

CORSO DI FONDAMENTI DI INFORMATICA

Prof. Salvatore Andolina - A.A. 2023/2024

Laboratorio di Programmazione in C - Laboratorio 5

Esercizio 1 - Array dinamici

Creare un programma che, dopo aver chiesto la lunghezza di un array di interi, ne chieda l'inserimento e lo visualizzi. *N.B.: L'array della dimensione giusta deve essere allocato dinamicamente.*

Esempio:

Inserisci la lunghezza dell'array: 5

9 6 0 4 2

Output:

L'array da te inserito e': [9, 6, 0, 4, 2]

Esercizio 2 - Array dinamici 2.0

Modificare il programma dell'esercizio precedente affinché non chieda preventivamente la dimensione dell'array, ma che aumenti la dimensione dell'array dinamicamente con l'inserimento di ulteriori dati. L'inserimento termina quando viene inserito da tastiera il numero sentinella 42.

*Suggerimento: usate la funzione `void *realloc(void *ptr, size_t dim);`*

Esercizio 3 - Raddoppia lista

Realizzare le seguenti funzioni:

```
ListaNodi inserisci_in_coda(ListaNodi L, int elem);
```

```
void printaLista(ListaNodi L);
```

```
ListaNodi raddoppia(ListaNodi L);
```

per, rispettivamente: inserire un elemento in coda alla lista, visualizzare la lista, e raddoppiare ogni elemento di una data lista.

Esempio:

Lista originale: 5 -> 0 -> 4 -> 2 -> -1

Lista raddoppiata: 5 -> 5 -> 0 -> 0 -> 4 -> 4 -> 2 -> 2 -> -1 -> -1

Suggerimento:

definire il tipo per il nodo e per la lista nel modo seguente:

```
typedef struct ElemNodo{
    int dato;
    struct ElemNodo * next;
} Nodo;
typedef Nodo * ListaNodi;
```

Esercizio 4 - Libreria matematica

Creare una libreria matematica contenente le funzioni per il calcolo della potenza e del massimo comune divisore.

Scrivere un `main()` che, ricevuti due valori interi da riga di comando (usando `argc` e `argv[]`), richiami entrambe le funzioni e stampi il risultato.

Esercizio 5 - Leggi file di testo

Scrivere un programma che, usando `argc` e `argv[]`, riceva da riga di comando il **nome di un file di testo** e un numero compreso tra **1** e **2**, e visualizzi il file su terminale:

- *carattere per carattere* se il secondo argomento è **"1"**

- *riga per riga* se il numero inserito è **"2"** (assumere che la lunghezza della riga non superi i 256 caratteri).

Gestire correttamente con un messaggio il caso in cui non si riesca ad aprire il file, e il caso in cui il numero di argomenti inserito non e' corretto.

Esempio:

testo.txt:

```
ciao sono un  
file di testo 123!
```

esempio input da terminale: `./textReader.exe testo.txt 1`

output:

```
ciao sono un  
file di testo 123!
```

Come compilare i sorgenti da terminale con librerie:

il sorgente main dovra' includere al suo interno i header con le funzioni delle librerie:

programma.c:

```
#include <stdio.h>  
#include "funzioni_aggiuntive.h"  
  
int main() {  
    ...  
}
```

funzioni_aggiuntive.c:

```
#include "funzioni_aggiuntive.h"  
void foo(int a) {  
    ...  
}  
int bar(char * b) {  
    ...  
}
```

funzioni_aggiuntive.h:

```
void foo(int a);  
int bar(char * b);
```

Compilare da terminale:

```
> gcc programma.c funzioni_aggiuntive.c -o programma.exe
```

(N.B. se usate Linux o MacOS, invece di `programma.exe`, dovreste omettere l'estensione. uguale per quando eseguite il programma!)