

Nozioni

exit(n)	Termino il processo in cui mi trovo
	n è il codice di uscita ()
	0 < n < 255
getpid()	Restituisce il PID del processo
kill(p, SIGKILL);	Faccio terminare immediata mente il processo con PID=p
wait()	Restituisce il PID del figlio terminato
int status; int pid = wait(&status);	Qui status prende tutte le informazioni di come è terminato il processo figlio
WIFEXITED(status)	True se il figlio è terminato normalmente
WEXITSTATUS(status)	Codice di uscita passato a exit()
WIFSIGNALED(status)	True se il figlio è terminato da un segnale
WTERMSIG(status)	Segnala quale segnale ha terminato il figlio

Tipi di dati e array

int	Intero (-1,0,1,2...)
float	Numero con la virgola
bool	True o False
char	Carattere ('a','b'...)
array	int arr[5] = {1, 2, 3, 4, 5};

Input/Output

using namespace std;	Evita di specificare std::
cin>>variabile;	Inserire a schermo un valore
cout<<variabile;	Stampa a schermo variabile
cout<<"frase";	Mostra a schermo "frase"
cout<<"..."<<endl;	Consente di andare a capo

Struttura ciclo while

```
while( con dizione == true) {
    // istruzioni eseguite
    fino a quando
    // la condizione è vera
}
```

Struttura ciclo for

```
for (int i = 0; i < n; i++){
    // ...
    // istruzioni eseguite n
    volte
}
```

Struttura funzione

```
int funzione(parametri){
    //Codice
    return 0
}
/*
Uso void se la funzione non restituisce
niente
altrimenti uso int se restituisce un intero
oppure bool o char..
*/
```

Pipe

int fd[2];	
pipe(fd);	Crea pipe: fd[0]=lettura, fd[1]=scrittura
write(fd[1], &data, size);	Scrive nel pipe
read(fd[0], &data, size);	Legge dal pipe
close(fd[0]);	Chiude lato lettura
close(fd[1]);	Chiude lato scrittura
Errore!	if(pipe(fd) == -1) { / gestione / }

Struttura If

```
if (condizione) {
    //istr uzioni
}
else {
    //istr uzioni
}
```

Condizioni

A==B	Se A è uguale a B, restituisce vero
A!=B	Se A è diverso da B, restituisce vero
A<B	Se A è minore di B, restituisce vero
A>B	Se A è maggiore di B, restituisce vero
A<=B	Se A è minore o uguale di B, restituisce vero
A>=B	Se A è maggiore o uguale di B, restituisce vero
A!B	A not B
A&- &B	A and B, vero se entrambe le condizioni sono vere
A B	A or B, vero se almeno una condizione è vera

Operatori

+	Addizione	a + b
-	Sottrazione	a - b
*	Moltiplicazione	a * b
/	Divisione	a / b
%	Modulo (resto)	a % b
++	Incremento	a++
--	Decremento	a--

Struttura classe

```
class NomeClasse {
private: // Accessibile solo dentro la classe
    int privato;
public: // Accessibile da ovunque
    int pubblico;
    // Costruttore
    NomeClasse() : pubblico(0) {}
    // Distruttore
    ~NomeClasse() { / libera memoria / }
    // Metodo
    void metodo() { cout << pubblico; }
};
```

fork()

int p = fork() Crea un processo figlio

Restituisce:

0 se sei nel processo figlio
> 0 se sono nel padre, p è il PID del figlio
- 1 se il figlio non è stato creato

Pipe + classi

```
// Per inviare oggetti via pipe:
class Studente {
    char nome[20]; // NO
    string, meglio char[]
    int voto;
};
// sizeof (St udente) funziona,
sizeof (st ring) NO
```

Librerie

```
#include <iostream>

#include <unistd.h>      per fork() ...
#include <sys/wait.h>    per wait() ...
#include <signal.h>      per kill()...
#include <cstdlib>      per rand()
```



By **zoyrox**
cheatography.com/zoyrox/

Not published yet.
Last updated 4th February, 2026.
Page 3 of 3.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopapad.com>

Random

```
srand(time(NULL));
n = rand()%max + min;
/*
n va da min fino a min + max-1
Senza min va da 0 a max-1
esempio:
da 30 a 50 faccio rand()%21+30
metto come primo numero la differenza e
come secondo numero il minimo
*/
```

Codice

```
#include <iostream>
#include <unistd.h>
#include <sys/wait.h>
using namespace std;
int dim_array=10;
int main(){
    srand(time(NULL));
    int pf[2];
    int fp[2];
    pipe(pf);
```

Codice (cont)

```
> pipe(fp);
int pid = fork();
if (pid==0){
    close(pf[1]);
    close(fp[0]);
    int array_figlio[dim_array];
    int valore;
    int array_padre[dim_array];
    int somma=0;
    read(pf[0], array_padre, sizeof(array-
_padre));
    for (int i=0; i< dim_array; i++){
        valore=array_padre[i];
        array_figlio[i]=valore;
        somma=somma+valore;
    }
    write(fp[1], &somma, sizeof(int));
    close(pf[0]);
    close(fp[1]);
    exit(0);
}
else{
```

Codice (cont)

```
> close(pf[0]);
close(fp[1]);
int array_padre[dim_array];
int somma;

for (int i = 0; i< dim_array; i++){
    array_padre[i]=rand()%101;
}
write(pf[1], array_padre, sizeof(array-
_padre));
read(fp[0], &somma, sizeof(int));
cout<<endl<<"La somma dei valori è:"
<<somma;
close(pf[1]);
close(fp[0]);
}
return 0;
}
```



By **zoyrox**

cheatography.com/zoyrox/

Not published yet.

Last updated 4th February, 2026.

Page 4 of 3.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>