

Compito di Programmazione I - Bioinformatica

2 febbraio 2024 (tempo disponibile: 2 ore)

Esercizio 1 (31 punti) (si consegna cifre.c e cifre.h)

Si crei il file di header `cifre.h` (2 punti) che definisce una lista di stringhe:

```
struct list {  
    char *head;  
    struct list *tail;  
};
```

e che dichiara le sei funzioni riportate sotto, che vanno invece completate dentro `cifre.c`:

```
// aggiungete #include se servissero  
  
// inizializza l'array indicato, lungo length,  
// in modo che diventi una stringa lunga length-1,  
// i cui caratteri siano caratteri di cifre decimali,  
// casuali, tali che nell'array risultante non ci  
// siano mai due caratteri consecutivi uguali  
void init(char arr[], int length) { // completare, 8 punti  
}  
  
// le stringhe italiane delle dieci cifre decimali  
char *digits[] = {  
    "zero", "uno", "due", "tre", "quattro",  
    "cinque", "sei", "sette", "otto", "nove"  
};  
  
// si assuma che s sia una stringa di caratteri di cifre decimali;  
// questa funzione crea e restituisce una lista di stringhe, fatta  
// dalle stringhe italiane corrispondenti ai caratteri di s; per  
// esempio, se i primi due caratteri di s fossero '3' e '5', allora  
// la lista risultante deve avere come primi due elementi la stringa  
// "tre" e poi la stringa "cinque";  
// questa funzione mette nel risultato le stringhe dell'array digits  
// definito sopra, direttamente, senza farne copie;  
// QUESTA FUNZIONE DEVE ESSERE RICORSIVA  
struct list *create_list(char *s) { // completare, 9 punti  
}  
  
// dealloca una lista di stringhe che era stata precedentemente  
// creata da create_list; QUESTA FUNZIONE DEVE ESSERE RICORSIVA  
void free_list(struct list *l) { // completare, 5 punti  
}
```

```

// identica alla funzione create_list, ma fa delle copie delle stringhe
// dell'array digits; QUESTA FUNZIONE DEVE ESSERE RICORSIVA
struct list *create_list2(char *s) { // completare, 4 punti
}

// identica alla funzione free_list, ma si applica a liste
// precedentemente allocate dalla funzione create_list2;
// QUESTA FUNZIONE DEVE ESSERE RICORSIVA
void free_list2(struct list *l) { // completare, 3 punti
}

// stampa una lista di stringhe, con uno spazio fra gli elementi,
// andando a capo alla fine
void print_list(struct list *l) {
    while (l) {
        printf("%s ", l -> head);
        l = l -> tail;
    }
    printf("\n");
}

```

Per esempio, l'esecuzione di main.c:

```

#include <stdlib.h>
#include <time.h>
#include "cifre.h"

int main(void) {
    srand(time(NULL));
    char s[31];
    init(s, 30);
    struct list *l = create_list(s);
    print_list(l);
    free_list(l);
    l = create_list2(s);
    print_list(l);
    free_list2(l);
    return 0;
}

```

potrebbe stampare qualcosa del tipo (la stampa è qui accorciata per mancanza di spazio):

```

due quattro cinque quattro sette nove [...altre cifre...] quattro uno quattro zero
due quattro cinque quattro sette nove [...altre cifre...] quattro uno quattro zero

```

Il file main.c è già scritto e completo, non va modificato e non va consegnato. Se servisse, si possono aggiungere funzioni ausiliarie dentro cifre.c.