

Esame di Programmazione (mod A)

CDL in Intelligenza Artificiale e Data Analytics.

Giulio Caravagna (gcaravagna@units.it)

II Appello: Febbraio 2021

Istruzioni

L'appello contiene **6** esercizi (A1, A2, A3, B1, B2, B3) da risolvere in 3 ore (tempo massimo).

- Scaricate il file **Appello.zip** da Moodle che contiene un template di 6 cartelle, una per esercizio.
- Le soluzioni devono essere caricate sul portale Moodle alla pagina del corso (in formato **zip**, ricompilando le cartelle che avete scaricato come template, dopo aver aggiunto le vostre soluzioni).

Importante. I primi 3 esercizi (A1, A2 e A3) sono considerati di *sbarramento* e permettono di raggiungere 18/30 (minimo per superare l'esame). I restanti 3 esercizi (B1, B2 e B3) sono opzionali, e valgono fino al raggiungimento del voto massimo di 30/30.

Risoluzione degli esercizi di sbarramento.

- ogni esercizio va risolto partendo dall'implementazione disponibile nel file **main.c**.
- Per esempio, se lavorate su **repl.it**:
 - create un nuovo repl per C, trascinatevi la cartella del template e verificate che compaiano le cartelle ed i files necessari per l'esercizio;
 - risolvete l'esercizio modificando **main.c**;
 - testate la vostra soluzione (**make test1**, **make test2** etc.):
 - * i files di input sono all'interno della cartella **input**.
 - * i risultati attesi (per input) sono all'interno della cartella **result**.
 - * i vostri risultati devono comparire nella cartella **output**.
 - NOTA: verificate la presenza dei vostri risultati dentro la cartella **output** - se non ci sono files, il vostro programma NON funziona.
 - una volta terminato scaricate la vostra soluzione e sostituirla al template.

Esercizi di sbarramento

Es. A1 (6 punti)

Si scriva un programma C che calcola, per un dato $n \geq 0$ in input, la seguente successione numerica

$$\begin{cases} a_0 = 1 \\ a_1 = 2 \\ a_n = \frac{(n + 3(a_{n-2} - 2))}{a_{n-1}} & \text{con } n \geq 3 \text{ se } n \text{ e' pari} \\ a_n = \frac{(n + 3(a_{n-1} - 2))}{a_{n-2}} & \text{con } n \geq 3 \text{ se } n \text{ e' dispari} \end{cases}$$

in modo esclusivamente *iterativo* (ciclo `for`).

Esempio di calcolo:

- $a_2 = (2 + 3(a_0 - 2))/a_1 = -0.5$
- $a_3 = (3 + 3(a_2 - 2))/a_1 = -2.25$
- $a_4 = \dots = 1.5555555$

Suggerimento. Fate attenzione ai tipi delle variabili dichiarate!

Es. A2 (6 punti)

Si scriva un programma C che, per un dato numero $n = c_0c_1 \dots c_k$ composto da k cifre $c_i \in 0, \dots, 9$, calcoli la somma delle cifre

$$s_n = \sum_{i=0}^k c_i$$

in modo *ricorsivo*.

Per esempio:

- se $n = 1234$ allora $s_n = 10$;
- se $n = 2435$ allora $s_n = 14$;

Es. A3 (6 punti)

Dato un insieme di n valori x_1, \dots, x_n definiamo la seguente formula per il calcolo ricorsivo della media

$$\begin{cases} \hat{x}_1 = x_1 \\ \hat{x}_n = \hat{x}_{n-1} + \frac{x_n - \hat{x}_{n-1}}{n} \quad \text{per } n > 1 \end{cases}$$

dove \hat{x}_n e' la media calcolata quando vediamo l' n -esimo elemento, in funzione della media dell'elemento $(n-1)$ -esimo e così via.

Scrivere un programma C che dato in input un array di n elementi con i numeri x_i , calcoli la loro media usando l'equazione proposta, implementata in maniera ricorsiva.

Suggerimento.

Attenzione che l'elemento i -esimo di x (cioè x_i) corrisponde alla cella $(i-1)$ -esima nel vettore con cui programmate la soluzione. Considerate questo anche per il valore di n nella formula!

Esercizi non di sbarramento

Es. B1 (4 punti)

Si consideri questo frammento di codice C

```
int main(void) {
    int i = 6;
    int *p = &i;
    int a[3] = {0, -1, 4}; // P1

    int *s = a;
```

```

    int v = *(s+a[0]);    // P2

    v = v + *(s+2) + *p;  // P3
}

```

Si disegni lo stato del programma (ambiente, memoria, heap) al punto P1, P2 ed al punto P3 (fine del `main`).

Potete risolvere questo esercizio carta e penna, e caricare uno screenshot del vostro esercizio nella cartella B1, oppure consegnare una copia cartacea della vostra soluzione.

Es. B2 (3 punti)

Si consideri questo frammento di programma C

```

int f(int * a, int to){

    if (to == 0)
        return *a;

    else return *(a+to) + f(a, to-1);
}

int main(void) {

    // carica vettore di N elementi (come con la nostra libreria)
    int N;
    double * x = read_double_array(argv[1], &N);

    int * y = (int *) malloc(size(int) * N); // N elementi

    for(int i = 0; i < N-1; i++)
    {
        y[i] = f(x, i);
    }

    return 0;
}

```

- Si definisca la formula calcolata dal programma per ciascun elemento dell'array `y`, cioè

$$y[i] = \dots$$

e si fornisca un esempio di calcolo per un array di 4 elementi a scelta vostra;

- Si riscriva il programma eliminando la funzione ricorsiva `f` ed usando solo due `for` nel `main`.

Es. B3 (5 punti)

Si definisca in Python una classe `MyPower` che:

- prende un valore $a > 0$ intero passato tramite costruttore
- implementi un *iteratore* che genera il flusso dei valori $f(n) = 3^{n+1}/n$ con $0 \leq n \leq a$, n intero.
- fornisce un metodo ulteriore `mean_pow_a(self)` che restituisce il valore medio calcolato da tutte le potenze generabili dall'iteratore appena definito (i.e., la media dei valori $f(0), f(1), \dots, f(a)$).