

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione FONDAMENTI DI INFORMATICA Appello 18 Febbraio 2019		COGNOME E NOME						
	RIGA	COLONNA	CODICE PERSONA						
<div style="text-align: right;">Spazio riservato ai docenti</div> <table border="1" style="margin-left: auto; margin-right: 0;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>									

- Il presente plico contiene 5 esercizi e **deve essere debitamente compilato con cognome e nome, codice persona.**
- Il tempo a disposizione è di 1 ora e 40 minuti.
- Non separate questi fogli. Scrivete la soluzione solo sui fogli distribuiti, utilizzando il retro delle pagine in caso di necessità. Cancellate le parti di brutta con un tratto di penna.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- È possibile scrivere a matita (e non occorre ricalcare al momento della consegna!).
- **È vietato utilizzare telefoni, pc o calcolatrici.** Chi tenti di farlo vedrà annullata la sua prova.
- **Non è ammessa la consultazione di libri e appunti.**
- **Qualsiasi tentativo di comunicare con altri studenti comporta l'espulsione dall'aula.**
- È possibile ritirarsi senza penalità.
- **Non è possibile lasciare l'aula conservando il tema della prova in corso.**

Esercizio 1 (6 punti)

Si definisce un numero intero positivo **bisottratto** se gode delle seguenti due proprietà:

- è formato da almeno 3 cifre;
 - a partire dalla terza cifra meno significativa, ogni cifra deve essere pari alla differenza in valore assoluto delle due cifre precedenti (precedenti nel senso di meno significative).
- A. Implementare la funzione `int genera(int c1, int c2, int n)` che restituisce il numero **bisottratto** di `n` cifre che ha `c1` come cifra meno significativa e `c2` come penultima cifra meno significativa. Se non esiste nessun numero **bisottratto** con le caratteristiche richieste (`n`, `c1` e `c2`) la funzione deve restituire -1.
- B. Implementare la funzione `void stampa(int n)` che stampa a video tutti i numeri **bisottratti** di `n` cifre (ogni numero su una riga diversa).

Esempi

- `genera(2, 5, 5)` restituirà il numero bisottratto che termina con 52, cioè 12352 (terz'ultima cifra $3=|5-2|$, quart'ultima $2=|3-5|$ e quint'ultima $1=|2-3|$).
- `genera(1, 0, 5)` restituirà -1 poiché non c'è un intero bisottratto di 5 cifre che termina con 01 (la terzultima cifra sarebbe $1=|0-1|$, la quart'ultima ancora $1=|1-0|$ ma la quint'ultima sarebbe $0=|1-1|$, quindi il numero risultante sarebbe 1101 che è un intero a 4 cifre).

Soluzione

Esercizio 2 (8 punti)

Date le seguenti definizioni di tipo,

```
struct nodo
{
    int el;
    struct nodo *next;
};

typedef struct nodo *lista;
```

scrivere la funzione C **ricorsiva** `int count_before (lista l, int el)`, che calcola quanti nodi ci sono nella lista `l` prima di raggiungere o un nodo contenente il valore `el` oppure la fine della lista (nel caso non ci sia nessun nodo contenente il valore `el`).

Nota: nel conteggio va escluso il nodo stesso contenente il valore `el`.

Soluzione

Esercizio 3 (8 punti)

Si scriva una funzione `void offusca(char in[], char out[])` che riceve in ingresso due stringhe contenenti i nomi di due file (compresi di path). La funzione deve aprire il file testuale il cui nome è specificato dalla stringa `in` e copiarne il contenuto in un file testuale il cui nome è specificato dalla stringa `out` (si ipotizzi che questo file di output non esista), sostituendo ogni occorrenza del carattere ``@`` con la sequenza di caratteri `"_AT_"` ed ogni occorrenza del carattere ``.`` con la sequenza di caratteri `"_DOT_"`.

Soluzione

Esercizio 4 (5 punti)

Scrivere un programma in linguaggio macchina che legge dal nastro di ingresso una sequenza di 10 interi e stampa sul nastro di uscita 1 se la sequenza è in ordine crescente, 0 nel caso non lo sia.

Soluzione

Esercizio 5 (3 punti)

Dati i due numeri $A = -62_{10}$ e $B = +5_{10}$ effettuare la conversione in base 2, notazione complemento a 2 (CPL2), sul numero minimo di bit necessari a rappresentare correttamente entrambi gli operandi. Si effettuino quindi le operazioni $A+B$ e $A-B$ indicando esplicitamente se si verifica overflow o meno, motivando la risposta. Mostrare i passaggi fatti.

Soluzione

