

Fondamenti di Informatica 2013-2014



File

Paola Mussida
Area Servizi ICT

Richiede il nome del file e vi
scrive 10 interi in formato
testo formattato.

Interi tratti da una
distribuzione uniforme
da -50 a 50

Esercizio Fi_1 - File di testo

3

```
void main() {  
  
    char nome[20];  
    int i, numero;  
    FILE *fp;  
  
    printf("\nNome File : ");  
    scanf("%s", nome);  
  
    fp=fopen(nome, "w");  
  
    if ( fp==NULL )  
        printf("\nErrore nell'apertura del file ! ");  
    else {  
        for (i=0;i<10;i++) {  
            numero= rand() % 101 - 50;  
            fprintf(fp,"%d\n",numero);  
        }  
        fclose(fp);  
    }  
}
```

Richiede il nome del file e vi scrive 10 interi in formato binario.

Interi tratti da una distribuzione uniforme da -50 a 50

```
void main() {  
  
    char nome[20];  
    int i, numero;  
  
    FILE *fp;  
    printf("\nNome File : ");  
    scanf("%s", nome);  
  
    fp=fopen(nome, "wb");  
    if ( fp==NULL )  
        printf("\nErrore nell'apertura del file ! ");  
    else {  
        for (i=0;i<10;i++) {  
            numero= rand() % 101 -50;  
            fwrite(&numero,sizeof(int), 1, fp);  
        }  
        fclose(fp);  
    }  
}
```

Richiede il nome dei due file creati negli esercizi Fi_1 ed Fi_2, confronta in ordine le coppie di numeri e indica se i valori coincidono.

```
void main() {  
    ...  
    fp_txt=fopen(nome_txt, "r");           /* apertura file di testo */  
    fp_bin=fopen(nome_bin, "rb");          /* apertura del file binario */  
    ...  
        /* prima lettura numero intero dal file di testo */  
        fscanf(fp_txt, "%d", &numero_txt);  
        /* prima lettura numero intero dal file binario */  
        fread(&numero_bin, sizeof(int), 1 , fp_bin);  
  
        /* ciclo di confronto dei numeri letti dai due file */  
        /* termina al raggiungimento della fine di un file */  
        while (                               ) {  
            /* elaborazione numero letto */  
        }  
        /* chiusura file */  
        fclose(fp_bin);  
        fclose(fp_txt);  
}
```



```
while ( ! (feof(fp_txt) || feof(fp_bin) ) )
{
    /* elaborazione numero letto */
    /* confronto */
    if (numero_txt==numero_bin)
        printf("\nI numeri coincidono: %d", numero_txt);
    else
        printf("\nI numeri non coincidono: %d != %d",
                numero_txt, numero_bin);

    /* lettura numero intero dal file di testo */
    fscanf(fp_txt, "%d", &numero_txt);

    /* lettura numero intero dal file binario */
    fread(&numero_bin, sizeof(int), 1 , fp_bin);
}
```


Legge carattere per carattere,
un file di testo, simile a
quello di esempio.

File di testo:

This is an example line. Line number 1
This is an example line. Line number 2
This is an example line. Line number 3
This is an example line. Line number 4
This is an example line. Line number 5
This is an example line. Line number 6
This is an example line. Line number 7
This is an example line. Line number 8
This is an example line. Line number 9
This is an example line. Line number 10

Esercizio Fi_4 - Lettura caratteri

11

```
FILE *funny;
...
/* apertura del file */
funny = fopen("TENLINES.TXT", "r");

/* verifica errori in apertura */
if ( funny == NULL )
    printf("File doesn't exist\n");
else {
    /* ciclo di lettura dei singoli caratteri */
    do {
        /* legge un carattere (anche i newline) */
        c = fgetc(funny);
        /* visualizza il carattere */
        if (c != EOF)
            putchar(c);
    } while ( c != EOF );
    /* ripete la lettura fino alla fine del file */
}
/* chiusura del file */
fclose(funny);
```

Legge riga per riga, il file
dell'esercizio precedente.

Esercizio Fi_5 - Lettura righe

13

```
void main() {
FILE *fp;
char oneword[100];
char *c;
    /* apertura file di testo */
    fp = fopen("TENLINES.TXT","r");
    /* ciclo di lettura delle righe */
    do {
        /* legge una riga dal file (n-1 caratteri) o fino alla
           fine della riga (\n rimane incluso nella stringa) */
        c = fgets(oneword,100,fp);

        /* verifica errori durante la lettura */
        if ( c != NULL )
            /*visualizza la stringa intera*/
            printf("%s",oneword);

    } while (c != NULL); /* ripete fino alla fine del file */

    fclose(fp);          /* chiusura file */
}
```

Elimina gli spazi in un file di testo, ricompattandolo in un altro file. La lettura avviene per singolo carattere.

```
char name_in[20], name_out[20];
char carattere=0;
FILE      *fp_in, *fp_out;

/* apertura di entrambi di file */
fp_in= fopen(name_in,"r"); /* file origine */
fp_out= fopen(name_out,"w"); /* file destinazione */

/* lettura di un carattere */
while ( (carattere=fgetc(fp_in)) != EOF )

    /* riscrivo nel secondo file solo i car. diversi da spazio */
    if (carattere!=' ')
        fputc(carattere, fp_out);

    /* il terminatore di riga viene letto e ricopiato */

/* chiusura di entrambi i file */
fclose(fp_in);
fclose(fp_out);
```

Differenza di conversione di
\n tra file di testo e file binari.


```
void main() {  
  
    FILE *f;  
    char c;  
    int a;  
  
    f = fopen("c:\\prova.txt", "w");  
    if (f==NULL)  
        printf("errore in scrittura wt\n");  
  
    fputs("riga1\n", f);  
    fputs("riga2\n", f);  
    fclose(f);  
}
```

```
printf("Lettura dei singoli caratteri come file di testo\n");
f = fopen("c:\\prova.txt","rt");
if (f==NULL)
    printf("errore in lettura rt\n");
else{
    a=0;
    while ( (c=fgetc(f))!=EOF )
        printf("%d - %c\n",a++, c);}
fclose(f);

printf("Lettura dei singoli caratteri come file binari\n");
f = fopen("c:\\prova.txt","rb");

if (f==NULL)
    printf("errore in lettura rb\n");
else{
    a=0;
    while ( (c=fgetc(f))!=EOF )
        printf("%d - %c\n",a++, c);}
fclose(f);
}
```

Lettura dei singoli
caratteri come file di
testo

0 - r
1 - i
2 - g
3 - a
4 - 1
5 -

6 - r
7 - i
8 - g
9 - a
10 - 2
11 -

Lettura dei singoli
caratteri come file
binario

0 - r
1 - i
2 - g
3 - a
4 - 1
5 -
6 -

7 - r
8 - i
9 - g
10 - a
11 - 2
12 -
13 -

Legge da un file di input una serie di righe di lunghezza variabile e le riscrive in un nuovo file, centrando in una finestra di 80 caratteri.

yy

jfjfjfkakdjfal

sldkfa

askfjaslfjaslfasklòdjgas

safasf

asdfaskljfas

asdlgfkxa

asdlkjgasdjgaslkjg

asgaskljgasklònbmxcnanagaskl

poirqopwgpujwgp

yy
jfjfkakdjfal
sldkfa
aslkfjaslfjaslfasklòdjgas
safasf
asdfasklfas
asdlgfsa
asdlkjgasdjgaslkjg
asgaskljgasklònbmxcnanagaskl
poirqopwgpujwgp

```
FILE *fp_in, *fp_out;

...
if ((fp_in=fopen(name_in,"r")) == NULL) /* apertura file di origine */
...

if ((fp_out=fopen(name_out,"w"))==NULL)/*apertura file di destinazione*/
...
while (fgets(riga,MAX_LINE, fp_in) != NULL) { /*lettura riga originale*/

/* calcolo spazi da aggiungere in testa alla stringa */
    k=(MAX_LINE - (strlen(riga) -1) )/2;
    /* scrittura degli spazi necessari */
    for (i=0;i<k;i++)
        fputc(' ',fp_out);

    fputs(riga, fp_out);      /* copia della stringa originale */
}

/* chiusura di entrambi i file */
fclose(fp_in);
fclose(fp_out);
```