

Fondamenti di Informatica - A.A. 2017-2018

Scuola di Ingegneria Industriale e dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del 19/02/2018



POLITECNICO
MILANO 1863

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

Quesito:	1	2	3	4	5	6	Totale
Peso / Punti Max:	5	3	4	6	6	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;
- come prima istruzione del sottoprogramma del quarto quesito, mettere un commento con il proprio cognome (1 punto extra);
- 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 possibile utilizzare sottoprogrammi di libreria.

Quesito 1 [5 punti]

Dati i due numeri $A = +63_{10MS}$ e $B = 1111111_{2C2}$ 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. Riportare poi qua i valori richiesti.

A:

B:

A+B:

A-B:

Quesito 2 [3 punti]

Dato il numero $A = -20.3125_{10MS}$ convertirlo in base 2, notazione IEEE 754, singola precisione, riportando tutti i bit della codifica. Rappresentare inoltre il valore risultante espresso in base 16. Dire inoltre se la codifica è *esatta* motivando la risposta. Mostrare i passaggi e riportare qua la codifica finale.

$A_{IEEE754}$:

$A_{IEEE754H}$:

Quesito 3 [4 punti]

Scrivere un programma che acquisisce i dati di un array di 25 valori interi. Una volta terminata l'acquisizione, il programma calcola e visualizza il massimo del prodotto di tre valori presenti nell'array. L'array, una volta acquisito, non deve essere modificato.

Quesito 4 [6 punti]

(5 punti) Scrivere un sottoprogramma `rep` che riceve in ingresso una stringa `s` e un intero `n` (senz'altro non negativo). Il sottoprogramma restituisce una nuova stringa ottenuta concatenando `n` copie di `s`. Ad esempio, se il sottoprogramma riceve in ingresso "esempio" e 3 restituisce la nuova "esempioesempioesempio", se riceve "esempio" e 0 restituisce una stringa vuota.

Non è consentito l'uso dei sottoprogrammi di libreria quali `strcat()`, `strcpy()` e simili.

(1 punto) Scrivere il programma che acquisisce da riga di comando la stringa `s` e l'intero `n` - utilizzando il sottoprogramma `rep` - visualizza la nuova stringa.

Quesito 5 [6 punti]

Si consideri una lista per la gestione di valori interi e si definisca un tipo opportuno `list_t`. Si scrivano due sottoprogrammi `kcon` e `knocon` che ricevuta in ingresso una lista del tipo definito ed un intero `k`, si comportino come specificato di seguito:

- `kcon`: restituisce 1 se la lista contiene almeno **due** occorrenze **consecutive** di `k`, 0 altrimenti, e
- `knocon`: restituisce 1 se la lista contiene almeno **due** occorrenze **non consecutive** di `k`, 0 altrimenti.

Un punto extra per ogni programma realizzato in modo ricorsivo.

Quesito 6 [6 punti]

Definire un tipo di dato opportuno `clist_t` per realizzare una lista dinamica che gestisce caratteri (ogni elemento un carattere) e serve per gestire *sequenze* di caratteri. Si definisce *sottosequenza* una sequenza di caratteri compresa tra una parentesi tonda iniziale (e una finale). Le sottosequenze possono anche essere vuote (parentesi aperta e poi chiusa senza altri caratteri intermedi). Scrivere un sottoprogramma `riceve` in ingresso una lista che costituisce una *sequenza* e la restituisce dopo aver sostituito le sottosequenze con il carattere #. La sequenza dei caratteri in ingresso è *ben formata*, ossia: i) per ogni parentesi tonda che si apre, c'è una parentesi tonda che si chiude, ii) non ci sono intersezioni tra coppie di parentesi. Per esempio, `ab(acg)be()a(xx)f` è una sequenza ben formata, `ab(a(c)g)b` e `aba(c)g)b` non lo sono.

Se il sottoprogramma riceve in ingresso la sequenza `ab(acg)be()a(xx)f(a)`, restituisce la sequenza `ab(#)be(#)a(#)f(#)`.

Si considerino già disponibili e non da sviluppare i sottoprogrammi seguenti:

```
/* inserisce in testa alla lista */
clist_t * push(clist_t *, char);
/* inserisce in coda alla lista */
clist_t * append(clist_t *, char);
/* elimina dalla lista il primo elemento */
clist_t * pop(clist_t *);
/* elimina dalla lista tutti gli elementi con il valore indicato */
clist_t * delete(clist_t *, char);
/* restituisce il riferimento all'elemento nella lista che ha il valore indicato, se esiste */
clist_t * exists(clist_t *, char);
/* restituisce il numero di elementi nella lista */
int length(clist_t *);
```