

C: Approfondimenti

Elementi di Informatica

Generazione numeri casuali (0-10)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int rendi_random(int min, int max);
int main() {
    int i,num,min=0,max=10;
    srand(time(0));
    for(i=0;i<=10;i++) {
        num=rendi_random(min, max);
        printf("\nNumero casuale: %d",num);
    }
    return 0;
}
int rendi_random(int min, int max) {
    int num;

    num = (rand() % (max - min + 1)) + min;
    return num;
}
```

scanf() e spazi

```
#include <stdio.h>
#define DIM 100

int main() {
    char stringa[DIM];
    scanf("%[^\\n]", stringa);
    printf("\\n%s", stringa);
    return 0;
}
```

Misurare tempo di calcolo

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

int main() {
    clock_t start,end;
    double elapsed;
    int i;
    start=clock();
    //Esempio di calcolo
    for(i=0;i<1000;i++)
        i*i;
    //Fine esempio
    end=clock();
    elapsed=((double)(end-start))/CLOCKS_PER_SEC;
    printf("\nTempo trascorso: %f",elapsed);
    return 0;
}
```

Ricorsione

Funzioni ricorsive: alternativa alla definizione di un ciclo

Una funzione chiama se stessa...

```
#include <stdio.h>
int fattoriale(int);

int main() {
    int num, fat;
    printf("\nInserisci un numero intero: ");
    scanf("%d",&num);
    fat=fattoriale(num);
    printf("\n%d!: %d",num,fat);
    return 0;
}

int fattoriale(int num) {
    if(num!=0) return (num*fattoriale(num-1));
    else return 1;
}
```

Ricorsione: nelle liste

```
void visualizza_lista(struct lista *p) {  
while(p!=NULL) {  
printf("%d ", p->num);  
p=p->next; }  
}
```

```
void visualizza_lista(struct lista *p) {  
if(p!=NULL) {  
printf("%d ", p->num);  
visualizza_lista(p->next); }  
}
```

