

Programmazione I (Tucci/Distasi)

PR1 MT/RD 13/02/2023

Modello: 1

Cognome: _____

Nome: _____

Matricola: _____

Email: _____

Regole del gioco: Compilare i dati personali prima d'incominciare. Una volta iniziata la prova, non è consentito lasciare l'aula. Usare questi stessi fogli (compreso il retro, dove necessario) per rispondere. *Buon lavoro!*

1. Scriviamo una funzione

```
char * iniz_maius(char *V[], int n)
```

che riceve come parametri un array di stringhe con il numero dei suoi elementi, e restituisce una stringa allocata dinamicamente formata dai caratteri iniziali delle stringhe di V[] che iniziano con lettera maiuscola. Lo spazio allocato per la stringa risultato dev'essere giusto, senza sprechi.

Per esempio, se V[] contiene le stringhe

```
"un", "Compito", "davvero", "facile", "Incredibilmente", "Agevole", "ed",  
"Ovvio",
```

la chiamata di funzione `iniz_maius(v, 8)` restituirà "CIA0".

2. Consideriamo una terna, ossia una struttura dati contenente tre campi: nome, cognome, matricola. I campi nome e cognome sono stringhe, matricola è un intero.

Realizziamo una funzione che, ricevendo come parametro un array di queste strutture, la taglia dell'array e una lettera, salvi in un file binario chiamato `sottoelenco.bin` tutte le terne in cui il cognome inizia con quella lettera, senza distinzione fra minuscole e maiuscole. Definiamo adeguatamente le opportune strutture necessarie.

Risposte PR1 MT/RD 13/02/2023 – 1

1. Scriviamo una funzione

```
char * iniz_maius(char *V[], int n)
```

che riceve come parametri un array di stringhe con il numero dei suoi elementi, e restituisce una stringa allocata dinamicamente formata dai caratteri iniziali delle stringhe di V[] che iniziano con lettera maiuscola. Lo spazio allocato per la stringa risultato dev'essere giusto, senza sprechi.

Per esempio, se V[] contiene le stringhe

```
"un", "Compito", "davvero", "facile", "Incredibilmente", "Agevole", "ed",  
"Ovvio",
```

la chiamata di funzione `iniz_maius(v, 8)` restituirà "CIA0".

Risposta Ecco una possibile soluzione. Questa soluzione non è stata controllata con compilazione e testing, cosicché lo studente volenteroso possa compiere le verifiche personalmente!

```
#include <stdio.h>  
#include <stdlib.h>  
#include <ctype.h>  
  
char *iniz_maius(char *V[], int n)  
{  
    int i, j, count;  
    char *result;  
  
    for (i = 0, count = 0; i < n; i++)  
    {  
        if (v[i] != NULL // anti-crash  
            && isupper(V[i][0])) // stringa i-esima inizia per maiuscola?  
        {  
            count++; // contiamole  
        }  
    }  
    result = malloc(count + 1); // count caratteri + 1 per terminatore  
    if (result == NULL)  
    {  
        return NULL;  
    }  
    // else  
    for (i = 0, j = 0; i < n; i++)  
    {  
        if (v[i] != NULL && isupper(V[i][0]))  
        {  
            result[j++] = V[i][0];  
        }  
    }  
    result[j] = '\0';  
    return result;  
}
```

2. Consideriamo una terna, ossia una struttura dati contenente tre campi: nome, cognome, matricola. I campi nome e cognome sono stringhe, matricola è un intero.

Realizziamo una funzione che, ricevendo come parametro un array di queste strutture, la taglia dell'array e una lettera, salvi in un file binario chiamato `sottoelenco.bin` tutte le terne in cui il cognome inizia con quella lettera, senza distinzione fra minuscole e maiuscole. Definiamo adeguatamente le opportune strutture necessarie.

Risposta Ecco una possibile soluzione. Questa soluzione non è stata controllata con compilazione e testing, cosicché lo studente volenteroso possa compiere le verifiche personalmente!

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

typedef struct
{
    char nome[20];
    char cognome[20];
    int matricola;
} Individuo;

// restituiamo il numero di elementi scritti nel file
int sottoelenco(Individuo *arr, int n, char iniziale)
{
    FILE * outf;
    int i, scritti;

    outf = fopen("sottoelenco.bin", "wb");
    if (outf == NULL)
    {
        fprintf(stderr, "Impossibile aprire sottoelenco.bin in scrittura\n");
        return 0;
    }
    scritti = 0;
    for (i = 0; i < n; i++)
    {
        if (toupper(arr[i].cognome[0]) == toupper(iniziale))
        {
            status = fwrite(&arr[i], sizeof(Individuo), 1, outf);
            if (status == 1)
            {
                scritti++;
            }
            else
            {
                fprintf(stderr, "Errore scrivendo in sottoelenco.bin\n");
                return scritti;
            }
        }
    }
    fclose(outf);
    return scritti;
}
```