

```
// CONTA I CARATTERI COMPRESI GLI SPAZI E UN CARATTERE IN PARTICOLARE SIA MINUSCOLO CHE MAIUSCOLO
#include <stdio.h>
#define NOME_FILE "prova.txt"

int main()
{
    FILE * file_in;
    char car;
    int nchar; /* numero di caratteri */
    int nr; // numero di r
    float freq; // frequenza di r

    file_in = fopen(NOME_FILE, "r");

    if(file_in == NULL)
    {
        printf( "Non trovo il file.\n" );
        exit(-1);
    }

    nchar = 0;
    nr = 0;

    fscanf(file_in , "%c", &car);
    while (!feof(file_in))
    {
        // 1. conta quanti caratteri ci sono nel file
        nchar++;
        // 2. conta quante volte compare la lettera 'r' maiuscola o minuscola) ;
        if (car == 'r' || car == 'R')
        {
            printf("%c \n", car);
            nr++;
        }
        fscanf(file_in , "%c", &car);
    }

    // 3. calcola come FLOAT la frequenza di occorrenza della lettera 'r'
    //   (maiuscola o minuscola)
    freq = ((float) nr) / nchar; // attenzione: e' necessario il cast!

    printf(" Il numero di caratteri e': %d\n", nchar);
    printf(" Il numero di volte che compare la lettera 'r' o 'R' e': %d\n", nr);
    printf(" La frequenza della lettera 'r' o 'R' e': %.3g\n", freq);

    fflush(file_in);
    fclose(file_in);

    fflush(stdin);
    getchar();
    exit(0);
}
```

```
// CONTA PAROLE CON ALMENO UNA R
```

```
#include <stdio.h>
#define NOMEFILE "testo3.txt"
```

```
int confronta( char *s );
```

```
int main()
{
    FILE *qFile;
    char string[100];
    int contaparole = 0;

    qFile = fopen(NOMEFILE, "r");

    if (qFile == NULL ) {
        printf("file non trovato\n");
        exit(-1);
    }

    while (fscanf(qFile, "%s", string ) == 1) {
        contaparole += confronta(string);
    }

    printf("%d\n", contaparole);

    fflush(qFile);
    fclose(qFile);

    return 0;
}
```

```
int confronta ( char *s )
{
    int i = 0, parole = 0;

    while ( s[i] != '\0') {
        if (s[i] == 'r') {
            ++parole;
            break;
        }
        ++i;
    }
}
```

```
    return parole;  
}
```

```
// visualizza la parola + lunga di un file
```

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

```
#define NOMEFILE "prova.txt"
```

```
int main()
```

```
{
```

```
    FILE *pFile;
```

```
    char string[500];
```

```
    char string2[500] = {0};
```

```
    int i = 0;
```

```
    char n;
```

```
    int max = 0;
```

```
    pFile = fopen(NOMEFILE, "r");
```

```
    if (pFile == NULL ) {
```

```
        printf("file non trovato\n");
```

```
        exit(-1);
```

```
    }
```

```
    while (fscanf(pFile, "%s", string ) == 1 ) {
```

```
        i = strlen(string);
```

```
        if ( i > max ) {
```

```
            max = i;
```

```
            strcpy(string2, string);
```

```
        }
```

```
    }
```

```
    printf("la parola piu' lunga e' %s\n", string2);
```

```
    fflush(pFile);
```

```
    fclose(pFile);
```

```
    return 0;
```

```
}
```

// occorrenze cifre di un numero

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

```
int main()
{
```

```
    int num = 757179997;
    int i = 0;
    int freq[10] = {0};
```

```
    while ( num > 0 ) {
        i = num % 10;
        ++freq[ i ];
        num /= 10;
    }
```

```
    for ( i = 0; i < 10; i++ ){
        if ( freq[i] != 0 )
            printf("freq[%d] = %d\n", i, freq[i]);
    }
```

```
    return 0;
```

```
}
```

// somma gli elementi di una colonna specifica di uan matrice

```
#include <stdio.h>
#define N 2
#define M 3

int main()
{
    int V[N][M] = {{ 1, 2, 3 }, { 4, 5, 6 }}; // 6 e 15
    int c = 2, r, somma = 0;

    for ( r = 0; r <= N-1; r++ )
        somma += V[r][c];

    printf("la somma della prima colonna della matrice e' %d\n", somma );

    return 0;
}
```

```
// visualizza quale numero di un vettore si ripete + volte
```

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

```
#define N 14
```

```
#define M 10
```

```
int main()
```

```
{
```

```
    int i, j, max = 0, indice = 0;
```

```
    int V[N] = { 3, 2, 4, 2, 9, 1, 2, 2, 7, 4, 1, 2 }; // contiene num da 0 a 9
```

```
    int freq[M] = {0};
```

```
    for ( i = 0; i < N; i++ )
```

```
        ++freq[ V[i] ];
```

```
    for ( i = 0; i < M; i++ )
```

```
        printf("V[%d] = %d\n", i, freq[i]); // per visualizzare tutte le occorrenze
```

```
        // così posso visualizzare se è giusto
```

```
    for ( j = 0; j < M; j++ ) {
```

```
        if (freq[j] > max) {
```

```
            max = freq[j];
```

```
            indice = j;
```

```
        }
```

```
    }
```

```
    printf("\nnumero + frequente: %d\n", indice);
```

```
    return 0;
```

```
}
```

```
// ricerca numeri pari
```

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

```
#define N 5
```

```
int main()
```

```
{
```

```
    int i, x = 0, pari = 0;
```

```
    int V[N] = { 3, 10, 4, 7, 9 };
```

```
    for ( i = 0; i < N; i++ ) {
```

```
        x = V[i];
```

```
        if(( x % 2) == 0 ) {
```

```
            pari++;
```

```
        }
```

```
    }
```

```
    printf("i numeri pari sono: %d\n", pari);
```

```
    return 0;
```

```
}
```



```
// trova numero mancante
```

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

```
#define N 14
```

```
int main()
```

```
{
```

```
    int string[] = {0, 1, 2, 3, 4, 6, 7, 3, 2, 7, 8, 9, 10};
```

```
    int V[N];
```

```
    int i;
```

```
    for( i = 0; i <= 10; i++ )
```

```
        V[i] = 0;
```

```
    for( i = 0; i < N; i++ )
```

```
        V[string[i]] = 1;
```

```
    printf("numeri mancanti:\n");
```

```
    for( i = 0; i <= N; i++ ){
```

```
        if( V[i] == 0 ){
```

```
            printf("%d \t", i );
```

```
        }
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```