

Università degli Studi di Cagliari

FACOLTA' DI INGEGNERIA E ARCHITETTURA Tutorati Elementi di Informatica

Esercitazione 5: Funzioni e Puntatori

Elettrica, Elettronica ed Informatica

Marco Melis marco.melis@unica.it

Tutor: Marco Melis marco.melis@unica.it

Funzioni

Costante matematica e (con funzioni)

- Provateci voi:
 - Considerando che il valore della costante matematica e (\sim 2.71828...) può essere espresso come una serie infinita:
 - e = 1 + 1/1! + 1/2! + 1/3! + ... + 1/n!
 - Scrivere una programma che calcoli attraverso apposite funzioni il valore di e approssimando all'n-esimo termine della serie, con n definito dall'utente
 - La funzione calcolo_termine deve occuparsi di calcolare il singolo termine della serie
 - La funzione calcolo_e deve occuparsi di sommare i termini
 - Il valore della costante deve essere stampato nella funzione main
- Suggerimenti:
 - Utilizzare un array di decimali per memorizzare i singoli termini

Tutor: Marco Melis marco.melis@unica.it

Costante matematica *e* (con funzioni)

```
#include<stdio.h>
 3 ⇆
      float calcolo termine(int num);
  '≒
      float calcolo e(float termini[], int dim);
 6
       int main(){
           float e;
           int n;
10
11
           printf("inserire il numero di termini: ");
12
           scanf("%d", &n);
13
14
           float termini[n];
15
16
           int i;
17
           for (i=0; i < n; i++){
18
               termini[i] = calcolo termine(i+1);
19
20
21
           e = calcolo e(termini, n);
22
23
           printf("Il valore di e risulta: %f", e);
24
           return 0;
25
26
```

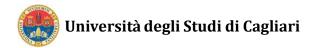
Costante matematica *e* (con funzioni)

```
28 \( \daggerightarrow \forall float \) calcolo termine(int num){
29
30
            int den = num;
31
            while ((num - 1) > 0)
                 den = den * (num - 1);
32
33
                 num - - ;
34
35
36
            return 1.0 / den;
37
38
39
      ≒float calcolo e(float termini[], int dim){
   ≒.
40
            int i;
41
            float e = 1;
42
43
            for(i = 0; i < dim; i++){}
44
                 e += termini[i];
45
46
47
            return e;
48
```

Funzioni e Puntatori

Statistiche partite (con funzioni)

- Provateci voi:
 - Si vuole realizzare un programma che stampi le statistiche dei goal fatti e goal subiti per una lista di squadre di calcio
 - L'utente inserirà un numero indefinito di partite (può inserire 0 per terminare l'inserimento o 1 per continuare)
 - Ogni **squadra** è identificata da **4 campi**:
 - Nome, codice, goal fatti, goal subiti
 - Ogni partita è identificata da 4 campi:
 - codice squadra1, codice squadra2, goal squadra 1, goal squadra2
 - La funzione void aggiorna_statistiche(struct partita elem, struct squadra *lista, int num_squadre) si occupa di aggiornare il conto dei goal fatti/subiti ad ogni inserimento
 - Utilizzare un array di strutture per registrare le statistiche
- Inizializzazione:
 - Juventus (codice 5), Milan (codice 3), Inter (codice 9)



Tutor: Marco Melis marco.melis@unica.it

```
#include <stdio.h>
       struct squadra {
           char nome[10];
           int codice;
           int goal fatti;
 6
           int goal subiti;
8
      ֆ};
9
       struct partita {
           int cod s1;
           int cod s2;
           int goal s1;
           int goal_s2;
14
      }};
16
      void aggiorna statistiche(
17 5
               struct partita elem, struct squadra *lista, int num squadre);
18
```

Funzioni e Puntatori

```
⇒void main(){
20
21
           struct squadra lista squadre[] = {
22
                    {"Juventus", 5, 0, 0},
23
                    {"Milan", 3, 0, 0},
24
                    {"Inter", 9, 0, 0}
25
26
27
           int num squadre = 3;
28
29
           struct partita elem;
30
31
           int scelta = 1;
32
           while(scelta != 0){
33
               printf("Inserisci i dati di una partita...\n");
34
35
               printf("Codice squadra 1: ");
36
               scanf("%d", &elem.cod s1);
37
```

```
printf("Codice squadra 2: ");
39
               scanf("%d", &elem.cod s2);
40
41
               printf("Goal squadra 1: ");
42
               scanf("%d", &elem.goal s1);
43
44
               printf("Goal squadra 2: ");
45
               scanf("%d", &elem.goal s2);
46
47
               aggiorna statistiche(elem, lista squadre, num squadre);
48
49
               printf("Se vuoi inserire altre partite digita 1, altrimenti 0: ");
50
               scanf("%d", &scelta);
51
52
53
54
55
           int i;
           for (i=0; i < num squadre; i++){}
56
               printf("%s: goal fatti %d, goal subiti %d\n",
57
                       lista squadre[i].nome,
58
                      lista squadre[i].goal fatti,
59
                       lista squadre[i].goal subiti);
60
61
62
63
```

```
void aggiorna statistiche(
65 5
               struct partita elem, struct squadra *lista, int num squadre){
66
67
           int i;
68
69
           for (i=0; i < num squadre; i++){}
               struct squadra *s = lista+i;
70
               if (s->codice == elem.cod s1){
72
                   s->goal fatti += elem.goal s1;
                   s->goal subiti += elem.goal s2;
               }else if (s->codice == elem.cod s2){
                   s->goal fatti += elem.goal s2;
75
76
                   s->goal subiti += elem.goal s1;
77
78
79
80
```