

Laboratorio di  
Fondamenti di Informatica  
5/11/2015

Anno accademico 2015/2016

# 1. Che anno è, che giorno è?

Ci ispiriamo alla funzione **strcmp**( ... ).

Si codifichi una funzione **datecmp**( ... ) che stabilisce se due date, passate come parametri, sono uguali (nel qual caso restituisce 0), oppure in ordine cronologico (e allora restituisce un numero negativo), oppure in ordine cronologico inverso (restituendo un numero positivo).

Si codifichi inoltre una funzione che **stampa** una data nel formato gg/mm/aaaa.

Si scriva anche un piccolo programma (main) per verificare che le funzioni siano state realizzate correttamente. Si utilizzino le seguenti definizioni:

```
typedef struct { int giorno;
                int mese;
                int anno; } Data;

int confronta( Data d1, Data d2 ) { /* ... */ }

void stampa( Data d ) { /* ... */ }

void stampaConLettere( Data d )
```

Che stampa le date in formato            15 Mar 44 a.C.        13 Ott 2013 d.C.

## Esercizio 2 – A volte ritornano...

Riconsideriamo (a scelta, secondo il gusto individuale!) qualcuno degli esercizi assegnati nei precedenti laboratori, e domandiamoci quali funzioni avremmo potuto definire per rendere più compatta e razionale la loro soluzione. Idealmente, si parta dal codice scritto allora, e lo si "adatti". Si codifichino (e si testino opportunamente tramite main) ad esempio le funzioni seguenti.

```
void fattprimi(int n)    Che stampa la scomposizione in fattori primi di n
void tabellina(int n)    Che stampa la tabellina di n
void floyd(int n)        Che stampa un triangolo di Floyd di n righe
int isPrimo(int n)        Che restituisce 1 se n è un numero primo, 0 altrimenti
int quantescompinquadperf(int n) Restituisce il numero di diversi modi in cui n si può
                               scomporre in quadrati perfetti: 0 se non è scomponibile (ad es. n=28), 2 per il 50, 6 per il 5525...
int divcom(int n1, int n2)
    che stampa tutti i divisori diversi da 1 comuni a n1 e n2, e restituisce il numero di tali divisori
int isPadovan(int n)      Che restituisce 1 se n è un numero di Padovan, 0 altrimenti
int isLucky(int n)        Che rest. 1 se n (< N fissato a priori) è un numero fortunato, 0 altrimenti
int wordAnagrams(char p1[], char p2[])
    che restituisce 1 se p1 e p2 sono anagrammi semplici, 0 altrimenti
int phraseAnagrams(char p1[], char p2[])
    che restituisce 1 se p1 e p2 sono anagrammi a frase, 0 altrimenti
```

## Esercizio 3 : ...e volte altre ne arrivano

- `int divcom(int n1, int n2, int blabla)` che trova tutti i divisori diversi da 1 comuni a `n1` e `n2`, e restituisce il numero di tali divisori. Se `blabla` vale 0 non stampa nulla, altrimenti i divisori vengono anche stampati a video.
  - `...pfl(...)` che restituisce 1 se due numeri interi sono primi tra loro, 0 altrimenti. **Si riusi la funz. `divcom()` !**
  - `int contaPP(int v[])` che, dato un vettore di interi positivi, restituisce il numero di valori del vettore che sono sia primi sia numeri di Padovan (ovviamente riusando le funzioni `isPrimo()` e `isPadovan()` precedentemente definite).
  - `...tuttiPFL(...)` che, dato un vettore di interi positivi, restituisce 1 se i numeri contenuti sono **tutti** primi tra loro (accoppiandoli in tutti i modi possibili, **ovviamente riusando `pfl()`** ).
  - `int sommaarray(int v[], int d)` che riceve come parametri un array `v` di interi e la sua dimensione `d`, e restituisce la somma dei suoi elementi
  - `int maxsottseq(int seq[], int d, int n)` che riceve un vettore di interi, la sua dimensione ed un intero `n` compreso tra 1 e 9, e restituisce la somma degli elementi della sottosequenza lunga `n` che, tra tutte le sottosequenze contigue di `seq` lunghe `n`, ha **somma massima**. (**Si riusino le funzioni `scorri()` e `sommaarray()`** )
- Suggerimento:** si dichiarino nel main opportuni dati di test (non li si legga da stdin!)

```
int test1[10] = {13,27,2,28,6,9,55,39,37,11},  
    test2[10] = {97,53,73,37,11,7,28,51,19,5};
```

# 1.bis Che anno è, che giorno è?

Si estenda il supporto alle date con le seguenti funzioniconodifichino (e si testino con il main) anche le funzioni

```
int valida( Data d )
```

Verifica se d è una data valida (ha un significato nel mondo reale) Esempi:

**non** il 32 gennaio, **non** il 30 febbraio, **non** il 31 novembre, **non** il 7 ottobre 1582

**ok** il 16 marzo 1354, **ok** il 9 giugno 1996, **ok** il 28 febbraio 2304

```
int distanza( Data d1, Data d2 )
```

Che calcola e restituisce il numero di giorni trascorsi da d1 a d2. Ci si ricordi che alcuni problemi sono già stati risolti (esempio: se un anno è bisestile... )

```
Data aggiungi( Data d, int giorni )
```

Che calcola e restituisce la data corrispondente al **giorni** dopo la data **d**

# Giochino 1. Tic Tac Toe

Si realizzi il gioco del tris. Si mostri a video una griglia 3 x 3 inizialmente vuota e si chieda a turno a ciascun giocatore di fare la sua mossa. Il gioco termina quando uno dei giocatori fa tris oppure non ci sono più celle libere (parità). In particolare, per rendere semplice e ordinata la soluzione, si ricorra alla definizione di opportuni tipi e funzioni (specificati nella slide successiva), di modo che il main sia:

```
int main() {
    Griglia g;
    char player1[]="Alice", player2[]="Bob";
    int game_over = 0;
    inizializza( g );
    stampa( g );
    while( ! game_over ) {
        game_over = mossa( g, player1, 'O' );
        if( ! game_over )
            game_over = mossa( g, player2, 'X' );
    }
}
```

```

typedef char Griglia[3][3];

void inizializza( Griglia g ); //Setta tutte le caselle a ''

void stampa( Griglia g ); //Stampa la griglia

int controlla_vincitore( Griglia g, char ch );
    // restituisce 1 se c'è un tris di ch, 0 altrimenti

int conta_vuote( Griglia g );
    // restituisce il numero di caselle vuote in g

int mossa( Griglia g,
           char nome[],
           char simbolo );
    // Chiede al giocatore nome la casella dove
    // apporre il suo segno (simbolo), verificando
    // che sia una casella valida (coordinate valide
    // e casella vuota), marca la griglia con il
    // simbolo, stampa la griglia modificata,
    // controlla se nome ha vinto (nel qual caso
    // stampa un messaggio di congratulazioni),
    // controlla anche se per caso la partita non
    // sia finita in pareggio, e restituisce 1 se la
    // partita è terminata (con vittoria o
    // pareggio), 0 se non è terminata.

```

```

D:\Documents\Didattica\Info1-AUT\2013-2014\L...
x \ y  0  1  2
0  |  |  |  |
1  |  |  |  |
2  |  |  |  |
-----
Mossa di Alice [x y] : 4 2
Mossa non valida. Ripeti la mossa [x y] : 0 2
x \ y  0  1  2
0  |  |  | O |
1  |  |  |  |
2  |  |  |  |
-----
Mossa di Bob [x y] : 1 1
x \ y  0  1  2
0  |  |  | O |
1  |  | X |  |
2  |  |  |  |
-----
Mossa di Alice [x y] :

```