Esame di Programmazione (mod A) - CdL AIDA

Appello III - Giugno 2022

Giulio Caravagna (gcaravagna@units.it)

A1. Si considerino i due insiemi di numeri naturali $A = \{x \mid i \leq x < j\}$ e $B = \{x \mid t < x \leq u\}$. Si scriva una funzione *iterativa* `set_size` che prenda in input i 4 interi positivi i, j, t, u e calcoli la cardinalita di $C = (A \cup B) \setminus (A \cap B)$, utilizzando una funzione ausiliaria `is_inside(x, y, z)` che restituisce 0 se $x \in [y, z]$, e -1 altrimenti.

Si mostrino gli elementi di A, B e C per $i = 2, j = 5, t = 4, u = 7$, ed il risultato della `set_size`.

A2. Si scriva un programma C *iterativo* che calcoli, per un dato $n \geq 3$ in input, la successione di interi

$$\begin{cases} a_1 = -1 \\ a_2 = 0 \\ a_n = (7 + a_{n-1} - 1)/n \quad \text{con } n \geq 3 \quad \text{se } a_{n-1} \geq 2a_{n-2} \\ a_n = (7 - a_{n-1} - 1)/(n-1) \quad \text{con } n \geq 3 \quad \text{se } a_{n-1} < 2a_{n-2} \end{cases}$$

A3. Si consideri la successione

$$F_0 = 1 \qquad F_1 = 3 \qquad F_n = 1 - \frac{(n-1)F_{n-2}}{2F_{n-1}} \qquad n \geq 2$$

e la quantità $\boldsymbol{F} = \prod_{i=1}^{N} F_{x_i}$ calcolata a partire da un *array* $\boldsymbol{x}$ *di* N *valori non negativi* $x_1, x_2, \ldots, x_N$.

Si scriva un programma C che dato $\boldsymbol{x}$ calcoli $\boldsymbol{F}$ *iterativamente* e ogni F_i *ricorsivamente*. Si riporti, senza calcolarla, la forma ansalitica esplicita di F_4.

B1. Si presenti la definizione di lista linkata in C, ed

- una funzione per creare tale lista aggiungendo elementi in coda,
- una funzione ricorsiva per il calcolo del numero di numeri pari nella lista.

B2. Si rappresenti la memoria del seguente programma C ai punti `A` e `B`, per ciascun ciclo di esecuzione del `for`.

```c
int x = 6;
int * y = (int *) malloc(sizeof(int));
*y = x;

for(int i = 0; i < *y; i++)
{
  // A
  if(i %2 != 0)
```

```
    {
      int x = *y;
      i = x + 1;
      // B
    }
    else
    {
      x = x * *y;
    }
  }
```

B3. Si considerino le seguenti classi

```
class C1:
  def __init__(self, i):
    self.i = 1

  def f(self)
    return self.i * self.i

  def g(self, x)
    return self.i * x + f()

class C2:
  def __init__(self, i):
    self.i = 1

  def f(self)
    return self.i * self.i + self.i

  def g(self, x)
    return self.i * x
```

Si riscrivano le due classi `C1` e `C2` utilizzando l'ereditarieta in Python, senza alterare il calcolo effettuato da `f` e `g`.