

Code Snippets

Lettura di intere righe da file in input

Quando conosco il numero di righe:

```
ifstream myin('filepath');
int numeroLinee = 420, maxsize = 100;
char **stack = new char *[numeroLinee];
for (int i = 0; i < numeroLinee; i++) {
    stack[i] = new char[maxsize + 1];
    myin.getline(stack[i], maxsize);
} // da deallocare con delete []
```

alternativa, con eof()

```
ifstream file_a('filepath');
int numeroLinee = 100, n_caratteri = 256;
char parola[n_caratteri];
char **parole = new char *[numeroLinee];
file_a >> parola;
for (; !file_a.eof();) {
    parole[i] = new char[strlen(parola) + 1];
    strcpy(parole[i], parola);
    file_a >> parola;
}
```

Strutture file

Linked List

```
struct Nodo { // nodo dal valore int
    int value;
    Nodo *next;
};
typedef Nodo *Lista;

void creaLista(Lista &);
void stampaLista(Lista);
void deallocaTutto(Lista);
```

Tree

```
struct Node;
typedef Node *Tree;
struct Node {
    char item;
    Tree left;
    Tree right;
}
```

Sintassi

Librerie usate

- <iostream> Per lavorare con l'input e output da terminale
- <fstream> Per lavorare con l'input e output da file
- <string.h> Per lavorare con le funzioni utili per manipolare/interagire con le stringhe/char[]

File .h

```
#ifndef NAME_H
#define NAME_H
// codice
#endif
```

Puntatori

- `&variabile` Riporta il suo indirizzo
 - `int *pointer = &variabile` assegna al puntatore l'indirizzo della variabile
- NB. Se faccio `cout << pointer` avrò l'indirizzo, se invece faccio `cout << *pointer` otterrò il valore puntato.

Funzioni ricorrenti

Stringhe / `char[]`

- `char *strcpy(char *dest, const char *src)` string copy, copia la stringa `src` nella stringa `dest`. Proviene da `<string.h>`
- `int strcmp (const char * str1, const char * str2)` Proviene da `<string.h>`
 - Restituisce 0 se le stringhe sono uguali
 - Restituisce un valore `< 0` se `str1` è minore di `str2`
 - Restituisce un valore `> 0` se `str1` è maggiore di `str2`
- `int strlen(char *string)` restituisce la lunghezza della stringa. Proviene da `<string.h>`