

# Laboratorio di algoritmi e strutture dati

Docente: Violetta Lonati

giovedì 26 ottobre 2017

## 1 La vostra malloc

- Scrivete una funzione `my_malloc` che allochi memoria usando la funzione `malloc` della libreria standard (file di intestazione `stdlib.h`) e verifichi il buon esito dell'allocazione. In caso di esito positivo, la funzione deve restituire l'indirizzo dello spazio allocato; in caso contrario la funzione deve stampare un messaggio di errore e provocare la terminazione del programma attraverso una chiamata della funzione `exit`.
- Scrivete un'analoga funzione `my_realloc`.

## 2 Rovescia

Scrivete due programmi che leggano una sequenza di interi e la stampino al contrario, allocando la memoria necessaria in modo dinamico attraverso l'uso della funzione `malloc`.

1. L'input è dato da una sequenza di numeri terminata da 0; non potendo prevedere quanti numeri verranno inseriti, il vettore andrà ridimensionato man mano: partite da una dimensione fissata piccola (es: 2) e raddoppiate la lunghezza del vettore ogni volta che questo si riempie.
2. L'input è dato da una sequenza di numeri terminata da 0, come nel caso precedente; di nuovo, il vettore andrà ridimensionato man mano: partite da una dimensione fissata (es: 15) e allungate il vettore di una lunghezza fissa (es: 10) ogni volta che questo si riempie.

## 3 Lettura di stringhe con allocazione di memoria

Scrivete due funzioni che leggano da standard input una sequenza di caratteri e la memorizzino in una stringa di dimensione opportuna allocata dinamicamente (scegliete la strategia che preferite, ad esempio una di quelle proposte nell'esercizio precedente):

1. `char *read_line( void )` deve leggere una riga terminata da `\n`;
2. `char *read_word( void )` deve leggere una parola di caratteri alfanumerici (nota: se il primo carattere letto non è alfanumerico, la stringa restituita sarà la stringa vuota).

Entrambe le funzioni devono restituire l'indirizzo del primo carattere della stringa memorizzata o il puntatore `NULL` in caso di errore.

## 4 Rettangoli

Aggiungete al programma dell'esercizio 6 della scheda sui puntatori una funzione che crei un nuovo rettangolo, ne assegni i membri con dati inseriti dall'utente, e ne restituisca l'indirizzo; la memoria necessaria va allocata dinamicamente usando la funzione `malloc`.