

Fondamenti di Informatica - A.A. 2017-2018

Scuola di Ingegneria Industriale e dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del 04/09/2018



POLITECNICO
MILANO 1863

| Cognome | Nome | Matricola o Cod. Persona |
|---------|------|--------------------------|
|---------|------|--------------------------|

| Quesito: | 1 | 2 | 3 | 4 | 5 | 6 | Totale |
|------------------------------|---|---|---|---|---|---|--------|
| Peso / Punti Max: | 5 | 2 | 6 | 6 | 5 | 6 | 30 |
| Valutazione in decimi (/10): | | | | | | | |

INIZIARE LA SOLUZIONE DI OGNI
ESERCIZIO SU UNA PAGINA NUOVA

RESTITUIRE COMPILATO ANCHE
NEL CASO IN CUI CI SI RITIRA

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 45m

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 numeri $A = -31_{16}$ e $B = +31_{10}$ effettuare la conversione in base 2, notazione complemento a 2, sul numero minimo di bit necessari a rappresentare gli operandi. Si effettuino poi, in tale rappresentazione, le operazioni $A+B$ e $A-B$ indicando esplicitamente se si verifica overflow o meno, e motivando la risposta. Mostrare i passaggi fatti e riportare qua solo i risultati finali (allineati a destra).

| | | | |
|-------------------|----------------------|-------------------|----------------------|
| A_{2C2} : | <input type="text"/> | B_{2C2} : | <input type="text"/> |
| $(A + B)_{2C2}$: | <input type="text"/> | $(A - B)_{2C2}$: | <input type="text"/> |

Quesito 2 [2 punti]

Scrivere la definizione di tipo per *elementi di una lista* per la gestione di rilevazioni ambientali di temperatura (valore reale) e umidità (valore intero) in un dato istante (si utilizzi una stringa di 12 caratteri).

Quesito 3 [6 punti]

(5 punti) Scrivere un sottoprogramma `rot` che riceva in ingresso una stringa `voc`, un numero intero senz'altro positivo `n` ed un intero `dir` (che vale senz'altro solo 0 o 1), la modifica in modo tale che la stringa finale sia quella iniziale, fatta scorrere di `n` posizioni a sinistra se `dir` vale 0, a destra se `dir` vale 1. con gli ultimi `n` caratteri riportati in testa. Se per esempio la stringa iniziale è `Esempio` ed `n` è 1 e `dir` è 1, la stringa finale sarà `oEsempi`. Data la stringa iniziale `Fondamenti`, `n` pari a 4 e `dir` pari a 0, la stringa finale è `amentiFond`.

(1 punto) Scrivere il programma che acquisisce **da riga di comando** la stringa `s`, l'intero `n` e l'intero `dir` e dopo aver utilizzato il sottoprogramma `rot` visualizza la stringa risultante.

Quesito 4 [6 punti]

Scrivere un sottoprogramma che riceva in ingresso una matrice di valori interi e qualsiasi altro parametro ritenuto strettamente necessario, calcola e restituisce l'*indice* della riga in cui la massima differenza ottenuta tra due suoi elementi, sia massima. Si ipotizzi che il numero di colonne della matrice sia definito tramite una direttiva `define` con il simbolo `NCOL`.

Quesito 5 [5 punti]

Scrivere un programma che chiede all'utente il nome di tre file di testo ASCII (ognuno di al più 80 caratteri, comprensivi di percorso ed estensione): i primi due file sono i file sorgente, il terzo quello destinazione ed un intero. Il programma crea il terzo file concatenando il contenuto dei due file in ingresso: se l'intero fornito vale 0, il programma concatena il contenuto del secondo file al primo, se l'intero fornito vale 1, il programma concatena il contenuto del primo al secondo. Nel caso in cui l'intero valga un valore diverso da 0 e da 1, il programma non fa nulla.

Quesito 6 [6 punti]

Scrivere un sottoprogramma in C `listebuffe` che riceve in ingresso una lista per la gestione di singoli caratteri (dichiarare il tipo) e restituisce un intero che vale 1 se i caratteri della lista corrispondono ad una stringa *buffa*, 0 altrimenti. Per determinare se i caratteri di una lista corrispondono ad una stringa *buffa*, si proceda nel seguente modo. Si crei una lista contenente gli elementi della lista di partenza, in ordine opposto (se la lista iniziale ha i caratteri $a \rightarrow b \rightarrow c \rightarrow d \rightarrow \perp$ la seconda lista ha gli elementi $d \rightarrow c \rightarrow b \rightarrow a \rightarrow \perp$). Si calcolano le differenze in valore assoluto tra i caratteri di due elementi adiacenti (per la prima stringa, la differenza tra a e b , tra b e c ...). Se la sequenza delle differenze in valore assoluto è la stessa per entrambe le liste, la lista corrisponde ad un stringa *buffa*. Per esempio, la lista $a \rightarrow c \rightarrow x \rightarrow z \rightarrow \perp$ corrisponde ad una stringa *buffa*, la lista $i \rightarrow v \rightarrow v \rightarrow k \rightarrow x \rightarrow \perp$ no.

È possibile utilizzare (senza doverli sviluppare) i sottoprogrammi i cui prototipi sono riportati di seguito.

```
/*inserisce l'elemento specificato in coda*/
elem* append(elem*, char);
/*inserisce l'elemento specificato in testa*/
elem* push(elem*, char);
/*inserisce l'elemento specificato in ordine crescente*/
elem* insertup(elem*, char);
/*inserisce l'elemento specificato in ordine decrescente*/
elem* insertdown(elem*, char);
/*rimuove la prima occorrenza dell'elemento specificato*/
elem* delete(elem*, char);
/*visualizza la lista*/
void view(elem*);
/*distrugge la lista*/
elem* destroy(elem*);
/*restituisce 1 se l'elemento e' presente nella lista, altrimenti 0*/
int exists(elem*, char);
```