

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione FONDAMENTI DI INFORMATICA Appello 5 Luglio 2017		COGNOME E NOME						
	RIGA	COLONNA	MATRICOLA						
<div style="text-align: right;">Spazio riservato ai docenti</div> <table border="1" style="margin-left: auto;"> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>									

- Il presente plico contiene 4 esercizi e **deve essere debitamente compilato con cognome e nome, numero di matricola.**
- Il tempo a disposizione è di 1 ore e 30 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 o pc.** Chi tenti di farlo vedrà annullata la sua prova.
- 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 (10 punti)

Implementare una funzione, `stampa_distinti`, che riceve come parametri in ingresso una stringa `nomefile`, contenente il nome di un file **testuale**, ed un numero intero `n`. La funzione ha il compito di aprire il file specificato attraverso il parametro `nomefile` e stampare a video i primi `n` caratteri **distinti** (cioè diversi fra loro) separati da uno spazio.

Note. Il file può contenere qualsiasi carattere ASCII.

Esempio

Si supponga che il file `nomefile` abbia questo contenuto: Ad esempio, abbiamo questo contenuto. La chiamata `stampa_distinti(nomefile, 12)` stamperà a video i seguenti caratteri :

A	d		e	s	m	p	i	o	,	a	b
---	---	--	---	---	---	---	---	---	---	---	---

Soluzione

```
void stampa_distinti(char nomefile[], int n)
{
    FILE *f;
    int caratteri[256] = {0}, stampati = 0;
    char c;

    f = fopen(nomefile, "r");
    while ( (c= fgetc(f)) != EOF && stampati < n)
        if (caratteri[c] == 0)
        {
            caratteri[c] = 1;
            printf("%c ", c);
            stampati++;
        }
}
```

Esercizio 2 (10 punti)

Scrivere una funzione **ricorsiva** che abbia il seguente prototipo:

```
char *inverti (char str[])
```

la funzione `inverti` dovrà restituire il puntatore ad una stringa contenente gli stessi caratteri di `str` ma in ordine inverso.

Esempio

Se `str` contiene “prova” la stringa restituita conterrà “avorp”.

Suggerimento: invertire “prova” equivale ad invertire “rova” e aggiungere una “p” in coda.

Soluzione

```
char * inverti(char str[])
{
    int l = strlen(str);
    char * result = calloc(l + 1, sizeof(char));

    if (l > 0)
    {
        char * inv = inverti(str + 1);
        strcpy(result, inv);
        free(inv);
        result[l - 1] = str[0];
        result[l] = '\0'; //superflua per l'uso di calloc
    }
    return result;
}
```

Esercizio 3 (4 punti)

- A. Dati i due numeri $X = -66_{10}$ e $Y = +111_{10}$ effettuare la conversione in base 2, utilizzando la codifica complemento a 2, con il numero minimo di bit necessari a rappresentare correttamente entrambi gli operandi.
- B. Si effettuino poi, in tale rappresentazione, le operazioni $X+Y$ e $X-Y$ indicando esplicitamente se si verifica overflow o meno, e motivando la risposta. Mostrare i calcoli svolti per rispondere alle domande.

Soluzione

Sono necessari 8 bit in CPL2 per rappresentare correttamente X e Y, dal momento che con 8 bit si possono rappresentare in CPL2 i numeri fra -128 a +128.

$$X = -66_{10} = 10111110_{\text{CPL2}}$$

$$Y = +111_{10} = 01101111_{\text{CPL2}}$$

$$-Y = \overline{01101111} + 1 = 10010001_{\text{CPL2}}$$

$$X+Y = 00101101_{\text{CPL2}} \text{ (non c'è overflow perchè gli operandi hanno segno discorde)}$$

$$\begin{array}{r} 1111111 \\ 10111110 + \\ 01101111 \\ \hline (1) 00101101 \end{array}$$

$$X-Y = \text{OVERFLOW} \text{ (c'è overflow perché il risultato ha segno discorde con gli operandi)}$$

$$\begin{array}{r} 1 \ 11 \\ 10111110 + \\ 10010001 \\ \hline (1) 01001111 \end{array}$$

Esercizio 4 (8 punti)

Si scriva in linguaggio macchina un programma che legga dal nastro di ingresso tre numeri interi, a, b e c.

Se b è uguale a 0, il programma scriverà sul nastro di uscita a+c; se b=1 scriverà a-c; altrimenti non scriverà niente.

Soluzione

1. READ
2. STORE 100
3. READ
4. STORE 101
5. READ
6. STORE 102
7. LOAD 101
8. BNE 13
9. LOAD 100
10. ADD 102
11. WRITE
12. BR 18
13. SUB= 1
14. BNE 18
15. LOAD 100
16. SUB 102
17. WRITE
18. END