



Cognome	Nome	Matricola	Voto: ... /30
----------------	-------------	------------------	----------------------

INIZIARE LA SOLUZIONE DI OGNI ESERCIZIO SU UNA PAGINA NUOVA

RESTITUIRE COMPILATO ANCHE NEL CASO IN CUI CI SI RITIRA

Quesito:	1	2	3	4	5	6	Tot.
Max:	5	3	6	5	5	6	30
Punti:							

Istruzioni:

- per contribuire alla valutazione finale, è necessario prendere almeno 18/30;
- non è possibile consultare libri, appunti, la calcolatrice o qualsiasi dispositivo elettronico, né comunicare;
- si può scrivere con qualsiasi colore, anche a matita, ad eccezione del rosso.
- tempo a disposizione: 2h 00m

Stile del codice C:

- non è necessario inserire direttive `#include`;
- i commenti non sono necessari, ma potrebbero essere utili nel caso di errore;
- è consentito l'utilizzo di funzioni di libreria.

Quesito 1 (5 punti) Punteggio ottenuto: .../5
 Dati i due numeri $A = -95_{16MS}$ e $B = -95_{10MS}$ effettuare la conversione in base 2, notazione complemento a 2 (2C2), sul numero minimo di bit necessari a rappresentare entrambi gli operandi. Si effettuino quindi le operazioni $A+B$ e $A-B$ indicando esplicitamente se si verifica overflow o meno, e motivando la risposta. Mostrare i passaggi fatti.

Quesito 2 (3 punti) Punteggio ottenuto: .../3
 Effettuare la conversione nel sistema decimale del valore $42e70000_{IEEE754}$.

Quesito 3 (6 punti) Punteggio ottenuto: .../6
 Si scriva un sottoprogramma che ricevuto in ingresso un array x di valori interi ed un valore intero n e qualsiasi parametro ritenuto strettamente necessario, trasmette al chiamante l'array di valori che costituiscono la **media mobile di x di finestra n** . L'elemento i -esimo della media mobile viene calcolato come **media** degli n valori del vettore in ingresso che precedono e includono l'elemento i . Se l'elemento i è preceduto da meno di $n-1$ valori, la media si calcola su quelli.

Ad esempio, se il sottoprogramma riceve in ingresso l'array seguente ed il valore $n=3$

10 2 3 13 101

le medie verranno calcolate su

- 10
- 10 2
- 10 2 3
- 2 3 13
- 3 13 101

Quesito 4 (5 punti) Punteggio ottenuto: .../5
 Scrivere un programma in C che acquisisca una stringa `str` (priva di spazi e lunga al più 500 caratteri). La stringa `str` contiene una sequenza ordinata (ordine alfabetico) di caratteri non necessariamente tutti distinti (può cioè contenere delle ripetizioni). Per ogni carattere `c` contenuto nella stringa `str`, a partire dall'ultimo fino ad arrivare al primo, il programma visualizza (senza lasciare spazi) il carattere `c`, seguito dal numero di volte che questo compare in `str`.

Per esempio, data la stringa `aabchhhhppppzzzz` il programma visualizza `z5p3h5c1b1a2`.

Quesito 5 (5 punti)

Punteggio ottenuto: .../5

I numeri di Catalan sono una successione C_n di interi positivi. Il nome di questi numeri è stato scelto in onore del matematico belga Eugène Charles Catalan (1814-1884) che li aveva studiati intorno al 1838 per stabilire in quanti modi si può ricondurre il prodotto di x fattori ad una successione di prodotti a coppie. I numeri di Catalan si possono ottenere ricorsivamente, secondo la seguente relazione:

$$C_0 = 1$$

$$C_n = \sum_{i=0}^{n-1} C_i \times C_{n-i-1}$$

Si scriva una funzione ricorsiva `cat` che, ricevuto un numero `n` come parametro formale, ritorna il valore di C_n .

Si scriva poi il programma principale che acquisisca dall'utente un numero intero positivo `x` (e finché non è tale lo richiede) e quindi calcola e visualizza i primi `x+1` numeri di Catalan, da C_0 a C_x .

Quesito 6 (6 punti)

Punteggio ottenuto: .../6

Si consideri il tipo di dato riportato di seguito, per rappresentare i termini di polinomi. Scrivere i sottoprogrammi `sum` e `prod` che dati due polinomi in ingresso calcolino e restituiscano rispettivamente il polinomio somma e il polinomio prodotto. I polinomi ricevuti in ingresso hanno i termini ordinati per potenze decrescenti, e così devono essere anche i polinomi creati.

```
typedef struct _term {
    int c; /* coefficiente */
    int p; /* potenza */
    struct _term * next;
} t_term;
```

Si preveda di disporre dei sottoprogrammi `lunghezza`, `conta`, `instesta`, `inscoda`, `insord`, `esiste`, `del`, `svuotalista`, i cui prototipi (con anche i nomi dei parametri) e funzionalità sono riportati di seguito. I sottoprogrammi qua riportati si riferiscono - per semplicità - al caso di lista per la gestione di dati interi: si immagini di disporre del sottoprogramma equivalente per la gestione di tipi di dati anche diversi, in base alle esigenze.

```
/* restituisce il numero di elementi presenti nella lista h */
int * lunghezza(t_elem * h);
/* restituisce il numero di elementi presenti nella lista h con campo informazione val */
int * conta(t_elem * h, int val);
/* crea un nuovo elemento con campo informazione val e lo inserisce in testa alla lista h, restituendo la testa */
t_elem * instesta(t_elem * h, int val);
/* crea un nuovo elemento di campo informazione val e lo inserisce in coda alla lista h, restituendo la testa */
t_elem * inscoda(t_elem * h, int val);
/* crea un nuovo elemento di campo informazione val e lo inserisce in nella lista h in ordine rispetto al
 * campo intero, restituendo la testa */
t_elem * insord(t_elem * h, int val);
/* cerca nella lista h un termine con campo informazione val e se esiste restituisce il puntatore a tale termine
 * altrimenti restituisce NULL */
t_term * esiste(t_elem * h, int val);
/* elimina dalla lista h un termine con campo informazione val e restituisce la testa della lista */
t_term * del(t_elem * h, int val);
/* svuota la lista h */
void svuotalista(t_elem * h);
```