

Fondamenti di Informatica - A.A. 2019-2020

Scuola di Ingegneria Industriale e dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del 17/02/2020



POLITECNICO
MILANO 1863

Cognome

Nome

Matricola o Cod. Persona

| | | | | | | |
|------------------------------|---|---|---|---|---|--------|
| Quesito: | 1 | 2 | 3 | 4 | 5 | Totale |
| Valutazione massima: | 5 | 8 | 6 | 6 | 5 | 30 |
| 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: 2h

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 [5 punti]

Dati i due valori $X = 11111010110101_{2MS}$ e $Y = 0110111011101101_{2C2}$ effettuare la conversione in base 2, notazione complemento a 2 ($2C2$), di ognuno degli operandi sul numero **minimo** di bit necessari. Si effettuino quindi le operazioni $X+Y$ e $X-Y$ indicando esplicitamente se si verifica overflow o meno, e motivando la risposta. Mostrare i passaggi fatti.

Riportare qua la codifica di X_{2C2} , Y_{2C2} e i risultati finali delle operazioni, utilizzando solo le caselle necessarie (**allineati a destra**) ed indicando se si è verificato overflow (segnare la casella corrispondente).

X_{2C2} :

Y_{2C2} :

$(X + Y)_{2C2}$:

$(X - Y)_{2C2}$:

overflow: si ☐ no ☐

overflow: si ☐ no ☐

Quesito 2 [8 punti]

(6 punti) Scrivere un sottoprogramma `conta` che riceve in ingresso due stringhe, `info` ed `cars`. Il sottoprogramma `conta` e restituisce al chiamante il numero di volte che ogni carattere presente in `cars` compare nella stringa `info`. Si crei una opportuna struttura dati per restituire al chiamante il risultato dell'elaborazione. Nella stringa `cars` non ci sono caratteri ripetuti. Per esempio, se le stringhe sono le seguenti:

```
info: TTGCAACAGCCTTAACAGCTTTGAATGACATGGGAAAGGTAAGAAA  
cars: ACGT
```

il sottoprogramma restituisce una struttura dati che contiene i valori 18 7 11 10.

Non sovradimensionare la struttura dati, deve essere definita su misura per contenere il numero di contatori strettamente necessari.

(2 punti) Scrivere un programma che acquisisce da riga di comando due stringhe, chiama il sottoprogramma `conta` e visualizza il risultato dell'elaborazione. Sempre con riferimento al precedente esempio:

```
./mioprogramma TTGCAACAGCCTTAACAGCTTTGAATGACATGGGAAAGGTAAGAAA ACGT  
A 18  
C 7  
G 11  
T 10
```

Quesito 3 [6 punti]

Nell'ambito dell'elaborazione delle immagini, un filtro a media mobile è un'operazione che riceve in ingresso un array bidimensionale di valori e per ciascun valore in posizione i, j calcola la media dell'array bidimensionale di dimensione 3×3 intorno a i, j ; nel caso la posizione i, j si trovi sul bordo, tutti gli elementi non esistenti vengono considerati valore 0. Realizzare un sottoprogramma che riceve in ingresso un array bidimensionale per l'immagine di partenza `image`, uno per l'immagine filtrata `filtered` e qualsiasi altro parametro ritenuto strettamente necessario che applica all'immagine di partenza il filtro a media mobile e lo trasmette al chiamante. Nel contesto di utilizzo del sottoprogramma, sono presenti le seguenti direttive/istruzioni riportate di seguito.

```
#define NR ...
#define NC ...

...
int main(int argc, char * argv[])
{
    ...
    float img[NR][NC], filtered[NR][NC];
    int dimR, dimC;
    ...
}
```

| | | | | | |
|------|-----|------|------|------|-----|
| 0.7 | 1.5 | 5.4 | 7.7 | 2.2 | 1.1 |
| -0.5 | 1.0 | 4.0 | 7.0 | 8.3 | 2.4 |
| -0.6 | 4.0 | 5.0 | 2.8 | -8.7 | 1.2 |
| 2.0 | 7.0 | -1.8 | -2.0 | 5.4 | 0.0 |
| 9.0 | 5.6 | 9.9 | -5.4 | 5.6 | 3.2 |

(a) immagine, l'array su cui calcolare la media

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 0.3 | 1.3 | 3.0 | 3.8 | 3.2 | 1.6 |
| 0.7 | 2.3 | 4.3 | 3.7 | 2.7 | 0.7 |
| 1.4 | 2.2 | 3.0 | 2.2 | 1.8 | 1.0 |
| 3.0 | 4.5 | 2.8 | 1.2 | 0.2 | 0.7 |
| 2.6 | 2.6 | 2.3 | 0.1 | 0.4 | 1.6 |

(b) immagine filtrata, un elemento in evidenza

Quesito 4 [6 punti]

Scrivere un programma che chiede all'utente il nome di un file di testo ASCII (al più 40 caratteri inclusi percorso ed estensione). Il file contiene righe di testo di al più 200 caratteri. Il programma conta e visualizza le seguenti informazioni:

- il numero di caratteri presenti nella riga più lunga contenuta nel file,
- il numero di caratteri presenti nella riga più corta contenuta nel file,
- il numero di righe, e
- il numero di frasi presenti nel file (ogni frase termina con un `.`).

Il file contiene un testo *ben formato*. Una frase può occupare più righe. Non ci sono `...` nel testo, quindi ogni punto **delimita senz'altro** una frase. La lunghezza di una riga è data dal numero di caratteri **diversi dallo spazio** in essa presenti. Il programma visualizza i valori conteggiati. Per esempio, se il contenuto del file fosse:

```
Testo per collaudare il programma. Devo vedere
quanto e' lunga la frase, e
quante frasi ci sono.
Inoltre conto quante righe ci sono.
E visualizzo.
Finito.
```

il programma visualizzerebbe:

```
max: 40
min: 7
righe: 6
frasi: 5
```

Nel caso si verificano problemi di accesso al file, il programma visualizza solo la scritta `ERRORE`.

Si suggerisce di creare un programma principale che acquisisce riga per riga il testo del file, e un sottoprogramma che data una riga la analizza per poter valutare le informazioni richieste dall'analisi.

Quesito 5 [5 punti]

Scrivere un sottoprogramma che riceve in ingresso una lista ed un valore intero k e restituisce il riferimento all'elemento in posizione k a partire dalla fine della lista. L'ultimo elemento della lista è identificato da $k=1$, il penultimo da $k=2$, e così di seguito. Se non esiste l'elemento in posizione k , il sottoprogramma restituisce `NULL`. È necessario sviluppare qualsiasi sottoprogramma di cui si abbia bisogno. Il tipo di dato è quello tradizionale della lista a puntatori semplice, con il riferimento all'elemento successivo.

Sapevo già programmare (rispondere anche se lo si è fatto nel precedente appello):

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