base 10 e che prende come argomento l'array di NBITS interi in cui è memorizzato il numero binario da convertire e restituisce il numero in base 10 corrispondente.

► **Seconda parte:** Utilizzando il file `frequenze.dat` generato per $N = 10000000$ creare con python il relativo grafico. Infine, salvare un'immagine di tale grafico in un file chiamato "frequenze.png". Lo script python si dovrà chiamare `nome_cognome.py`.