

## Tecniche di Programmazione (2018/19)

# Esercitazione 3

### Argomenti: funzioni e passaggio dei parametri

Scrivere la funzione C

```
int* allocaInt();
```

che alloca dinamicamente in memoria un intero e ne restituisce in uscita il puntatore.

### Esercizio 3.2

Scrivere la funzione C

```
void printInt(int *i1, int *i2);
```

che, dati in ingresso due puntatore ad intero **i1** e **i2**, ne stampi a schermo il valore

### Esercizio 3.3

Scrivere la funzione C

```
int MCD(int i1, int i2);
```

che, dati in ingresso due interi **i1** e **i2**, ne calcoli, e stampi a schermo, il massimo comun divisore

### Esercizio 3.4

Scrivere la funzione C

```
int mcm(int *i1, int *i2);
```

che, dati in ingresso due interi **i1** e **i2**, ne calcoli, e stampi a schermo, il minimo comune multiplo.

## Esercizio 3.5

Scrivere la funzione C

```
void conversioneTemperatura(int t, char c);
```

che, dati in ingresso un intero `t` e un char `c` rappresentanti rispettivamente un valore di temperatura e la scala di temperatura scelta ed effettui la conversione nelle altre scale e ne stampi a schermo il risultato. Le scale da considerare sono Celsius (carattere "C"), Kelvin (carattere "K") e Fahrenheit (carattere "F"). Formule di conversione:

$$K = C + 273.15$$

$$F = C * 9/5 + 32$$

## Esercizio 3.6

Scrivere la funzione C

```
void* conversioneTemperatura(int *t, char c);
```

che, dati in ingresso un puntatore ad intero `t` e un char `c`, svolga lo stesso compito della funzione precedente. Le due conversioni risultanti (ovvero due valori di temperatura, e due caratteri rappresentati la scala) devono essere salvate in una zona di memoria allocata dinamicamente con un'unica chiamata alla funzione `malloc()` e restituite in uscita.

## Esercizio 3.7

Scrivere la funzione C

```
void printConversione(void *temperatura);
```

che, dato in ingresso un puntatore a `void`, stampi il risultato della funzione precedente.

## Esercizio 3.8

Scrivere la funzione C

```
void soluzioneSistemaLineare(int i1, int i2);
```

che, dati in ingresso due interi `i1` e `i2`, risolva l'equazione:

$$i1 \cdot x + i2 = 0$$

e ne stampi il risultato a schermo.

## Esercizio 3.9

Scrivere la funzione C

```
void differenzaPuntatori(int *i1, int *i2);
```

che, dati in ingresso due puntatore ad intero `i1` ed `i2`, ne calcoli la distanza in memoria (tramite differenza di puntatori) e, successivamente, modifichi il contenuto di `i2` scrivendo una espressione che contiene solo il puntatore a `i2` e la differenza in memoria tra `i1` e `i2`.

## Esercizio 3.10

Scrivere la funzione C

```
void fibonacci(int N);
```

che, dato in ingresso un intero `N`, calcoli e stampi i primi `N` numeri della serie di Fibonacci. La serie di Fibonacci inizia con 1, 1 ed ogni numero successivo è dato dalla somma dei due precedenti: 1, 1, 2, 3, 5, 8, 13, 21, . . .

## Altri Esercizi Proposti

Creare un programma che sulla base di una selezione fatta attraverso l'input di un carattere consenta all'utente di acquisire i parametri di input ed invocare le funzioni sui numeri interi definite in precedenza (3.10, 3.4, 3.3) e stampi i rispettivi risultati. L'input e l'output dei risultati dovranno essere realizzati da opportune funzioni.