

[4]

Si consideri la seguente funzione `get_random()` per la generazione di numeri pseudo-casuali interi positivi:

<https://pastebin.com/f6eAKNQy>

NB: ai fini della generazione di numeri in virgola mobile, si faccia uso della costante `UINT_MAX` (<limits.h>) unitamente alla funzione `get_random()`.

Si scriva un programma C come segue:

- **A (8 punti)** - il programma prenda un input da tastiera (argomenti della funzione `main`) costituito da intero positivo N in $[20,25]$, un intero positivo M in $[10,15]$ e due numeri in virgola mobile positivi x,y tali che x in $[5.0, 10.0]$, y in $[40, 50]$. Se gli argomenti a riga di comando non rispondono ai suddetti requisiti, il programma stampa un messaggio di errore sullo standard error e termina la propria esecuzione.
- **B (15 punti)** - il programma produca una sequenza di operazioni di inserimento (`push()`) di numeri double casuali in $[x,y]$ ed operazioni `pop()` in una struttura dati LIFO (stack o pila) di dimensione massima $N/2$ da implementare mediante array, nel seguente modo:
 - lo stack (sia $S1$) sarà rappresentato da una struct che contiene almeno le seguenti informazioni: array di double + informazione che rappresenta il top dello stack (indice o puntatore);
 - la prima operazione sarà di inserimento (`push()`);
 - ogni qual volta il numero inserito (sia w) è tale che $w \geq (x+y)/2$, allora la successiva operazione sarà una rimozione dallo stack (`pop()`);
 - l'operazione successiva ad ogni rimozione (`pop()`) sarà un inserimento (`push()`);
 - andranno eseguite esattamente N operazioni di inserimento, mentre le rimozioni `pop()` potranno essere al più M ;
 - se si raggiunge il numero massimo di rimozioni (`pop()`), le successive operazioni saranno inserimenti fino al numero massimo N ;
 - per ogni operazione di `pop()`, l'elemento rimosso andrà stampato sullo standard error;
 - un tentativo di inserimento in uno stack pieno termina la procedura;
- **C (5 punti)** si modifichi la procedura al punto B in modo da inserire ogni elemento rimosso dallo stack del punto precedente in un altro analogo stack (sia $S2$) di dimensione massima $N/2$;
- **D (2 punti)** si stampi sullo standard output il contenuto di $S1$ ed $S2$, dal top alla base;

Il programma potrà essere articolato in un unico file sorgente, ma dovrà contenere almeno le seguenti funzioni (con opportuni parametri formali):

- **puntoA()**: una funzione che prenda in input i parametri `argv` e `argc` della funzione `main` e che restituisca un record (struct) con i parametri `o`, in caso i requisiti dei parametri non siano soddisfatti un puntatore `NULL`;