

```

#include <vector>
//funzioni vector

vector <int> mioVettore; //crea un vector con elementi di tipo <"tipo">

v.push_back(elemento) //inserisce un elemento (in fondo)
v.pop_back(); //elimina ultimo elemento
v.erase(v.begin()+3); //elimina il terzo elemento e toglie gli spazi vuoti
v.erase(v.begin(), v.end()-2); //elimina gli elementi da 0 (begin) a lunghezza
//vector - 2 e toglie gli spazi vuoti
v.back(); //ritorna l'ultimo elemento
v.empty(); //se il vettore è vuoto ritorna 1 altrimenti 0
v.size(); //dimensione del vettore, numero di elementi (da usare nel for)
v.clear(); //svuota il vettore rendendolo di dimensione 0
vector<int> mioVettore (5,10); //crea un vector di 5 elementi uguali a 10

//stampa di un vector
for(int i=0; i<mioVettore.size(); ++i){
    cout<<mioVettore[i]<<endl;
}

#include <algorithm>
//funzioni algorithm
sort(v.begin(), v.end()); //ordina in modo crescente il vettore da begin a end con
//complessità nlogn
cout<<*min_element(v.begin(), v.end()); //ritorna l'elemento minimo
cout<<*max_element(v.begin(), v.end()); //ritorna l'elemento massimo
reverse(v.begin(), v.end()); //inverte l'ordine degli elementi del vector
//(utile per mettere ordine decrescente dopo aver usato sort)

//comandi terminale
gedit main.cpp & >/dev/null //avviare in modo corretto gedit con un file main.cpp
g++ -g -o main.exe main.cpp //crea il file eseguibile
./main.exe //esegue il file
./main.exe < input0.txt //se input0.txt nella stessa cartella esegue il file con i
//dati di input0
./main.exe < input0.txt | diff - output0.txt //mostra differenze tra l'output del
//proprio elaborato e output0
pwd //mostra percorso della cartella in cui ci si trova
ls //mostra i file presenti nella cartella
cd //sale nella directory superiore
cd cartella1/cartella2 //ci porta nella directory "cartella2"

```