

Compito di Programmazione I - Bioinformatica

11 luglio 2024 (tempo disponibile: 2 ore)

Esercizio 1 (12 punti) (si consegna `prefix.c` e `prefix.h`)

Si completi `prefix.c` in modo che la funzione `prefix(s1,s2)` determini se la stringa `s1` è un prefisso della stringa `s2` (cioè se `s2` comincia con `s1`). Tale funzione deve essere ricorsiva. Si scriva quindi il file di header `prefix.h` per l'esportazione di tale funzione.

Se tutto è corretto, compilando insieme a `main_prefix.c` (già scritto, da non modificare) ed eseguendo il risultato, verrà stampato:

```
prefix("anno","annata"): falso
prefix("anno","annoso"): vero
prefix("annata","annoso"): falso
prefix("annata","anno"): falso
prefix("annoso","annoso"): vero
prefix("", "annata"): vero
prefix("annata",""): falso
```

Esercizio 2 (19 punti) (si consegna `panopto.c`)

Un corso è caratterizzato da un codice univoco alfanumerico di 5 caratteri, il numero di CFU, la data di inizio corso. Il primo carattere del codice corso rappresenta il settore del corso, quindi ad esempio per il corso "BIO03", la B indica che fa parte del gruppo dei corsi di biologia.

Si costruisca una catalogo di corsi, rappresentato come lista concatenata, completando opportunamente il seguente programma.

La funzione `main` è già scritta e completa, non va modificata. Si possono aggiungere funzioni ausiliarie dentro `panopto.c`.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct data {
    int giorno;
    int mese;
    int anno;
};

struct corso {
    char codice_corso[6];
    int CFU;
    struct data data_inizio;
```

```

    struct corso *next;
};

struct corso *construct_entry(char codice_corso[], int CFU, struct data d,
    struct corso *next) {
//da completare
}

int list_length(struct corso *l) {
//da completare
}

void print_list(struct corso *l) {
//da completare
}

void destroy_list(struct corso *l) {
//da completare
}

int valid_study_plan (struct corso *l, int anno){
//da completare
}

struct corso * create_by_sector(struct corso *l, char s){
//da completare
}

struct corso * add_from_input (struct corso *l) {
//da completare
}

int main(void) {
    struct corso *l = NULL;
    l = construct_entry( "BI006", 10, (struct data){2,3,2024}, l);
    l = construct_entry( "INF04", 10,(struct data) {2,3,2024}, l);
    l = construct_entry("INF03",8,(struct data){11,11,2023}, l);
    l = construct_entry( "BI005", 12,(struct data) {1,3,2024}, l);
    l = construct_entry( "INF02", 12,(struct data) {12,4,2023}, l);
    l = construct_entry("INF01",8,(struct data){11,6,2023}, l);
    l = construct_entry( "BI004", 12,(struct data) {21,5,2024}, l);
    l = construct_entry( "BI003", 12,(struct data) {21,5,2024}, l);
    l = construct_entry( "BI002", 12,(struct data) {21,5,2024}, l);
    l = construct_entry( "BI001", 12,(struct data) {21,5,2024}, l);
    l = add_from_input(l);

    printf("La lista di corsi e' lunga %d\n", list_length(l));
    printf("I corsi sono:\n");
    print_list(l);
    printf("Piano di studio valido: %d\n", valid_study_plan (l, 2023));
    printf("Piano di studio valido: %d\n", valid_study_plan (l, 2024));
}

```

```

    struct corso * sector_plan=create_by_sector(1, 'B');
    print_list(sector_plan);
    destroy_list(sector_plan);
    destroy_list(1);
    return 0;
}

```

In particolare si completino le funzioni:

- `construct_entry` che crea (in testa) e inizializza un nodo della lista di corsi;
- `add_from_input` che riceve in input una lista e aggiunge 5 corsi da input, verificando la correttezza dei valori inseriti. I CFU devono essere al più 12, mentre l'anno di inizio corso deve essere compreso tra il 1990 ed il 2025. Un esempio di dati di test è contenuto nel file `corsi.txt` allegato.
- `list_length` che calcola la lunghezza della lista.
- `print_list` che stampa la lista.
- `destroy_list` che cancella la lista.
- `valid_study_plan` che riceve in input una lista e un anno, e verifica se la somma dei CFU dei corsi iniziati nell'anno dato e' almeno 60.
- `create_by_sector` che riceve in input una lista e un settore, e crea una lista contenente solo i corsi di quel settore. `create_by_sector` NON deve essere implementata in modo ricorsivo.

Se tutto è corretto, un esempio di esecuzione di `panopto.c` con la lista come inizializzata nel main e con i corsi inclusi in `corsi.txt` è:

```

La lista di corsi e' lunga 15
I corsi sono:
anno 2022: MAT09: 11 CFU
anno 2022: MAT10: 10 CFU
anno 2022: MAT11: 11 CFU
anno 2022: FIS01: 6 CFU
anno 2022: MAT12: 12 CFU
anno 2024: BI001: 12 CFU
anno 2024: BI002: 12 CFU
anno 2024: BI003: 12 CFU
anno 2024: BI004: 12 CFU
anno 2023: INF01: 8 CFU
anno 2023: INF02: 12 CFU
anno 2024: BI005: 12 CFU
anno 2023: INF03: 8 CFU
anno 2024: INF04: 10 CFU
anno 2024: BI006: 10 CFU
totale cfu 28
Piano di studio valido: 0
totale cfu 80
Piano di studio valido: 1
anno 2024: BI001: 12 CFU

```

anno	2024:	BI002:	12	CFU
anno	2024:	BI003:	12	CFU
anno	2024:	BI004:	12	CFU
anno	2024:	BI005:	12	CFU
anno	2024:	BI006:	10	CFU