

# Introduzione al C

Corso di Fondamenti di Informatica  
Ingegneria Clinica

## Esercitazione 4

DIPARTIMENTO DI INFORMATICA  
E SISTEMISTICA ANTONIO RUBERTI



SAPIENZA  
UNIVERSITÀ DI ROMA

**Domenico Daniele Bloisi**

**Raffaele Nicolussi**

# Esercizio numero da 1 a 100

---

- Scrivere un programma che stampi i numeri:
  - Da 1 a 100
  - Da 100 a 1
  - Da X a Y (X e Y qualsiasi)

# Esempio: Stampa numeri da 1 a 100

---

```
int i ;  
for ( i = 1 ; i <= 100 ; i = i + 1 )  
    printf( "%d", i ) ;
```

equivale a

```
int i ;  
i = 1 ;  
while ( i <= 100 ) {  
    printf( "%d", i ) ;  
    i = i + 1 ;  
}
```

# Funzioni sui caratteri

---

- Scrivere la funzione
  - int **codicecarattere** (char c)  
che dato un carattere c ne stampi il suo codice ASCII
  - char **carattere** (int codice)  
che dato un codice ne stampi il carattere relativo
  - void **stampacaratteritraXeY** (int x, in y)  
che stampi tutti i caratteri compresi tra i codici X e Y

# Esempio

---

- Stampare il codice numerico dei caratteri compresi tra 33 e 85

```
for (int i = 33; i <= 85; i++)  
    printf("i = %d → c = %c", i, i);
```

# Fattoriale di un numero

---

$$\text{Fattoriale } (n) = n(n-1)(n-2) \dots 1$$

$$\text{fattoriale } (0) = 1$$

$$\text{fattoriale } (n) = n * \text{fattoriale } (n-1)$$

- *Esempio:* fattoriale (4) = 4\*3\*2\*1

# Fattoriale

---

- Scrivere la funzione
  - int **fattoriale** (int N)  
che restituisca il fattoriale del numero N

# Esempio: calcolo del fattoriale

---

```
#include <stdio.h>
```

```
int main () {  
    int n, i ;  
    unsigned long int p = 1;  
  
    printf("Inserire numero intero maggiore di zero: ");  
    scanf("%d", &n);  
    for ( i = 1 ; i <= n ; i++)  
        p =p* i;  
    printf("\n%d != %lu\n", n, p);  
    return 0;  
}
```



# Esercizio 1: Stampa max e min

---

Leggere da standard input una sequenza di 12 voti (tra 0 e 30) e stampare il massimo ed il minimo dei voti

Analisi dei dati (variabili da usare!)

Quattro **variabili** di tipo **unsigned short int**: contatore (**i**), voto corrente (**voto**), minimo (**min**) e massimo (**max**)

Algoritmo

Inizializza **min** e **max** (attenzione!)

*Per ogni voto letto*

se **min** è maggiore di **voto**, allora

**min = voto**

se **max** è minore di **voto**, allora

**max = voto**

Stampa **min** e **max**



# Soluzione 1

---

```
#include <stdio.h>
#include <stdlib.h>
#define NUM_VOTI 12

int main () {
    int voto=0, min, max, i = 1;

    printf("\nInserire voto %d: ", i);
    scanf("%d", &voto);
    printf("Voto inserito: %d\n", voto);

    while ( voto > 30 || voto < 0 ) {
        printf("Voto non valido, inserire voto %d: ", i);
        scanf("%d", &voto);
        printf("Voto inserito: %d\n", voto);
    }
}
```

# Soluzione 1

---

```
i = i + 1;
min = max = voto;
while ( i <= NUM_VOTI ) {
    printf("\nInserire voto %d: ", i);
    scanf("%d", &voto);
    printf("Voto inserito: %d\n", voto);
    while ( voto > 30 || voto < 0 ) {
        printf("Voto non valido, inserire voto %d: ", i);
        scanf("%d", &voto);
        printf("Voto inserito: %d\n", voto);
    }
    if ( voto > max )
        max = voto;
    if ( voto < min )
        min = voto;
    i = i + 1;
}

printf("Massimo = %d, Minimo = %d\n", max, min);
return 0;
}
```

# Esercizio 2

---

Leggere da standard input una sequenza di caratteri terminata da invio e determinare il numero di occorrenze delle vocali all'interno della sequenza

```
Inserire una sequenza di caratteri terminata da invio:  
Prova di esecuzione dell'esercizio numero 2  
Numero occorrenze lettera a: 1  
Numero occorrenze lettera e: 7  
Numero occorrenze lettera i: 4  
Numero occorrenze lettera o: 4  
Numero occorrenze lettera u: 2  
Premere un tasto per continuare . . .
```

# Soluzione 2

---

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int cont_a=0, cont_e=0, cont_i=0, cont_o=0, cont_u=0;
    char carattere;

    printf("\nInserire una sequenza di caratteri terminata da invio: \n");
    scanf("%c", &carattere);

    while ( carattere != '\n' ) {
        if (carattere== 'a') cont_a = cont_a +1;
        if (carattere== 'e') cont_e = cont_e +1;
        if (carattere== 'i') cont_i = cont_i +1;
        if (carattere== 'o') cont_o = cont_o +1;
        if (carattere== 'u') cont_u = cont_u +1;
    }
    scanf("%c", &carattere);
}
```

# Soluzione 2

---

```
printf("Numero occorrenze lettera a: %d\n", cont_a);  
printf("Numero occorrenze lettera e: %d\n", cont_e);  
printf("Numero occorrenze lettera i: %d\n", cont_i);  
printf("Numero occorrenze lettera o: %d\n", cont_o);  
printf("Numero occorrenze lettera u: %d\n", cont_u);  
  
system("pause");  
return 0;  
}
```

## Esercizio 3

---

Leggere da standard input un numero intero  $N \geq 1$  e stampare la somma dei numeri da 1 a  $N$ .

```
Inserire n(>=1): 100
```

```
Risultato = 5050
```

```
Premere un tasto per continuare . . .
```

# Soluzione 3

---

```
int n, i = 1, somma = 0;
printf("\nInserire n(>=1): ");
scanf("%d", &n);

while ( n <= 0 ) {
    printf ("Inserire n (>=1): ");
    scanf("%d", &n);
}
while ( i <= n ) {
    somma = somma + i;
    i = i + 1;
}
printf("Risultato = %d\n", somma);
```



# Esercizio 4

---

Leggere da input due numeri interi n ed m e stampare tutti i numeri pari compresi tra n ed m se n è minore di m, oppure compresi tra m ed n se m è minore di n

```
Inserire n ed m: 4 71
6  8  10  12  14  16  18  20  22  24  26  28  30
32  34  36  38  40  42  44  46  48  50  52  54  56
58  60  62  64  66  68  70
Premere un tasto per continuare . . .
```

# Soluzione 4

```
int n, m, i;
```

```
printf("\nInserire n ed m: ");  
scanf("%d%d", &n, &m);
```

```
if ( n < m ) {  
    i = n + 1;  
    while ( i < m ) {  
        if ( i%2 == 0 )  
            printf ("%d ", i);  
        i = i + 1;  
    }  
}  
else {  
    i = m + 1;  
    while ( i < n ) {  
        if ( i%2 == 0 )  
            printf ("%d ", i);  
        i = i + 1;  
    }  
}
```

```
if ( n < m ) {  
    i = n + 1;  
    while ( i < m ) {  
        if ( i%2 == 0 )  
            printf ("%d ", i);  
        i = i + 1;  
    }  
}
```

```
i = m + 1;  
while ( i < n ) {  
    if ( i%2 == 0 )  
        printf ("%d ", i);  
    i = i + 1;  
}
```

# Esercizio 4 bis

---

Leggere da input due numeri interi  $n$  ed  $m$  e stampare tutti i numeri dispari compresi tra  $n$  ed  $m$  se  $n$  è minore di  $m$ , oppure compresi tra  $m$  ed  $n$  se  $m$  è minore di  $n$

# Soluzione 4 bis

---

```
int n, m, i, max;

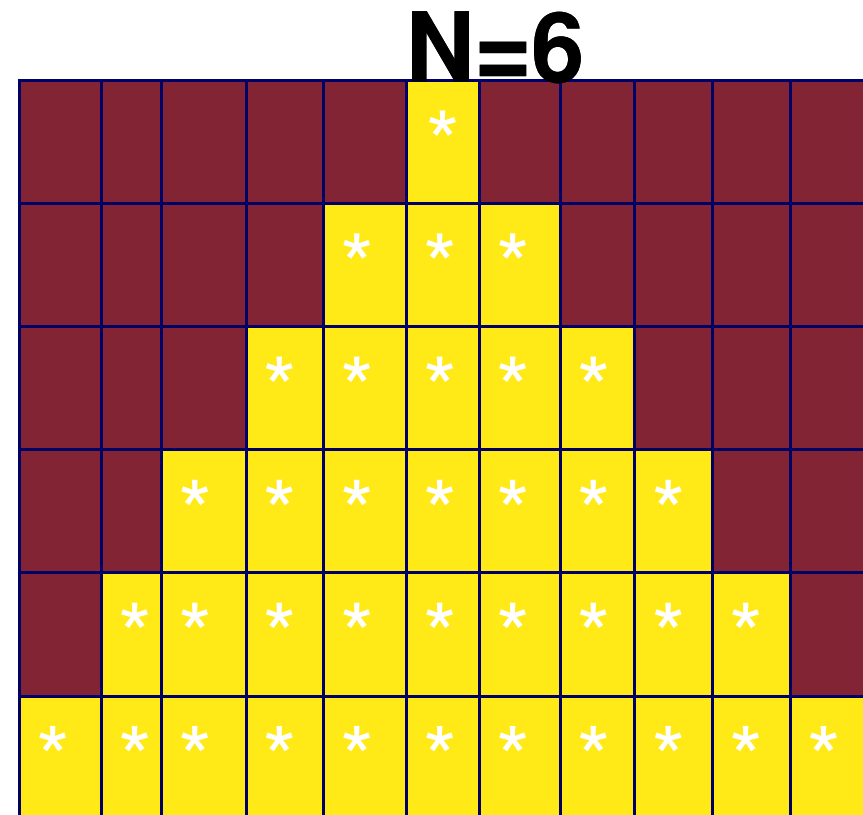
printf("\nInserire n ed m: ");
scanf("%d%d", &n, &m);

if ( n < m ) {
    max = m;
    i = n + 1;
}
else {
    max = n;
    i = m + 1;
}

while ( i < max ) {
    if ( i%2 != 0 )
        printf ("%d ", i);
    i = i + 1;
}
```

# Piramide di asterischi

- Leggere un intero compreso fra 1 e N e stampare una piramide di asterischi di altezza pari h al numero letto
- **Asterischi sulla riga: numero\_riga \* 2 - 1**
- 1 nella prima riga
- 3 nella seconda riga
- 5 nella terza riga
- 7 nella quarta
- 9 nella quinta
- 11 nella sesta
- **Spazi bianchi: h - numero\_riga**
- 5 nella prima riga
- 4 nella seconda riga
- 3 nella terza riga
- 2 nella quarta
- 1 nella quinta
- 0 nella sesta



---

Finché il numero delle righe non è pari all'altezza data

1. Stampa gli spazi
2. Stampa gli asterischi
3. Vai a capo

1. Stampa gli spazi

$i=1$

Finché  $i$  è minore o uguale a (altezza-riga)

Stampa uno spazio

$i=i+1$

2. Stampa gli asterischi

$j=1$

Finché  $j$  è minore o uguale a ( $\text{riga}^2-1$ )

Stampa asterisco

$j=j+1$

# Esercizio

---

- Realizzare un programma C che converta un intero decimale in binario attraverso la tecnica delle divisioni successive
- Algoritmo base
  1. Leggi il numero da convertire
  2. Calcola le varie cifre binarie del numero convertito attraverso un ciclo
  3. Stampa il numero convertito
- Problemi:
  - Il numero mi viene all'incontrario ...

# Sub – problema

---

- Dato un numero scomposto nelle sue cifre e fornite dalla meno significativa a quella più significativa, memorizzarlo in una variabile e stamparlo
  - Input da tastiera:
    - 4 2 7 2 9 7 4 0 7
  - Output a video: 704792724



# Algoritmo

---

- Come si fa?
  - Moltiplicatore = 1, numero\_finale = 0;
  - CICLO
    - Trovo la cifra\_attuale;
    - Numero\_finale = numero\_finale + cifra\_attuale \* moltiplicatore
    - Moltiplicatore = moltiplicatore \* 10;
  - FINE CICLO

# Soluzione completa (1)

---

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int numero, resto, bit, risultato=0;
```

```
    int moltiplicatore = 1;
```

```
    printf("Inserisci un numero decimale: ");
```

```
    scanf("%d", &numero);
```

# Soluzione completa (2)

---

```
while (numero>0)
{
    bit = numero % 2;
    numero = numero / 2;

    risultato= risultato + (moltiplicatore * bit);

    moltiplicatore = moltiplicatore * 10;

}

printf("Il numero in binario e': %d\n", risultato);
system ("Pause");

return 0;
}
```