

Fondamenti di Informatica - A.A. 2020-2021

Scuola di Ingegneria Industriale e dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del 05/02/2021



POLITECNICO
MILANO 1863

Cognome

Nome

Matricola o Cod. Persona

Quesito:	1	2	3	4	Totale
Valutazione massima:	8	5	6	14	33
Valutazione in decimi (/10):					

Istruzioni:

- 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: 1h 20m

Stile del codice C:

- non è necessario inserire direttive `#include`;
- i commenti non sono necessari, ma potrebbero essere utili nel caso di errore;
- è possibile utilizzare sottoprogrammi di libreria.

Quesito 1 [8 pts]

Domande vero/falso sulla rappresentazione dell'informazione

(1 pts) Il **principale** vantaggio della notazione in complemento alla base è

- ☐ l'immediatezza delle operazioni aritmetiche
- ☐ un più ampio insieme di valori rappresentabili a parità di numero di bit utilizzati
- ☐ la non ridondanza della rappresentazione del valore 0
- ☐ un più semplice metodo di calcolo dell'overflow nell'aritmetica

(1 pts) Dati i valori $A = 10010101010_{2C2}$ e $B = 11110101010_{2C2}$

- ☐ $A > B$
- ☐ $A < B$

(1 pts) Dati i valori $A = 10010101010_{2C2}$ e $B = 10011111110_{2C2}$

- ☐ $|A| > |B|$
- ☐ $|A| < |B|$

(1 pts) Il valore $A = 10010101010_{2C2}$ è rappresentato utilizzando il numero minimo di bit strettamente necessari

- ☐ Vero
- ☐ Falso

(1 pts) Il valore $11010101010_{2C2} + 01000000001_{2C2} = 00010101011_{2C2}$ e si verifica overflow.

- ☐ Vero
- ☐ Falso

(1 pts) Rappresentazione dei numeri reali in base 2, notazione IEEE 754: si tratta di una notazione che adotta il modulo e segno per rappresentare il valore

- ☐ Vero
- ☐ Falso

(1 pts) Rappresentazione dei numeri reali in base 2, notazione IEEE 754: il valore del campo esponente è rappresentato in complemento a 2, per una efficace manipolazione degli esponenti nelle operazioni di prodotto

- ☐ Vero
- ☐ Falso

(1 pts) Rappresentazione dei numeri reali in base 2, notazione IEEE 754: nessun valore viene mai esattamente rappresentato, si commette sempre un errore di approssimazione: nella notazione in singola precisione l'errore commesso è più piccolo di quello in doppia precisione.

- ☐ Vero
☐ Falso

Quesito 2 [5 pts]

Scrivere un sottoprogramma `baricentro` che ricevuto in ingresso un array di valori interi e qualsiasi altro parametro ritenuto strettamente necessario restituisce l'indice dell'elemento che fa da *baricentro* agli elementi dell'array, -1 se non esiste. Definiamo *baricentro* di un array di h elementi, l'elemento in posizione i tale per cui:

$$v[0] + v[1] + \dots + v[i] = v[i + 1] + v[i + 2] + \dots + v[h - 1]$$

Se dovessero esserci più baricentri, il sottoprogramma restituisce quello più a sinistra.

Quesito 3 [6 pts]

Scrivere un sottoprogramma `distanzaLineare` che riceve in ingresso le seguenti informazioni:

- `start`: il primo valore (valore reale)
- `stop`: l'ultimo valore (valore reale)
- `numero`: il numero di valori da distribuire nell'intervallo $[start, stop]$ (valore intero)
- `escludiStop`: 1 se si vuole escludere il valore `stop`, ossia se i valori devono coprire l'intervallo $[start, stop)$, 0 altrimenti

Il sottoprogramma crea e restituisce un insieme di numeri uniformemente distribuiti nell'intervallo $[start, stop]$ (oppure $[start, stop)$).

Se il sottoprogramma riceve in ingresso i valori: 2.0, 3.0, 5.0 l'insieme dei dati conterrà i seguenti valori:

2.0 2.25 2.50 2.75 3.0

Se il sottoprogramma riceve in ingresso i valori: 2.0, 3.0, 5.1 l'insieme dei dati conterrà i seguenti valori:

2.0 2.2 2.4 2.6 2.8

Quesito 4 [14 pts]

Si vuole realizzare un sottoprogramma `visualizzaAlternative` di supporto al correttore ortografico, che quando rileva un vocabolo che non esiste, cerca nel file dizionario tutti i vocaboli che l'utente potrebbe aver voluto scrivere commettendo un solo errore di battitura all'inizio del vocabolo: per esempio ha digitato `dasse` (invece di `casse`) e il correttore vuole suggerire i vocaboli *compatibili* con quello inesistente, a meno del primo carattere.

- (4 pts) Scrivere un sottoprogramma che riceve in ingresso una stringa `cerca` che contiene un vocabolo inesistente ed il riferimento ad un file di testo ASCII (c'è un vocabolo per riga, ogni vocabolo è al più di 30 caratteri) già aperto e visualizza tutti i vocaboli trovati nel file che differiscono per il solo primo carattere. Il sottoprogramma restituisce al chiamante il numero di vocaboli visualizzati.
- (5 pts) Rivedere il sottoprogramma precedente per definire il sottoprogramma `calcolaAlternative` in modo tale che restituisca al chiamante una lista contenente i vocaboli trovati, invece di visualizzarli. Non è importante l'ordine con cui i vocaboli sono visualizzati, ed allo scopo si utilizzi il sottoprogramma `push`, sotto riportato.
- (1 pts) Si definisca il tipo `slista_t` utilizzato dal sottoprogramma `push` e `calcolaAlternative`.
- (1 pts) Il sottoprogramma `push` qua riportato è stato realizzato correttamente?

```
slista_t * push(slista_t * h, char s[])
{
    slista_t * tmp;

    if (tmp = malloc(sizeof(slista_t))) {
        strcpy(tmp->parola, s);
        tmp->parola = h;
        h = tmp;
    } else
        printf("errore nell'allocare memoria per %s\n", s);
    return h;
}
```

- (3 pts) Scrivere un programma che acquisisce da riga di comando una stringa corrispondente ad un vocabolo scritto in modo errato e si avvale del sottoprogramma `visualizzaAlternative` per visualizzare i vocaboli corretti esistenti nel file `dizionario.txt`, e visualizzare poi il numero di vocaboli trovati. Per esempio, se si esegue il programma con il vocabolo `dasse`, ciò che viene visualizzato è quanto segue:

```
./cercaAlternative dasse
basse
casse
gasse
lasse
masse
nasse
passe
tasse
8 alternative
```

Sapevo già programmare:

☐ No, non è vero ☐ in C ☐ in C++/C# ☐ in Python ☐ in Java ☐ in PHP/javascript ☐ in VB* ☐ in altro linguaggio