

# Introduzione al C

Corso di Fondamenti di Informatica  
Ingegneria Clinica

## Esercitazione 5

DIPARTIMENTO DI INFORMATICA  
E SISTEMISTICA ANTONIO RUBERTI



SAPIENZA  
UNIVERSITÀ DI ROMA

**Domenico Daniele Bloisi**

**Raffaele Nicolussi**

# 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;
}
```

# 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;  
}
```

# Piramide di asterischi

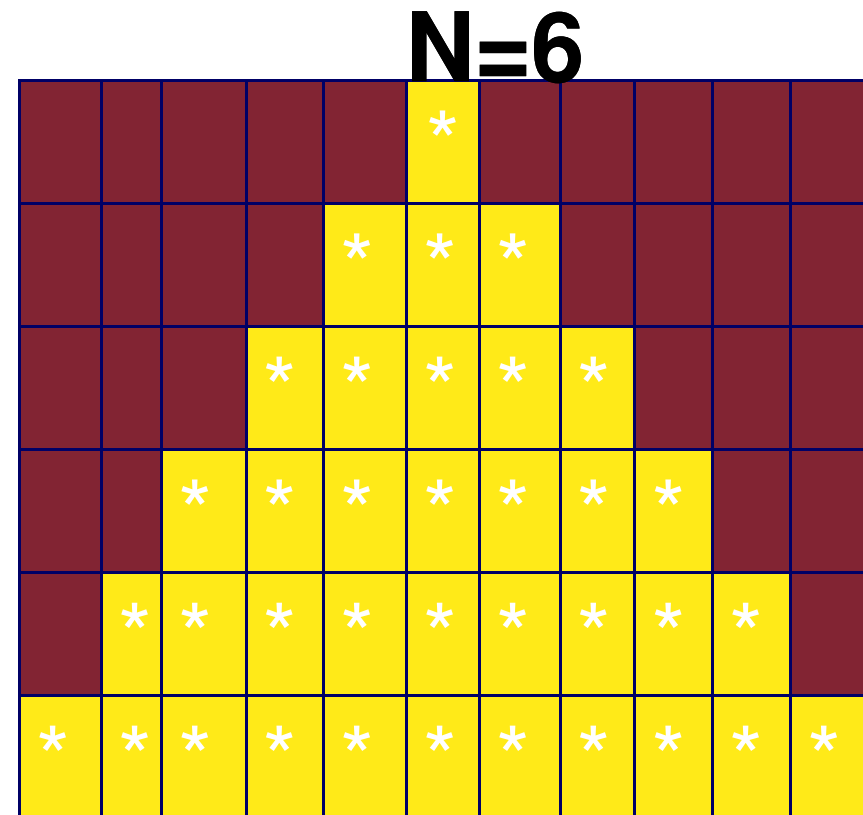
- Leggere un intero compreso fra 1 e N e stampare una piramide di asterischi di altezza pari h al numero letto

- **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 (riga\*2-1)

Stampa asterisco

$j=j+1$