

Esame di Informatica Generale I

Mauro Brunato, Andrea Delai

Martedì 10 febbraio 2004

Esercizio 1

Quali sono i principali tipi numerici in C? Quanta memoria occupano e che range di valori possono rappresentare?

Suggerimento — *Ricordare che anche i caratteri sono rappresentati numericamente!*

Esercizio 2

Scrivere una funzione ricorsiva

int potenza (int m, int n)

che, dati due interi non negativi m e n , restituisce m^n .

Tale funzione *deve* basarsi sulla seguente formulazione induttiva ($m, n \in \mathbb{N}$):

$$m^n = \begin{cases} 1 & \text{se } n = 0; \\ m^{\frac{n}{2}} \cdot m^{\frac{n}{2}} & \text{se } n \text{ è pari}; \\ m^{1+\lfloor \frac{n}{2} \rfloor} \cdot m^{\lfloor \frac{n}{2} \rfloor} & \text{se } n \text{ è dispari.} \end{cases}$$

Esercizio 3

Si consideri il seguente programma:

```
1. #include <stdio.h>
2. int a;
3. int f (int *b)
4. {
5.     int c;
6.     for ( c = 0; c < a; c++ )
7.         (*b) -= c;
8.     return c;
9. }
10. int main (void)
11. {
12.     int b;
13.     a = 9;
14.     b = f (&a);
15.     printf ("%d %d\n", a, b);
16.     return 0;
17. }
```

3.1) Quali sono gli intervalli di esistenza e di visibilità delle variabili dichiarate nel codice?

3.2) Che cosa scrive il programma, e perché?

Un vettore $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ si dice *supercrescente* se ogni suo elemento è maggiore della somma di quelli che lo precedono:

$$\forall i = 1, \dots, n \quad x_i > \sum_{j=1}^{i-1} x_j.$$

Considerato nullo il valore della sommatoria vuota, è chiaro che il primo termine dev'essere positivo, e così tutti gli altri per induzione. È altrettanto chiaro che un vettore supercrescente è ordinato.

Esercizio 4

4.1) Scrivere una funzione

```
int supercrescente (double v[], int n)
```

che accetti come parametro un vettore `v` di numeri `double` e un intero `n` rappresentante il numero di elementi del vettore. La funzione restituisce vero se e solo se il vettore `v` di `n` elementi è supercrescente.

4.2) Utilizzando ove opportuno una chiamata alla funzione definita al punto 4.1, scrivere una funzione `main` che chiede la lunghezza di un vettore di `double`, lo alloca dinamicamente, lo riempie di valori letti al terminale, dice se è supercrescente oppure no, infine libera la memoria.

Esercizio 5

5.1) Supponiamo per semplicità che due parole realizzino una *rima* se e solo se hanno almeno gli ultimi tre caratteri uguali. Scrivere una funzione che accetta a parametro due stringhe e che restituisce vero se e solo se le parole contenute nelle due stringhe rimano.

5.2) Scrivere un programma principale che utilizza la funzione scritta al punto 5.1 per dire all'utente se due parole da lui inserite rimano o no.