Fondamenti di Informatica - A.A. 2022-2023

Scuola di Ingegneria Industriale e dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del **12/01/2023**

POLITECNICO
MILANO 1863

Cognome	Nome	Matricola o Cod. Persona

INIZIARE LA SOLUZIONE DI OGNI ESERCIZIO SU UNA PAGINA NUOVA

Quesito:	1	2	3	4	5	Totale
Valutazione massima (in /30):	4	7	6	6	7	30
Valutazione quesito in decimi (/10):						

RESTITUIRE COMPILATO ANCHE NEL CASO IN CUI CI SI RITIRA

Istruzioni:

- gli esercizi devono essere risolti utilizzando il C **ANSI 89**, in linea con quanto fatto durante il corso;
- 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 40m

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.

Sapevo già programmare:

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

Quesito 1 [4 pti]

Dati i due valori X = 111110001101101011_{2MS} e Y = $+81DA_{16MS}$ 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 e motivare la risposta relativa all'overflow.**

Riportare nello spazio sottostante la codifica di X_{2C2}, Y_{2C2} e i risultati finali delle operazioni (tutti i passaggi devono essere sui fogli di protocollo), utilizzando solo le caselle necessarie (**allineati a destra**) ed indicando se si è verificato overflow (segnare la casella corrispondente).

X_{2C2} [ ] Y_{2C2} [ ]

$X+Y_{2C2}$ [ ] ☐ OVF $X-Y_{2C2}$ [ ] ☐ OVF

Quesito 2 [7 pti]

Scrivere un sottoprogramma che riceve come parametri due stringhe dette `sequenza` e `template`, entrambe composte soltanto da lettere dell'alfabeto sia minuscole sia maiuscole; inoltre `template` può contenere anche zero o più occorrenze del carattere speciale '_'. Assumendo che il carattere speciale '_' possa esser considerato uguale a qualsiasi carattere dell'alfabeto, il sottoprogramma conta e restituisce al chiamante quante volte `template` compare in `sequenza`.

Esempio:
ingresso: `sequenza` = **`aba`**`bcdAaabc`**`ada`**`cd` `template=a_a`
uscita: 2

Quesito 3 [6 pti]

Scrivere un programma che chiede all'utente il nome di una serie di file di testo (ciascuno di al più 30 caratteri) presenti nel file system e per ciascuno di essi conta il numero di caratteri in essi presenti. Il programma termina quando l'utente fornisce il nome di un file non accessibile e visualizza il numero totale di caratteri di tutti i file cui si è fatto accesso ed il nome del file con più caratteri.

Quesito 4 [6 pti]

Dato un array bidimensionale di interi, un elemento è detto **picco** se è strettamente maggiore di tutti gli altri 8 vicini (che devono esistere tutti e 8!). Scrivere un sottoprogramma che ricevuto in ingresso un array bidimensionale di interi (definito nel chiamante con un numero di colonne specificato tramite una `#define NCOL ...`) e qualsiasi altro parametro ritenuto strettamente necessario, calcola e trasmette al chiamante gli **indici** del picco di valore massimo. Si noti che se esistono più picchi con stesso valore massimo viene trasmesso al chiamante le coordinate del primo elemento trovato ed inoltre se non esiste alcun picco vanno trasmessi i valori -1 -1. Nell'esempio mostrato sotto, il picco di valore massimo è il valore 5 riportato in blu.

```
1 2 1 4 5
2 5 1 1 6
2 1 2 5 4
1 3 3 1 3
4 1 1 4 1
1 5 3 2 2
```

Quesito 5 [7 pti]

Scrivere un sottoprogramma che riceve in ingresso una stringa (chiamata `str`) e un carattere (chiamato `sep`). La stringa `str` è senz'altro composta da più sequenze di cifre ('0', ..., '9') separate da una singola occorrenza del carattere `sep` (si assuma che la stringa sia ben formata). Il sottoprogramma crea e restituisce una lista di interi strettamente positivi (`str` non contiene il carattere `'-'`) in cui ciascun elemento della lista contiene un valore intero corrispondente ad una sequenza di cifre presenti nella stringa `str`. Non è consentito modificare la stringa di ingresso.

Esempio:

ingressi: `str: "132,34,9121,1,29"`

`sep: ','`

uscita: `132 -> 34 -> 9121 -> 1 -> 29`

Si considerino già disponibili (e quindi non da sviluppare) i sottoprogrammi seguenti, validi per qualsiasi tipo di lista che gestisca un campo intero:

```
/* inserisce in testa alla lista elemento con valore specificato */
listtype * push(listtype *, int);
/* inserisce in coda alla lista elemento con valore specificato */
listtype * append(listtype *, int);
/* inserisce ordinatamente in lista elemento con valore specificato, in senso crescente */
listtype * increasing(listtype *, int);
/* inserisce ordinatamente in lista elemento con valore specificato, in senso decrescente */
listtype * decreasing(listtype *, int);
/* elimina dalla lista il primo elemento */
listtype * pop(listtype *);
/* elimina dalla lista tutti gli elementi con il valore indicato */
listtype * delete(listtype *, int);
/* restituisce il riferimento all'elemento che ha il valore indicato, se esiste, NULL altrimenti */
listtype * find(listtype *, int);
/* restituisce il numero di elementi nella lista */
int length(listtype *);
/* elimina la lista */
listtype * emptylist(listtype *);
```