

<stdlib.h>

Conversione tra tipi

```
double atof(const char *str)    //Converte una stringa in un numero in virgola mobile.
                                //Equivalente a strtod(s, (char**)NULL).
int atoi(const char *str)       //Converte una stringa in un numero intero. Equivalente a
                                //(int)strtol(s, (char**)NULL, 10).
long int atol(const char *str)  //Converte una stringa in un numero intero lungo (long
                                //int). Equivalente a strtol(s, (char**)NULL, 10).
double strtod(const char *str, char **endptr) //Converte una stringa in un double
                                                //(numero a virgola mobile), effettuando dei controlli sull'overflow e restituendo anche
                                                //l'eventuale parte non convertita della stringa.
long int strtol(const char *str, char **endptr, int base) //Converte una stringa, che
                                                            //rappresenta un numero in una base arbitraria compresa tra 2 e 36, in un double (numero
                                                            //a virgola mobile), effettuando dei controlli sull'overflow e restituendo anche
                                                            //l'eventuale parte non convertita della stringa.
unsigned long int strtoul(const char *str, char **endptr, int base) //Equivalente a
                                                                    //strtol() tranne per il tipo del risultato, che è unsigned long.
```

Generazione di numeri pseudocasuali

```
int rand(void)    //Restituisce un numero intero pseudocasuale compreso tra 0 e
                  //RAND_MAX.
void srand(unsigned int seed) //Inizializza il seme per la sequenza di numeri
                              //pseudocasuali della funzione rand().
```

Allocazione e deallocazione di memoria

```
void* calloc(size_t nitems, size_t size),
void* malloc(size_t size) e
void* realloc(void *ptr, size_t size)
```

Funzioni che si occupano dell'allocazione dinamica della memoria.

```
void free(void *ptr)    //Libera la memoria allocata dinamicamente dalla famiglia di
                        //funzioni di allocazione dinamica.
```

Ricerca ed ordinamento

```
void *bsearch(const void *key, const void *base, size_t nitems, size_t size, int
              (*compar)(const void *, const void *)) //Implementa in maniera generica l'algoritmo di
              //ricerca dicotomica.
void qsort(void *base, size_t nitems, size_t size, int (*compar)(const void *, const
void*)) //Implementa in maniera generica l'algoritmo di ordinamento quicksort.
```

Matematica semplice - presenti anche in math.h

```
int abs(int x)
long int labs(long int x) //Calcola il valore assoluto dell'argomento.
div_t div(int numer, int denom),
ldiv_t ldiv(long int numer, long int denom) //Calcola il quoziente ed il resto della
divisione intera tra il dividendo ed il divisore forniti.
```