



# UNIVERSITÀ TELEMATICA INTERNAZIONALE UNINETTUNO

Facoltà di Ingegneria

Prova scritta di

**INFORMATICA**

Sede di \_\_\_\_\_

03/12/2019

COGNOME	
NOME	
MATRICOLA	
FIRMA LEGGIBILE	

*SOLUZIONE*

**Istruzioni:**

1. **PER LE DOMANDE DI TEORIA USARE ESCLUSIVAMENTE LO SPAZIO A DISPOSIZIONE SOTTO CIASCUNA DOMANDA. PER L'ESERCIZIO DI PROGRAMMAZIONE USARE UN FOGLIO PROTOCOLLO.**
2. Non è possibile consultare né libri né appunti.
3. Tempo a disposizione: 120 minuti
4. Si ricorda che il compito è sufficiente solo se entrambe le parti sono sufficienti (ossia si ottengono almeno 7 punti per la parte di Teoria e almeno 11 per quella di Programmazione).

#include <stdio.h>	#include <stdlib.h>	#include <ctype.h>	#include <limits.h>
<b>FILE *fopen(char *filename, char *mode)</b> – Apertura di un file (mode: “r” lettura – “w” scrittura – “a” append)	<b>int abs (int n)</b> – valore assoluto intero	<b>int isalnum(int c)</b> – Vero se c è alfanumerico	<b>INT_MAX</b> - Indica il più grande valore che è possibile rappresentare con un int
<b>FILE *freopen(char *filename, char *mode, FILE *file_pointer)</b> - Riassegna un file puntatore ad un file diverso	<b>long labs(long n)</b> – valore assoluto	<b>int isalpha(int c)</b> - Vero se c è una lettera dell'alfabeto (maiuscola o minuscola)	<b>INT_MIN</b> - Indica il più piccolo valore che è possibile rappresentare con un int
<b>int fdclose(FILE *file_pointer)</b> - Chiude un file	<b>double atof(char *string)</b> - Converte una stringa in un valore in floating point	<b>int iscntrl(int c)</b> - Vero se c è un carattere di controllo	<b>LONG_MAX</b> - Indica il più grande valore che è possibile rappresentare con un long
<b>int feof(FILE *file_pointer)</b> - Controlla se è stato incontrato un end-of-file in un file	<b>int atoi(char *string)</b> - Converte una stringa in un valore int	<b>int isdigit(int c)</b> - Vero se c è una cifra decimale	<b>LONG_MIN</b> - Indica il più piccolo valore che è possibile rappresentare con un long
<b>int fflush(FILE *file_pointer)</b> - Svuota il buffer di un file	<b>int atol(char *string)</b> - Converte una stringa in un valore long int	<b>int islower(int c)</b> - Vero se c è una lettera minuscola	
<b>int fgetc(char *buffer)</b> - Legge un carattere da "stdin" (tastiera)	<b>void exit(int val)</b> – Termina il programma, restituendo al S.O. il valore 'val'	<b>int isprint(int c)</b> - Vero se c è un carattere stampabile	<b>FLT_MAX, DBL_MAX</b> - Indica il più grande valore che è possibile rappresentare con un float (o double)
<b>int fgetc(FILE *file_pointer)</b> - Prende un carattere da un file	<b>EXIT_FAILURE</b> - costante per segnalare terminazione senza successo del programma con exit() o return; valore diverso da zero	<b>int isspace(int c)</b> - Vero se c è un carattere spazio, tab, ritorno a capo	<b>FLT_MIN, DBL_MIN</b> - Indica il più piccolo valore che è possibile rappresentare con un float (o double)
<b>char *gets(char *buffer)</b> - Legge una riga da "stdin" (tastiera)	<b>EXIT_SUCCESS</b> - segnala terminazione con successo del programma con exit() o return; vale 0	<b>int isupper(int c)</b> - Vero se c è una lettera maiuscola.	
<b>char *fgets(char *string, int maxchar, FILE *file_pointer)</b> - Legge una riga da un file	#include <string.h>	<b>int tolower(int c)</b> - Converte c in minuscolo, dà un valore int	
<b>int printf(char *format_string, ...)</b> - Scrive output formattato su "stdout" (schermo)	<b>char *strcpy(char *dest, char *src)</b> - Copia una stringa in un'altra. Restituisce dest	<b>int toupper(int c)</b> - Converte c in maiuscolo, dà un valore int	
<b>int fprintf(FILE *file_pointer, char *format_string, ...)</b> - Scrive output formattato in un file.	<b>char *strncpy(char *s1, char *s2, size_t n)</b> - Copia i primi "n" caratteri di s2 in s1. Restituisce s1	<i>Gli angoli sono in radianti (180°=π rad)</i>	
<b>int sprintf(char *string, char *format_string, ...)</b> - Scrive output formattato su una stringa	<b>int strcmp(char *s1, char *s2)</b> - Confronta s1 e s2 per determinare l'ordine alfabetico (<0, s1 prima di s2, 0 uguali, >0 s1 dopo s2)	<b>double fabs(double x)</b> – valore assoluto	
<b>int fputc(int c, FILE *file_pointer)</b> - Scrive un carattere in un file	<b>int strncmp(char *s1, char *s2, size_t n)</b> - Confronta i primi n caratteri di due stringhe.	<b>double acos(double x)</b> – arccoseno	
<b>int putchar(int c)</b> - Scrive un carattere su "stdout" (schermo)	<b>char *strncpy(char *s1, char *s2)</b> - Copia s2 in s1. Restituisce s1	<b>double asin(double x)</b> – arcseno	
<b>int puts(char *string)</b> - Scrive una stringa su "stdout" (schermo)	<b>int strlen(char *string)</b> - Determina la lunghezza di una stringa	<b>double atan(double x)</b> – arcotangente	
<b>int fputs(char *string, FILE *file_pointer)</b> - Scrive una stringa in un file.	<b>char *strcat(char *s1, char *s2, size_t n)</b> - Aggiunge s2 a s1. Restituisce s1	<b>double atan2(double y, double x)</b> – arcotangente di y/x	
<b>int scanf(char *format_string, args)</b> - Legge input formattato da "stdin" (tastiera)	<b>char *strchr(char *string, int c)</b> - Cerca la prima occorrenza del carattere c in string; restituisce un puntatore alla prima occorrenza di c in s, NULL se non presente	<b>double ceil(double x)</b> – intero superiore a x	
<b>int fscanf(FILE *file_pointer, char *format_string, args)</b> - Legge input formattato da file	<b>char *strrchr(char *string, int c)</b> - Cerca l'ultima occorrenza del carattere c in string	<b>double floor(double x)</b> – intero inferiore a x	
<b>int sscanf(char *buffer, char *format_string, args)</b> - Legge input formattato da una stringa	<b>char* strstr(char* s, char* t)</b> - Restituisce un puntatore alla prima occorrenza di t all'interno di s. Restituisce NULL se t non è presente in s	<b>double cos(double x)</b> – coseno	
<b>EOF</b> – end of file (costante a valore negativo)		<b>double sin(double x)</b> – seno	
<b>NULL</b> - puntatore nullo (valore 0)		<b>double tan(double x)</b> – tangente	
		<b>double cosh(double x)</b> – coseno iperbolico	
		<b>double sinh(double x)</b> – seno iperbolico	
		<b>double tanh(double x)</b> – tangente iperbolica	
		<b>double exp(double x)</b> – e <sup>x</sup>	
		<b>double log(double x)</b> – log <sub>e</sub> (x)	
		<b>double log10(double x)</b> – log <sub>10</sub> (x)	
		<b>double pow(double x, double y)</b> - x <sup>y</sup>	
		<b>double sqrt(double x)</b> – radice quadrata	

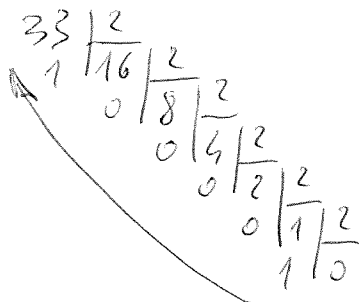
DOMANDA 1. (4 punti)

Si converta il numero -33.25 nel formato floating point IEEE P754 su 32 bit così definito: 1 bit di segno, 8 bit di esponente in eccesso 127, 23 bit di mantissa (in forma normalizzata 1.xxx... con bit nascosto).

-33.25

SEGNO: -  $\Rightarrow$  1

$$33_{10} = 100001_2$$



$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1,0$$

$$0,25_{10} = 0,01_2$$

$$33,25_{10} = 100001,01 = 1,0000101 \cdot 2^5$$

MANTESSA

ESPOLENTE:  $5 + 127 = 132 = 1000100_2$

S	ESPON	MANTESSA
1	10000100	00001010...0
1	8	23

DOMANDA 2. (4 punti)

Convertire il numero -12.346 (RIPORTANDO I CALCOLI EFFETTUATI) nei seguenti formati: MODULO E SEGNO con 1 bit di segno, 7 bit di parte intera e 8 bit di parte frazionaria:

>  
> 1 0001100 01011000

COMPLEMENTO A DUE con 1 bit di segno, 7 bit di parte intera e 8 bit di parte frazionaria:

>  
> 1 1110011 10101000

$$12_{10} = 1100_2$$

$$8 \text{ bit} \left\{ \begin{array}{l} 0,346 \times 2 = 0,692 \\ 0,692 \times 2 = 1,384 \\ 0,384 \times 2 = 0,768 \\ 0,768 \times 2 = 1,536 \\ 0,536 \times 2 = 1,072 \\ 0,072 \times 2 = 0,144 \\ 0,144 \times 2 = 0,288 \\ 0,288 \times 2 = 0,576 \end{array} \right.$$

$$0,346 = 0,01011000$$

$$12,346 = 1100,01011000$$

$$MS: \underbrace{1}_1 \underbrace{0001100}_7 \underbrace{01011000}_8 MS$$

$$CA2: \begin{array}{r} 0 \ 0001100 \ 01011000 \rightarrow +12,346 \\ 1 \ 1110011 \ 10101000 \rightarrow -12,346 \end{array}$$

DOMANDA 3. (4 punti)

Descrivere il meccanismo di funzionamento e gli usi dei sistemi di crittografia Simmetrici e Asimmetrici.

Vedere Videolezioni

#### DOMANDA 4. (18 punti)

Un file di testo di lunghezza ignota contiene in ogni singola riga, separati da spazi:

- il nome di un prodotto (stringa alfanumerica, priva di spazi)
- il costo del singolo prodotto (valore float)
- la quantità delle scorte a magazzino di quel prodotto (valore int)

Esempio di file:

Pinze 4.50 22

Tenaglie 7.99 12

Chiodi 0.02 234

Si scriva un programma in Linguaggio C89 che, letto questo file (il file non contiene errori, quindi non serve fare controlli sulla correttezza dei dati), ne crei un secondo modificato come descritto in seguito.

Il programma deve chiedere all'utente:

1. il nome del file da leggere
2. il nome del file da scrivere
3. il nome del prodotto di cui si vogliono modificare le scorte
4. un numero intero che indica come variare la quantità delle scorte di quel prodotto (se il numero è positivo, il valore corrispondente alle scorte di quel prodotto deve essere incrementato di quel numero, se negativo decrementato).

Dopo aver fatto la modifica indicata e scritto il nuovo file, il programma deve visualizzare il valore complessivo dei beni contenuti nel magazzino (ossia la somma di scorte\*costo) precedente e successivo alla modifica richiesta e quindi terminare l'esecuzione.

Il programma deve segnalare se il prodotto non è presente nel database mediante la scritta "Prodotto inesistente".

Per semplicità non si facciano controlli sulla consistenza dei valori (cioè sul fatto di avere quantità risultanti negative o nulle).

Esempio di esecuzione:

File in input: magaz.txt

File in output: magaz\_nuovo.txt

Prodotto: Tenaglie

Variazione: -5

Esempio di output:

Valore precedente: 199.56

Valore attuale: 159.61

Esempio di file prodotto:

Pinze 4.50 22

Tenaglie 7.99 7

Chiodi 0.02 234



# UNIVERSITÀ TELEMATICA INTERNAZIONALE UNINETTUNO

Facoltà di Ingegneria

Prova scritta di

**INFORMATICA**

Sede di \_\_\_\_\_

04/12/2019

COGNOME	
NOME	SOLUZIONE
MATRICOLA	
FIRMA LEGGIBILE	

## Istruzioni:

1. **PER LE DOMANDE DI TEORIA USARE ESCLUSIVAMENTE LO SPAZIO A DISPOSIZIONE SOTTO CIASCUNA DOMANDA. PER L'ESERCIZIO DI PROGRAMMAZIONE USARE UN FOGLIO PROTOCOLLO.**
2. Non è possibile consultare né libri né appunti.
3. Tempo a disposizione: 120 minuti
4. Si ricorda che il compito è sufficiente solo se entrambe le parti sono sufficienti (ossia si ottengono almeno 7 punti per la parte di Teoria e almeno 11 per quella di Programmazione).

```

#include <stdio.h>
FILE *fopen(char *filename, char *mode) -
Apertura di un file (mode: "r" lettura - "w"
scrittura - "a" append)
FILE *freopen(char *filename, char *mode, FILE
*file_pointer) - Riassegna un file puntatore ad
un file diverso
int fclose(FILE *file_pointer) - Chiude un file
int feof(FILE *file_pointer) - Controlla se è stato
incontrato un end-of-file in un file
int fflush(FILE *file_pointer) - Svuota il buffer di un
file
int fgetc(void) - Legge un carattere da
"stdin" (tastiera)
int fgetc(FILE *file_pointer) - Prende un
carattere da un file
char *gets(char *buffer) - Legge una riga da "stdin"
(tastiera)
char *fgets(char *string, int maxchar, FILE
file_pointer) - Legge una riga da un file
int printf(char *format_string, ...) - Scrive
output formattato su "stdout" (schermo)
int fprintf(FILE *file_pointer, char
format_string, ...) - Scrive output formattato in un
file.
int sprintf(char *string, char *format_string,
...) - Scrive output formattato su una stringa
int fputs(int c, FILE *file_pointer) - Scrive un
carattere in un file
int putchar(int c) - Scrive un carattere su
"stdout" (schermo)
int puts(char *string) - Scrive una stringa su
"stdout" (schermo)
int fputs(char *string, FILE *file_pointer) -
Scriva una stringa in un file.
int scanf(char *format_string, args) - Legge input
formattato da "stdin" (tastiera)
int fscanf(FILE *file_pointer, char *format_string,
args) - Legge input formattato da file
int scanf(char *buffer, char *format_string, args)
- Legge input formattato da una stringa
EOF - end of file (costante a valore negativo)
NULL - puntatore nullo (valore 0)

#include <stdlib.h>
int abs(int n) - valore assoluto intero
long labs(long n) - valore assoluto
double atof(char *string) - Converta una stringa in
un valore in floating point
int atoi(char *string) - Converta una stringa in un
valore int
int atoi(char *string) - Converta una stringa in un
valore long int
void exit(int val) - Termina il programma,
restituendo al S.O. il valore 'val'
EXIT_FAILURE - costante per segnalare
terminazione senza successo del programma
con exit() o return; valore diverso da zero
EXIT_SUCCESS - segnala terminazione con
successo del programma con exit() o return;
vale 0

#include <string.h>
char *strcpy(char *dest, char *src) - Copia una
stringa in un'altra. Restituisce dest
char *strncpy(char *s1, char *s2, size_t n) - Copia i
primi "n" caratteri di s2 in s1. Restituisce s1
int strcmp(char *s1, char *s2) - Confronta s1 e s2
per determinare l'ordine alfabetico (<0, s1
prima di s2, 0 uguali, >0 s1 dopo s2)
int strncmp(char *s1, char *s2, size_t n) -
Confronta i primi n caratteri di due stringhe.
char *strcpy(char *s1, char *s2) - Copia s2 in s1.
Restituisce s1
int strlen(char *string) - Determina la lunghezza di
una stringa
char *strcat(char *s1, char *s2, size_t n) -
Aggiunge s2 a s1. Restituisce s1
char *strncat(char *s1, char *s2, size_t n) -
Aggiunge "n" caratteri di s2 a s1. Restituisce s1
char *strchr(char *string, int c) - Cerca la prima
occorrenza del carattere c in string; restituisce
un puntatore alla prima occorrenza di c in s,
NULL se non presente
char *strchr(char *string, int c) - Cerca l'ultima
occorrenza del carattere c in string
char *strrchr(char *s, char *t) - Restituisce un
puntatore alla prima occorrenza di t all'interno
di s. Restituisce NULL se t non è presente in s

#include <ctype.h>
int isalnum(int c) - Vero se c è alfanumerico
int isalpha(int c) - Vero se c è una lettera
dell'alfabeto (maiuscola o minuscola)
int iscntrl(int c) - Vero se c è un carattere di
controllo
int isdigit(int c) - Vero se c è una cifra decimale
int islower(int c) - Vero se c è una lettera
minuscola
int isprint(int c) - Vero se c è un carattere
stampabile
int ispunct(int c) - Vero se c è un carattere di
punteggiatura
int isspace(int c) - Vero se c è un carattere spazio,
tab, ritorno a capo
int isupper(int c) - Vero se c è una lettera
maiuscola.
int tolower(int c) - Converta c in minuscolo, dà un
valore int
int toupper(int c) - Converta c in maiuscolo, dà un
valore int

#include <math.h>
Gli angoli sono in radianti (180°=π rad)
double fabs(double x) - valore assoluto
double acos(double x) - arcocoseno
double asin(double x) - arcseno
double atan(double x) - arcotangente
double atan2(double y, double x) - arcotangente
di y/x
double ceil(double x) - intero superiore a x
double floor(double x) - intero inferiore a x
double cos(double x) - coseno
double sin(double x) - seno
double tan(double x) - tangente
double cosh(double x) - coseno iperbolico
double sinh(double x) - seno iperbolico
double tanh(double x) - tangente iperbolica
double exp(double x) - ex
double log(double x) - loge(x)
double log10(double x) - log10(x)
double pow(double x, double y) - xy
double sqrt(double x) - radice quadrata

```



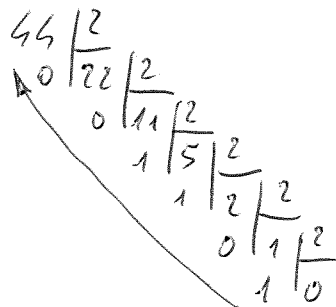
DOMANDA 1. (4 punti)

Si converta il numero 44.25 nel formato floating point IEEE P754 su 32 bit così definito: 1 bit di segno, 8 bit di esponente in eccesso 127, 23 bit di mantissa (in forma normalizzata 1.xxx... con bit nascosto).

+44.25

+: 0

44<sub>10</sub> = 101100<sub>2</sub>



$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1,0$$

$$0,25_{10} = 0.01$$

$$44,25 = 101100,01 = 1,0110001 \cdot 2^5$$

(MANTISSA)

ESPOLENTE:  $5 + 127 = 132 = 10000100_2$

S	ESPON	MANTISSA
0	10000100	01100010...0
1	8	23

DOMANDA 2. (4 punti)

Convertire il numero -12.346 (RIPORTANDO I CALCOLI EFFETTUATI) nei seguenti formati:  
MODULO E SEGNO con 1 bit di segno, 7 bit di parte intera e 8 bit di parte frazionaria:

> 1 0001100 01011000

COMPLEMENTO A DUE con 1 bit di segno, 7 bit di parte intera e 8 bit di parte frazionaria:

> 1 1110011 10101000

$$12 = 1100_2$$

8bit {

0,346	$\times 2 =$	0.692
0,692	$\times 2 =$	1.384
0,384	$\times 2 =$	0.768
0,768	$\times 2 =$	1.536
0,536	$\times 2 =$	1.072
0,072	$\times 2 =$	0.144
0,144	$\times 2 =$	0.288
0,288	$\times 2 =$	0.576

0.346 = 0.01011000

$$12.346 = 1100.01011000$$

MS: 1 0001100 01011000<sub>MS</sub>

1      7      8

CA2: 0 0001100 01011000<sub>CA2</sub> } (+12.346)  
1 1110011 10101000<sub>CA2</sub> } (-12.346)

DOMANDA 3. (4 punti)

Descrivere la codifica delle immagini in movimento.

Vedere Videolezioni

#### DOMANDA 4. (18 punti)

Un file di testo di lunghezza ignota contiene su ogni riga 24 numeri con la virgola, separati da uno spazio, corrispondenti alle temperature di una sorgente termale rilevate ad ogni ora.

Ogni riga corrisponde ad un giorno e i 24 valori sono preceduti dalla data nel formato gg/mm/aaaa. Il file non contiene errori (quindi non serve fare controlli sulla correttezza dei dati).

Si scriva un programma in Linguaggio C89 che legga questo file, il cui nome viene richiesto all'utente, e per ciascun giorno:

1. calcoli la media delle 24 temperature,

2. valuti se ci sono valori di temperatura che si discostano di più del 5% dalla media di quel giorno, nel qual caso scriva in un secondo file (di nome "Fuori\_media.txt") la data e la media giornaliera (i giorni in cui non ci sono variazioni di più del 5% non producono output).

Esempio di file in input:

06/12/2018 56.42 56.11 55.99 56.3 54.9 ecc.

07/12/2018 54.12 56.43 57.1 56.3 52.3 ecc.

08/12/2018 55.45 56.63 57.2 56.3 54.96 ecc.

09/12/2018 56.4 56.1 57.1 58.39 52.9 ecc.

10/12/2018 56.47 56.32 57.1 56.3 51.94 ecc.

ecc.

|

Esempio di file in output (le sole date in cui ci sono variazioni > 5%):

07/12/2018 55.96

09/12/2018 56.43

ecc...



# UNIVERSITÀ TELEMATICA INTERNAZIONALE UNINETTUNO

Facoltà di Ingegneria

Prova scritta di

**INFORMATICA**

Sede di \_\_\_\_\_

05/12/2019

COGNOME	SOLUZIONE
NOME	
MATRICOLA	
FIRMA LEGGIBILE	

**Istruzioni:**

1. **PER LE DOMANDE DI TEORIA USARE ESCLUSIVAMENTE LO SPAZIO A DISPOSIZIONE SOTTO CIASCUNA DOMANDA. PER L'ESERCIZIO DI PROGRAMMAZIONE USARE UN FOGLIO PROTOCOLLO.**
2. Non è possibile consultare né libri né appunti.
3. Tempo a disposizione: 120 minuti
4. Si ricorda che il compito è sufficiente solo se entrambe le parti sono sufficienti (ossia si ottengono almeno 7 punti per la parte di Teoria e almeno 11 per quella di Programmazione).

```
#include <stdio.h>
FILE *fopen(char *filename, char *mode) -
Apertura di un file (mode: "r" lettura - "w"
scrittura - "a" append)
FILE *freopen(char *filename, char *mode, FILE
*file_pointer) - Riassegna un file puntatore ad
un file diverso
int fclose(FILE *file_pointer) - Chiude un file
int feof(FILE *file_pointer) - Controlla se è stato
incontrato un end-of-file in un file
int fflush(FILE *file_pointer) - Svuota il buffer di un
file
int getchar(void) - Legge un carattere da
"stdin" (tastiera)
int fgetc(FILE *file_pointer) - Prende un
carattere da un file
char *gets(char *buffer) - Legge una riga da "stdin"
(tastiera)
char *fgets(char *string, int maxchar, FILE
file_pointer) - Legge una riga da un file
int printf(char *format_string, ...) - Scrive
output formattato su "stdout" (schermo)
int fprintf(FILE *file_pointer, char
format_string, ...) - Scrive output formattato in un
file.
int sprintf(char *string, char *format_string,
...) - Scrive output formattato su una stringa
int fputs(int c, FILE *file_pointer) - Scrive un
carattere in un file
int putchar(int c) - Scrive un carattere su
"stdout" (schermo)
int puts(char *string) - Scrive una stringa su
"stdout" (schermo)
int fputs(char *string, FILE *file_pointer) -
Scriva una stringa in un file.
int scanf(char *format_string, args) - Legge input
formattato da "stdin" (tastiera)
int fscanf(FILE *file_pointer, char *format_string,
args) - Legge input formattato da file
int sscanf(char *buffer, char *format_string, args)
- Legge input formattato da una stringa
EOF - end of file (costante a valore negativo)
NULL - puntatore nullo (valore 0)
```

```
#include <stdlib.h>
int abs (int n) - valore assoluto intero
long labs(long n) - valore assoluto
double atof(char *string) - Converta una stringa in
un valore in floating point
int atoi(char *string) - Converta una stringa in un
valore int
int atoi(char *string) - Converta una stringa in un
valore long int
void exit(int val) - Termina il programma,
restituendo al S.O. il valore 'val'
EXIT_FAILURE - costante per segnalare
terminazione senza successo del programma
con exit() o return; valore diverso da zero
EXIT_SUCCESS - segnala terminazione con
successo del programma con exit() o return;
vale 0
#include <string.h>
char *strcpy (char *dest, char *src) - Copia una
stringa in un'altra. Restituisce dest
char *strncpy(char *s1, char *s2, size_t n) - Copia i
primi "n" caratteri di s2 in s1. Restituisce s1
int strcmp(char *s1, char *s2) - Confronta s1 e s2
per determinare l'ordine alfabetico (<0, s1
prima di s2, 0 uguali, >0 s1 dopo s2)
int strncmp(char *s1, char *s2, size_t n) -
Confronta i primi n caratteri di due stringhe.
char *strcpy(char *s1, char *s2) - Copia s2 in s1.
Restituisce s1
int strlen(char *string) - Determina la lunghezza di
una stringa
char *strcat(char *s1, char *s2, size_t n) -
Aggiunge s2 a s1. Restituisce s1
char *strncat(char *s1, char *s2, size_t n) -
Aggiunge "n" caratteri di s2 a s1. Restituisce s1
char *strchr(char *string, int c) - Cerca la prima
occorrenza del carattere c in string; restituisce
un puntatore alla prima occorrenza di c in s,
NULL se non presente
char *strrchr(char *string, int c) - Cerca l'ultima
occorrenza del carattere c in string
char* strstr(char* s, char* t) - Restituisce un
puntatore alla prima occorrenza di t all'interno
di s. Restituisce NULL se t non è presente in s
```

```
#include <ctype.h>
int isalnum(int c) - Vero se c è alfanumerico
int isalpha(int c) - Vero se c è una lettera
dell'alfabeto (maiuscola o minuscola)
int iscntrl(int c) - Vero se c è un carattere di
controllo
int isdigit(int c) - Vero se c è una cifra decimale
int islower(int c) - Vero se c è una lettera
minuscola
int isprint(int c) - Vero se c è un carattere
stampabile
int ispunct (int c) - Vero se c è un carattere di
punteggiatura
int isspace(int c) - Vero se c è un carattere spazio,
tab, ritorno a capo
int isupper(int c) - Vero se c è una lettera
maiuscola.
int tolower(int c) - Converta c in minuscolo, dà un
valore int
int toupper(int c) - Converta c in maiuscolo, dà un
valore int
#include <math.h>
Gli angoli sono in radianti (180°= π rad)
double fabs (double x) - valore assoluto
double acos(double x) - arcocoseno
double asin(double x) - arcseno
double atan(double x) - arcotangente
double atan2(double y, double x) - arcotangente
di y/x
double ceil(double x) - intero superiore a x
double floor(double x) - intero inferiore a x
double cos(double x) - coseno
double sin(double x) - seno
double tan(double x) - tangente
double cosh(double x) - coseno iperbolico
double sinh(double x) - seno iperbolico
double tanh(double x) - tangente iperbolica
double exp(double x) - ex
double log(double x) - loge(x)
double log10 (double x) - log10(x)
double pow(double x, double y) - xy
double sqrt(double x) - radice quadrata
```

```
#include <limits.h>
INT_MAX - Indica il più grande valore che è
possibile rappresentare con un int
INT_MIN - Indica il più piccolo valore che è
possibile rappresentare con un int
LONG_MAX - Indica il più grande valore che è
possibile rappresentare con un long
LONG_MIN - Indica il più piccolo valore che è
possibile rappresentare con un long
#include <float.h>
FLT_MAX, DBL_MAX - Indica il più grande valore
che è possibile rappresentare con un float (o
double)
FLT_MIN, DBL_MIN - Indica il più piccolo valore
che è possibile rappresentare con un float (o
double)
```

DOMANDA 1. (4 punti)

Si converta il numero 36.625 nel formato floating point IEEE P754 su 32 bit così definito: 1 bit di segno, 8 bit di esponente in eccesso 127, 23 bit di mantissa (in forma normalizzata 1.xxx... con bit nascosto).

+36.625

+ : 0

$$36 = 100100_2$$

$$\begin{array}{r} 36 \mid 2 \\ 0 \mid 18 \mid 2 \\ 0 \mid 9 \mid 2 \\ 1 \mid 4 \mid 2 \\ 0 \mid 2 \mid 2 \\ 0 \mid 1 \mid 2 \\ 1 \mid 0 \end{array}$$

$$0,625 \times 2 = 1,25$$

$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1,0$$

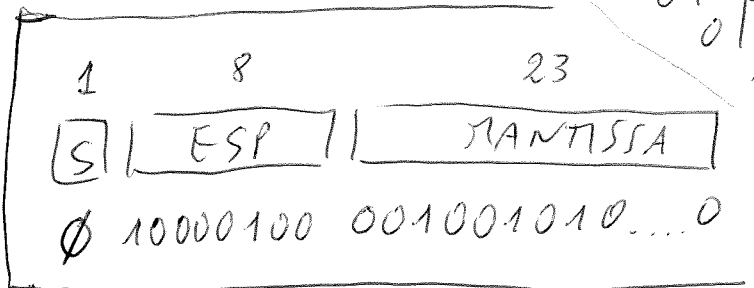
$$0,625 = 0,101_2$$

MANTISSA

$$36,625 = 100100,101 = 1,00100101 \cdot 2^5$$

ESPOLENTE :  $5 + 127 = 132 = 10000100$

$$\begin{array}{r} 132 \mid 2 \\ 0 \mid 66 \mid 2 \\ 0 \mid 33 \mid 2 \\ 1 \mid 16 \mid 2 \\ 0 \mid 8 \mid 2 \\ 0 \mid 4 \mid 2 \\ 0 \mid 2 \mid 2 \\ 0 \mid 1 \mid 2 \\ 1 \mid 0 \end{array}$$



DOMANDA 2. (4 punti)

Convertire il numero -12.346 (RIPORTANDO I CALCOLI EFFETTUATI) nei seguenti formati:  
MODULO E SEGNO con 1 bit di segno, 7 bit di parte intera e 8 bit di parte frazionaria:

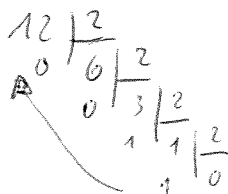
> 1 000 1100 01011000

COMPLEMENTO A DUE con 1 bit di segno, 7 bit di parte intera e 8 bit di parte frazionaria:

> 1 110011 10101000

-12.346

$12_{10} = 1100_2$



$$\begin{array}{l} 0,345 \times 2 = 0,692 \\ 0,692 \times 2 = 1,384 \\ 0,384 \times 2 = 0,768 \\ 0,768 \times 2 = 1,536 \\ 0,536 \times 2 = 1,072 \\ 0,072 \times 2 = 0,144 \\ 0,144 \times 2 = 0,288 \\ 0,288 \times 2 = 0,576 \end{array}$$

8 bit  
 $0,345 = 0.01011000$

$12.346 = 1100.01011000$

MS (1+7+8bit):  $\Rightarrow \underbrace{1}_{1} \underbrace{0001100}_{7} \underbrace{01011000}_{8}$

CA2 (1+7+8bit):  $\Rightarrow \begin{array}{ccc} 0 & 0001100 & 01011000 \\ 1 & 1110011 & 10101000 \end{array} \begin{array}{l} \rightarrow +12.346 \\ \downarrow \text{CA2} \rightarrow -12.346 \end{array}$



DOMANDA 3. (4 punti)

Descrivere la codifica del suono.

Vedere videolezioni

#### DOMANDA 4. (18 punti)

Un file di testo di lunghezza non nota a priori contiene l'elenco di oggetti venduti da un cartolaio in diversi giorni, ciascuna riga del file contiene, separati da spazi:

- il nome dell'oggetto venduto (non contiene spazi)
- quanti di quegli oggetti sono stati venduti
- il costo di ciascun oggetto
- la data nel formato aaaammgg (anno mese giorno)
- l'ora nel formato hhmm (ore minuti)

Esempio di file di input:

matita 2 0.90 20170305 0945

colla 1 2.50 20170305 0945

copertina 3 1.00 20170306 1155

quaderno 8 1.50 20170306 1155

matita 1 0.95 20170306 1145

...

|

Si scriva un programma in Linguaggio C89 che chieda all'utente il nome del file con le vendite (il file non contiene errori, quindi non serve fare controlli sulla correttezza dei dati) e calcoli e visualizzi per ciascun giorno (separati da una virgola):

- la data (nello stesso formato),
- il numero degli oggetti venduti quel giorno
- il ricavo complessivo di quel giorno

|

Esempio di output:

Data: 20170305, Oggetti: 3, Ricavo: 4.3

Data: 20170306, Oggetti: 12, Ricavo: 15.95



UNIVERSITÀ TELEMATICA  
INTERNAZIONALE UNINETTUNO

# UNIVERSITÀ TELEMATICA INTERNAZIONALE UNINETTUNO

Facoltà di Ingegneria

Prova scritta di

**INFORMATICA**

Sede di \_\_\_\_\_

06/12/2019

COGNOME	
NOME	SOLUZIONE
MATRICOLA	
FIRMA LEGGIBILE	

## Istruzioni:

1. **PER LE DOMANDE DI TEORIA USARE ESCLUSIVAMENTE LO SPAZIO A DISPOSIZIONE SOTTO CIASCUNA DOMANDA. PER L'ESERCIZIO DI PROGRAMMAZIONE USARE UN FOGLIO PROTOCOLLO.**
2. Non è possibile consultare né libri né appunti.
3. Tempo a disposizione: 120 minuti
4. Si ricorda che il compito è sufficiente solo se entrambe le parti sono sufficienti (ossia si ottengono almeno 7 punti per la parte di Teoria e almeno 11 per quella di Programmazione).

```

#include <stdio.h>
FILE *fopen(char *filename, char *mode) -
Apertura di un file (mode: "r" lettura - "w"
scrittura - "a" append)
FILE *freopen(char *filename, char *mode, FILE
*file_pointer) - Riassegna un file puntatore ad
un file diverso
int fclose(FILE *file_pointer) - Chiude un file
int feof(FILE *file_pointer) - Controlla se è stato
incontrato un end-of-file in un file
int fflush(FILE *file_pointer) - Svuota il buffer di un
file
int fgetc(void) - Legge un carattere da
"stdin" (tastiera)
int fgetc(FILE *file_pointer) - Prende un
carattere da un file
char *gets(char *buffer) - Legge una riga da "stdin"
(tastiera)
char *fgets(char *string, int maxchar, FILE
file_pointer) - Legge una riga da un file
int printf(char *format_string, ...) - Scrive
output formattato su "stdout" (schermo)
int fprintf(FILE *file_pointer, char
format_string, ...) - Scrive output formattato in un
file.
int sprintf(char *string, char *format_string,
...) - Scrive output formattato su una stringa
int fputs(int c, FILE *file_pointer) - Scrive un
carattere in un file
int putchar(int c) - Scrive un carattere su
"stdout" (schermo)
int puts(char *string) - Scrive una stringa su
"stdout" (schermo)
int fputs(char *string, FILE *file_pointer) -
Scriva una stringa in un file.
int scanf(char *format_string, args) - Legge input
formattato da "stdin" (tastiera)
int fscanf(FILE *file_pointer, char *format_string,
args) - Legge input formattato da file
int sscanf(char *buffer, char *format_string, args)
- Legge input formattato da una stringa
EOF - end of file (costante a valore negativo)
NULL - puntatore nullo (valore 0)

#include <stdlib.h>
int abs(int n) - valore assoluto intero
long labs(long n) - valore assoluto
double atof(char *string) - Converta una stringa in
un valore in floating point
int atoi(char *string) - Converta una stringa in un
valore int
int atol(char *string) - Converta una stringa in un
valore long int
void exit(int val) - Termina il programma,
restituendo al S.O. il valore 'val'
EXIT_FAILURE - costante per segnalare
terminazione senza successo del programma
con exit() o return; valore diverso da zero
EXIT_SUCCESS - segnala terminazione con
successo del programma con exit() o return;
vale 0

#include <string.h>
char *strcpy(char *dest, char *src) - Copia una
stringa in un'altra. Restituisce dest
char *strncpy(char *s1, char *s2, size_t n) - Copia i
primi "n" caratteri di s2 in s1. Restituisce s1
int strcmp(char *s1, char *s2) - Confronta s1 e s2
per determinare l'ordine alfabetico (<0, s1
prima di s2, 0 uguali, >0 s1 dopo s2)
int strncmp(char *s1, char *s2, size_t n) -
Confronta i primi n caratteri di due stringhe.
char *strncpy(char *s1, char *s2) - Copia s2 in s1.
Restituisce s1
int strlen(char *string) - Determina la lunghezza di
una stringa
char *strcat(char *s1, char *s2, size_t n) -
Aggiunge s2 a s1. Restituisce s1
char *strncat(char *s1, char *s2, size_t n) -
Aggiunge "n" caratteri di s2 a s1. Restituisce s1
char *strchr(char *string, int c) - Cerca la prima
occorrenza del carattere c in string; restituisce
un puntatore alla prima occorrenza di c in s,
NULL se non presente
char *strrchr(char *string, int c) - Cerca l'ultima
occorrenza del carattere c in string
char* strstr(char* s, char* t) - Restituisce un
puntatore alla prima occorrenza di t all'interno
di s. Restituisce NULL se t non è presente in s

#include <ctype.h>
int isalnum(int c) - Vero se c è alfanumerico
int isalpha(int c) - Vero se c è una lettera
dell'alfabeto (maiuscola o minuscola)
int iscntrl(int c) - Vero se c è un carattere di
controllo
int isdigit(int c) - Vero se c è una cifra decimale
int islower(int c) - Vero se c è una lettera
minuscola
int isprint(int c) - Vero se c è un carattere
stampabile
int ispunct(int c) - Vero se c è un carattere di
punteggiatura
int isspace(int c) - Vero se c è un carattere spazio,
tab, ritorno a capo
int isupper(int c) - Vero se c è una lettera
maiuscola.
int tolower(int c) - Converta c in minuscolo, dà un
valore int
int toupper(int c) - Converta c in maiuscolo, dà un
valore int

#include <math.h>
Gli angoli sono in radianti (180°= π rad)
double fabs(double x) - valore assoluto
double acos(double x) - arcocoseno
double asin(double x) - arcseno
double atan(double x) - arcotangente
double atan2(double y, double x) - arcotangente
di y/x
double ceil(double x) - intero superiore a x
double floor(double x) - intero inferiore a x
double cos(double x) - coseno
double sin(double x) - seno
double tan(double x) - tangente
double cosh(double x) - coseno iperbolico
double sinh(double x) - seno iperbolico
double tanh(double x) - tangente iperbolica
double exp(double x) - ex
double log(double x) - loge(x)
double log10(double x) - log10(x)
double pow(double x, double y) - xy
double sqrt(double x) - radice quadrata

```

DOMANDA 1. (4 punti)

Si converta il numero -13.625 nel formato floating point IEEE P754 su 32 bit così definito: 1 bit di segno, 8 bit di esponente in eccesso 127, 23 bit di mantissa (in forma normalizzata 1.xxx... con bit nascosto).

-13.625

SEGNO: -  $\Rightarrow$  1

13  $\frac{1}{2}$   
1  $\frac{6}{2}$   
0  $\frac{3}{2}$   
1  $\frac{1}{2}$   
1  $\frac{0}{2}$   
1100

$$0,625 \times 2 = 1,25$$

$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1,0$$

0.101

$$13.625 = 1100.101 = 1.\underbrace{100101}_{\text{MANTISSA}} \cdot 2^3$$

$$\text{ESPOLENTE: } 3 + 127 = 130 = 10000010_2$$

RICOMPONENDO:

S	ESPOLENTE	MANTISSA
1	10000010	1001010...0
1	8	23

DOMANDA 2. (4 punti)

Convertire il numero 10010110 (RIPORTANDO I CALCOLI EFFETTUATI) considerandolo di volta in volta espresso in:

BINARIO PURO:

> 150

MODULO E SEGNO:

> -22

COMPLEMENTO A DUE:

> -106

$$10010110 = 2^7 + 2^5 + 2^2 + 2^1 = 128 + 16 + 4 + 2 = 150$$

$$\text{MS: } 1 \overline{) 0010110_{10}} \\ \underline{- 22}$$

$$\text{CA2: } 1 \overline{) 0010110_{\text{CA2}}} \rightarrow -2^7 + 2^5 + 2^2 + 2^1 = -128 + 16 + 4 + 2 = \boxed{-106}$$

oppure

$$\begin{array}{r} 10010110 \\ 01101010 \end{array} \Rightarrow -(2^6 + 2^5 + 2^3 + 2^1) = - (64 + 32 + 8 + 2) = \boxed{-106}$$

DOMANDA 3. (4 punti)

Descrivere l'architettura generica di un elaboratore e le sue parti

Vedere Videotexto

#### DOMANDA 4. (18 punti)

Un file di testo di lunghezza non nota a priori contiene la misurazione del livello di un liquido in un serbatoio relativa a diversi giorni. Su ciascuna riga, separate da spazi, il file contiene le seguenti informazioni:

- la data nel formato aaaammgg (anno mese giorno)
- l'ora nel formato hhmm (ore minuti)
- il livello (in metri)
- la temperatura del liquido (in gradi)

Esempio di file di input

```
20070305 2244 2.13 21.4
20070305 2332 2.21 21.1
20070306 0255 2.33 20.92
20070306 0857 2.17 22.1
20070306 1122 2.1 24.2
20070307 1221 2.45 25.71
```

...

|

Si scriva un programma in Linguaggio C89 che chieda all'utente il nome del file (il file non contiene errori, quindi non serve fare controlli sulla correttezza dei dati) e visualizzi per ciascun giorno:

- il giorno,
- i valori minimo e massimo del livello,
- i valori minimo e massimo della temperatura.

|

Esempio di output

```
Data: 20070305 Lmin: 2.13 Lmax: 2.21 Tmin: 21.1 Tmax: 21.4
Data: 20070306 Lmin: 2.1 Lmax: 2.33 Tmin: 20.92 Tmax: 24.2
Data: 20070307 Lmin: 2.45 Lmax: 2.45 Tmin: 25.71 Tmax: 25.71
```





# UNIVERSITÀ TELEMATICA INTERNAZIONALE UNINETTUNO

Facoltà di Ingegneria

Prova scritta di

**INFORMATICA**

Sede di \_\_\_\_\_

09/12/2019

COGNOME	
NOME	SOLUZIONE
MATRICOLA	
FIRMA LEGGIBILE	

## Istruzioni:

1. **PER LE DOMANDE DI TEORIA USARE ESCLUSIVAMENTE LO SPAZIO A DISPOSIZIONE SOTTO CIASCUNA DOMANDA. PER L'ESERCIZIO DI PROGRAMMAZIONE USARE UN FOGLIO PROTOCOLLO.**
2. Non è possibile consultare né libri né appunti.
3. Tempo a disposizione: 120 minuti
4. Si ricorda che il compito è sufficiente solo se entrambe le parti sono sufficienti (ossia si ottengono almeno 7 punti per la parte di Teoria e almeno 11 per quella di Programmazione).

#include <stdio.h>	#include <stdlib.h>	#include <ctype.h>	#include <limits.h>
<b>FILE *fopen(char *filename, char *mode)</b> – Apertura di un file (mode: “r” lettura – “w” scrittura – “a” append)	<b>int abs (int n)</b> – valore assoluto intero	<b>int isalnum(int c)</b> - Vero se c è alfanumerico	<b>INT_MAX</b> - Indica il più grande valore che è possibile rappresentare con un int
<b>FILE *freopen(char *filename, char *mode, FILE *file_pointer)</b> - Riassegna un file puntatore ad un file diverso	<b>double atof(char *string)</b> - Converte una stringa in un valore in floating point	<b>int isalpha(int c)</b> - Vero se c è una lettera dell'alfabeto (maiuscola o minuscola)	<b>INT_MIN</b> - Indica il più piccolo valore che è possibile rappresentare con un int
<b>int fclose(FILE *file_pointer)</b> - Chiude un file	<b>int atoi(char *string)</b> - Converte una stringa in un valore int	<b>int iscntrl(int c)</b> - Vero se c è un carattere di controllo	<b>LONG_MAX</b> - Indica il più grande valore che è possibile rappresentare con un long
<b>int feof(FILE *file_pointer)</b> - Controlla se è stato incontrato un end-of-file in un file	<b>int atol(char *string)</b> - Converte una stringa in un valore long int	<b>int isdigit(int c)</b> - Vero se c è una cifra decimale	possibile rappresentare con un long
<b>int fflush(FILE *file_pointer)</b> - Svuota il buffer di un file	<b>void exit(int val)</b> – Termina il programma, restituendo al S.O. il valore ‘val’	<b>int islower(int c)</b> - Vero se c è una lettera minuscola	<b>LONG_MIN</b> - Indica il più piccolo valore che è possibile rappresentare con un long
<b>int getchar(void)</b> - Legge un carattere da "stdin" (tastiera)	<b>EXIT_FAILURE</b> - costante per segnalare terminazione senza successo del programma con exit() o return; valore diverso da zero	<b>int isprint(int c)</b> - Vero se c è un carattere stampabile	<b>FLT_MAX, DBL_MAX</b> - Indica il più grande valore che è possibile rappresentare con un float (o double)
<b>int fgetc(FILE *file_pointer)</b> - Prende un carattere da un file	<b>EXIT_SUCCESS</b> - segnala terminazione con successo del programma con exit() o return; vale 0	<b>int isspace(int c)</b> - Vero se c è un carattere spazio, tab, ritorno a capo	<b>FLT_MIN, DBL_MIN</b> - Indica il più piccolo valore che è possibile rappresentare con un float (o double)
<b>char *gets(char *buffer)</b> - Legge una riga da "stdin" (tastiera)		<b>int isupper(int c)</b> - Vero se c è una lettera maiuscola.	
<b>char *fgets(char *string, int maxchar, FILE *file_pointer)</b> - Legge una riga da un file	#include <string.h>	<b>int tolower(int c)</b> - Converte c in minuscolo, dà un valore int	
<b>int printf(char *format_string, ...)</b> - Scrive output formattato su "stdout" (schermo)	<b>char *strcpy(char *dest, char *src)</b> - Copia una stringa in un'altra. Restituisce dest	<b>int toupper(int c)</b> - Converte c in maiuscolo, dà un valore int	
<b>int fprintf(FILE *file_pointer, char *format_string, ...)</b> - Scrive output formattato in un file.	<b>char *strncpy(char *s1, char *s2, size_t n)</b> - Copia i primi "n" caratteri di s2 in s1. Restituisce s1	<i>Gli angoli sono in radianti (<math>180^\circ = \pi \text{ rad}</math>)</i>	
<b>int sprintf(char *string, char *format_string, ...)</b> - Scrive output formattato su una stringa	<b>int strcmp(char *s1, char *s2)</b> - Confronta s1 e s2 per determinare l'ordine alfabetico (<0, s1 prima di s2, 0 uguali, >0 s1 dopo s2)	<b>double fabs (double x)</b> – valore assoluto	
<b>int fputc(int c, FILE *file_pointer)</b> - Scrive un carattere in un file	<b>int strncmp(char *s1, char *s2, size_t n)</b> - Confronta i primi n caratteri di due stringhe.	<b>double acos(double x)</b> – arcocoseno	
<b>int putchar(int c)</b> - Scrive un carattere su "stdout" (schermo)	<b>char *strcpy(char *s1, char *s2)</b> - Copia s2 in s1. Restituisce s1	<b>double asin(double x)</b> – arcseno	
<b>int puts(char *string)</b> - Scrive una stringa su "stdout" (schermo)	<b>int strlen(char *string)</b> - Determina la lunghezza di una stringa	<b>double atan(double x)</b> – arcotangente	
<b>int fputs(char *string, FILE *file_pointer)</b> - Scrive una stringa in un file.	<b>char *strcat(char *s1, char *s2, size_t n)</b> - Aggiunge s2 a s1. Restituisce s1	<b>double atan2(double y, double x)</b> – arcotangente di y/x	
<b>int scanf(char *format_string, args)</b> - Legge input formattato da "stdin" (tastiera)	<b>char *strncat(char *s1, char *s2, size_t n)</b> - Aggiunge "n" caratteri di s2 a s1. Restituisce s1	<b>double ceil(double x)</b> – intero superiore a x	
<b>int fscanf(FILE *file_pointer, char *format_string, args)</b> - Legge input formattato da file	occorrenza del carattere c in string; restituisce un puntatore alla prima occorrenza di c in s, NULL se non presente	<b>double floor(double x)</b> – intero inferiore a x	
<b>int sscanf(char *buffer, char *format_string, args)</b> - Legge input formattato da una stringa	<b>char *strchr(char *s, char *t)</b> - Restituisce un puntatore alla prima occorrenza di t all'interno di s. Restituisce NULL se t non è presente in s	<b>double cos(double x)</b> – coseno	
<b>EOF</b> – end of file (costante a valore negativo)		<b>double sin(double x)</b> – seno	
<b>NULL</b> - puntatore nullo (valore 0)		<b>double tan(double x)</b> – tangente	
		<b>double cosh(double x)</b> – coseno iperbolico	
		<b>double sinh(double x)</b> – seno iperbolico	
		<b>double tanh(double x)</b> – tangente iperbolica	
		<b>double exp(double x)</b> – e <sup>x</sup>	
		<b>double log(double x)</b> – log <sub>e</sub> (x)	
		<b>double log10 (double x)</b> – log <sub>10</sub> (x)	
		<b>double pow(double x, double y)</b> - x <sup>y</sup>	
		<b>double sqrt(double x)</b> – radice quadrata	

DOMANDA 1. (4 punti)

Si converta il numero 18.125 nel formato floating point IEEE P754 su 32 bit così definito: 1 bit di segno, 8 bit di esponente in eccesso 127, 23 bit di mantissa (in forma normalizzata 1.xxx... con bit nascosto).

$$+18.125$$

SEGNO:  $+$   $\Rightarrow$  0

$$18_{10} = 10010_2$$

$$\begin{array}{r} 18 \div 2 = 9 \text{ residuo } 0 \\ 9 \div 2 = 4 \text{ residuo } 1 \\ 4 \div 2 = 2 \text{ residuo } 0 \\ 2 \div 2 = 1 \text{ residuo } 0 \\ 1 \div 2 = 0 \text{ residuo } 1 \end{array}$$

$$0.125 \times 2 = 0.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1$$

$$0.125_{10} = 0.001_2$$

$$18.125_{10} = 10010.001 = 1.0010001 \cdot 2^4$$

| MANTISSA |

ESPOLENTE:  $4 + 127 = 131 = 10000011_2$

S	ESPON	MANTISSA
0	10000011	00100010...0
1	8	23

DOMANDA 2. (4 punti)

Convertire il numero 10110110 (RIPORTANDO I CALCOLI EFFETTUATI) considerandolo di volta in volta espresso in:

BINARIO PURO:

> 182

MODULO E SEGNO:

> - 54

COMPLEMENTO A DUE:

> - 74

$$\text{B.P.: } \overset{7}{1}\overset{6}{0}\overset{5}{1}\overset{4}{1}\overset{3}{0}\overset{2}{1}\overset{1}{1}\overset{0}{0} = 2^7 + 2^5 + 2^4 + 2^2 + 2^1 = 128 + 32 + 16 + 4 + 2 = 182$$

$$\text{M.S.: } 10110110 = -(32 + 16 + 4 + 2) = -54$$

$$\text{CA2: } \overset{7}{1}\overset{6}{0}\overset{5}{1}\overset{4}{1}\overset{3}{0}\overset{2}{1}\overset{1}{1}\overset{0}{0} \xrightarrow{\text{ALTO}} -2^7 + 2^5 + 2^4 + 2^2 + 2^1 = -128 + 32 + 16 + 4 + 2 = -74$$

ALTO  
MOD0

$$\text{CA2 } \overset{7}{0}\overset{6}{1}\overset{5}{0}\overset{4}{0}\overset{3}{1}\overset{2}{0}\overset{1}{1}\overset{0}{0} = -(2^6 + 2^3 + 2^1) = -(64 + 8 + 2) = -74$$

DOMANDA 3. (4 punti)  
Descrivere i metodi di memorizzazione su file di immagini grafiche.

Vedere Videolezioni

DOMANDA 4. (18 punti)

Un file di testo di lunghezza ignota contiene l'elenco degli esami superati da uno studente: in ciascuna riga sono elencati 4 elementi separati da spazi:

- il codice dell'insegnamento (stringa di 5 caratteri alfanumerici, senza spazi)
- l'anno in cui è inserito quell'insegnamento (1, 2, 3)
- il nome dell'insegnamento (senza spazi intermedi)
- il corrispondente numero di crediti (numero intero)

Si scriva un programma in Linguaggio C89 che chieda all'utente il nome del file di input (il file non contiene errori, quindi non serve fare controlli sulla correttezza dei dati) e visualizzi il numero totale di esami superati, il numero totale di crediti conseguiti, il numero di crediti conseguiti per ciascun anno.

|  
Esempio di file in input:

```
AA728 1 Calcolo 9
H5TD1 1 Informatica 9
7JG66 2 Algoritmi 9
AA443 3 Reti_di_calcolatori 9
TDT32 1 Fisica 9
E7HH1 1 Chimica 9
YY787 2 Basi_di_dati 6
RR55T 3 Probabilità 6
WER56 2 Electronica 9
OIU89 3 Sistemi_Operativi 9
```

|  
Esempio di file in output:

```
Esami superati: 10
Crediti conseguiti: 84
Crediti anno 1: 36
Crediti anno 2: 24
Crediti anno 3: 24
```