

# Fondamenti di Informatica L-A

## Esercitazione del 30/10/07

Ing. Giorgio Villani



## Ripasso Teoria

- Ciclo do ... while
  - do {istruzioni} while(espressione);

```
int i;
do {
    printf("Scegli un numero tra 1 e 3");
    scanf("%d", &i);} while (i<1 || i>3);
```

- Le istruzioni all'interno del blocco 'do' vengono eseguite la prima volta senza verificare l'espressione del while.
- Il ciclo termina quando l'espressione non è più verificata.

# Ripasso Teoria

## ■ Ciclo while

□ while(espressione) {istruzioni}

```
int i = 1;
while(i<=10){
    printf("3 * %d = %d", i, 3*i);
    i++;}
```

# Ripasso Teoria

## ■ Switch

□ switch(variabile){  
    case valore\_variabile\_1: istruzioni...; break;  
    case valore\_variabile\_2: istruzioni...; break;  
    ...}

```
char car;
printf("Digitare y, n, ?\n");
scanf("%c",&car);
switch(car){
    case 'y': printf("yes"); break;
    case 'n': printf("no"); break;
    case '?': printf("forse"); break;}
```

# Ripasso Teoria

## ■ break

- Uscita dal ciclo

```
int a, b, i=0;
while(i<5){
    printf("Inserire due interi\n");
    scanf("%d %d",&a, &b);
    if(a>b){
        printf("a maggiore di b\n");
        break;
    }
    i++;
}
// il ciclo termina quando a>b (break) oppure
// sono state inserite 5 coppie di valori
```

# Ripasso Teoria

## ■ continue

- Ignora le istruzioni restanti del ciclo corrente e passa al ciclo successivo

```
int a, b, i=0;
while(i<5){
    printf("Inserire due interi\n");
    scanf("%d %d",&a, &b);
    if(a>b){
        printf("a maggiore di b\n");
        continue;
    }
    i++;
}
// il ciclo termina quando sono state inserite
// 5 coppie di valori tali che b<=a
```

# Esercizio 1

- Scrivere un programma che legge in input un intero e scrive in output “Inserito numero pari!” se il numero letto è pari. Il programma continua a leggere numeri fino a quando non è stato inserito un numero pari.
- Utilizzare **do-while**

## Soluzione

```
#include <stdio.h>

void main(){
    int i;
    do{
        printf("Inserire numero pari\n");
        scanf("%d", &i);
        i= i%2;} while(i != 0); /*si continuano a leggere numeri
                                fino quando il numero letto è
                                dispari */

    printf("Inserito numero pari!\n");
}
```

# Esercizio 2

- Scrivere un programma che legga in input dei numeri interi. Il programma termina dopo che sono stati digitati MAX=10 interi o se viene digitato '0'.
- Utilizzare **while**.
- Definire MAX come una variabile globale

## Soluzione

```
#include <stdio.h>

#define MAX 10 // ATTENZIONE: non scrivere MAX = 10; → 2 errori!!
                                                     ↑   ↑
void main(){

    int i=0, num; // 'i' conta quanti numeri sono stati letti

    printf("Digitare un numero intero o 0 per terminare\n");
    scanf("%d",&num);

    while((i < MAX-1) && (num != 0)){ /* MAX-1 → il primo numero è
                                             stato letto prima del ciclo*/
        printf("Digitare un numero da inserire o 0 per terminare\n");
        scanf("%d",&num);
        i++;
    }
}
```

# Soluzione alternativa

```
#include <stdio.h>

#define MAX 10

void main(){

    int i=0, num;

    printf("Digitare un numero intero o 0 per terminare\n");
    scanf("%d",&num);
    i++; //incremento il contatore dei numeri letti

    while((i < MAX) && (num != 0)){
        printf("Digitare un numero da inserire o 0 per terminare\n");
        scanf("%d",&num);
        i++;
    }
}
```

## Esercizio 3

- Scrivere un programma che legge in input un numero di tipo double A. L'utente digita:
  - 'e' se vuole calcolare l'esponenziale di A,
  - 'q' se vuole calcolare il quadrato di A,
  - 'f' se vuole calcolare il fattoriale di A.

Il programma restituisce il valore richiesto dall'utente.

- Utilizzare **switch**
- Includere la libreria **<math.h>** che contiene la funzione **double exp(double)**
  - es: double A, E; E=exp(A);
- NOTA: double → **%lf**

# Soluzione

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

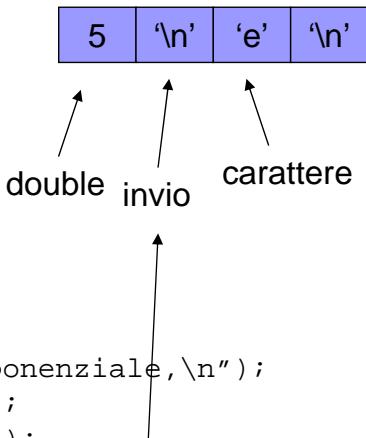
void main(){
    char c;
    int i, fatt=1;
    double A, ex;

    printf("Digitare un numero\n");
    scanf("%lf", &A);

    printf("Digitare: e per calcolare l'esponenziale,\n");
    printf("q per calcolare il quadrato,\n");
    printf("f per calcolare il fattoriale\n");
    getchar(); // legge '\n' che è contenuto nel buffer
    scanf("%c", &c); //legge il carattere digitato dall'utente

    // continua..
```

Esempio Buffer INPUT



# Soluzione

```
// ..segue

switch(c){
    case 'e': //i valori carattere sono compresi tra apici
        ex = exp(A); // libreria math.h
        printf("L'esponenziale di %lf e' %lf\n", A, ex);
        break;

    case 'q':
        printf("Il quadrato di %lf e' %lf\n", A, A*A);
        break;

    case 'f':
        for(i=2;i<=A;i++)
            fatt= fatt*i;
        printf("Il fattoriale di %lf e' %d\n", A, fatt);
        break;
}
```

# Esercizio 4

- Scrivere un programma che legga in input un numero.
  - Se il numero letto è 1 scrive in output “uno”;
  - se il numero letto è 2, 4, 6 scrive in output “pari”;
  - altrimenti scrive in output “no”.
- Utilizzare **switch** e **break**.

## Soluzione

```
#include <stdio.h>

void main(){
    int i;

    printf("Digitare un numero intero\n");
    scanf("%d", &i);

    switch(i){
        case 1: //NOTA: i valori numerici non sono compresi tra apici
                  printf("uno\n");
                  break;
        case 2:
        case 4:
        case 6:
                  printf("pari\n");
                  break;
        default:
                  printf("no\n");
    }
}
```