## Esercizio 29 (12 punti)

In enigmistica la cerniera è uno schema per cui, date due parole, si ottiene una terza parola scartando il medesimo gruppo di lettere (e nel medesimo ordine) dalla testa (prefisso) della prima parola e dalla coda (suffisso) della seconda parola, unendo infine i caratteri rimasti ordinatamente nella prima e nella seconda parola. Per es. "**abc**def" /gh**abc**" restituisce "defgh".

1) **[3punti]** Scrivere in C il sottoprogramma

   `...CercaIndice(...)`

   avente come argomenti un carattere e una stringa, che restituisca la posizione della prima occorrenza del carattere nella stringa scorrendola da destra verso sinistra. Se il carattere non è presente nella stringa data, il sottoprogramma restituisce il valore -1.

   Esempi:
   il sottoprogramma invocato con argomenti `'t'`, `"tatto"` restituisce `3`;
   il sottoprogramma invocato con argomenti `'b'`, `"tatto"` restituisce `-1`;

2) **[6 punti]** Scrivere in C il sottoprogramma

   `…cerniera(…)`

   che abbia come parametri due stringhe e che indichi se la cerniera tra le due parole in esse contenute è possibile.
   Per indicare che la cerniera è possibile, il sotto-programma deve restituire la lunghezza della comune sequenza di caratteri che risulta essere a prefisso della prima stringa e a suffisso della seconda; mentre deve restituire -1 nel caso in cui la cerniera non sia possibile.

   Esempi:
   `"osti"`, `"mao"` → restituisce `1` (la loro cerniera è : `"stima"`)
   `"mare"`, `"tema"` → restituisce `2` (la loro cerniera è : `"rete"` )
   `"drago"`, `"ladra"` → restituisce `3` (la loro cerniera è ': `"gola"` )
   `"manico"` , `"stamani"` → restituisce `4` (la loro cerniera è : `"costa"`)
   `"papa"`, `"papa"` → restituisce `2` (la loro cerniera è : `"papa"`)
   `"ciao"`, `"ciao"` → restituisce `4` (la loro cerniera è : parola vuota)
   `"ciao"`, `"per"` → restituisce `-1` (la loro cerniera NON è possibile)

   Suggerimento:
   - Se si assume (come negli esempi) di fare la cerniera considerando prefissi e suffissi SENZA ripetizioni di caratteri, l'idea di soluzione potrebbe far uso del sotto-programma richiesto in 1)
   - È possibile far uso delle funzioni incluse nella libreria `string.h` (per es. per ottenere la lunghezza di una stringa).

3) **[3punti]** Scrivere in C un programma principale che faccia uso dei sotto-programmi precedenti e che chieda all'utente di inserire due parole. Il programma deve riportare a video un messaggio che indichi se la cerniera delle due parole sia o non sia possibile, e in caso affermativo visualizzarla.

**Soluzione Domanda 1)**

```
int cercaIndice(char c, char str[]) {
   int idx = strlen(str)-1;
   while (idx >= 0 && c != str[idx]) {
       idx = idx - 1;
   }
   return idx;
} // end cercaIndice
```

**Soluzione Domanda 2)**

```
// Assumendo di fare la cerniera considerando
// prefissi e suffissi SENZA ripetizioni di caratteri.
int cerniera(char prima[], char seconda[]) {

   int idxSuffisso = cercaIndice(prima[0], seconda);
   if (idxSuffisso < 0) return -1;

   int lung = strlen(seconda) - idxSuffisso;
   int flag = 1;
   for (int i = 1; i < lung && flag == 1; i++)
      if (prima[i] != seconda[idxSuffisso+i])
        flag = 0;

   if (flag == 0) return -1;
   return lung;
} // end cerniera

// Soluzione generale (NON RICHIESTA)- senza far uso di ...cercaIndice(...)
int cerniera(char prima[], char seconda[]) {

   int idxSeconda = strlen(seconda)-1;
   if (idxSeconda < 0) return -1;

   int riconoscimento_iniziato_su_prima, idxPrima, lung = 0;

   riconoscimento_iniziato_su_prima = idxPrima = strlen(prima)-1;
   while ( idxPrima >= 0 && idxSeconda >= 0) {
     if (prima[idxPrima] == seconda[idxSeconda]) {
       if (lung == 0) // fase di inizio riconoscimento
          riconoscimento_iniziato_su_prima = idxPrima;
       lung++;
       idxPrima--;
       idxSeconda--;
     }
     else {
            idxSeconda = strlen(seconda)-1;
            idxPrima = riconoscimento_iniziato_su_prima - 1;
            riconoscimento_iniziato_su_prima = idxPrima;
            lung = 0;
     } // end 1mo-if-else
    } // end while
    if (lung == 0)
      return -1;
    else
      return lung;
} // end cerniera
```

**Soluzione Domanda 3)**

```
#include <stdio.h>
#include <string.h>
#define MAX_LUNG_PAROLA 50
```

```c
int cercaIndice(char c, char str[]);
int cerniera(char prima[], char seconda[]);

int main() {
   char primaParola[MAX_LUNG_PAROLA+1] = {'\0'};
   char secondaParola[MAX_LUNG_PAROLA+1] = {'\0'};

   printf("\n Programma di valutazione della cerniera tra due parole \n");
   printf("\n Inserisci la 1ma parola: "); scanf("%s", primaParola);
   printf("\n Inserisci la 2da parola: "); scanf("%s", secondaParola);

   int lung = cerniera(primaParola, secondaParola);

   if (lung < 0) {
     printf("\n La cerniera NON e' possibile! \n");
     return 1;
   }
   if (lung == strlen(primaParola) && lung == strlen(secondaParola)) {
     printf("\n La cerniera e' la stringa vuota! \n");
   }
   else {
         printf("\n La cerniera e': ");
         for (int i = lung; i < strlen(primaParola); i++)
            printf("%c", primaParola[i]);
         for (int i = 0; i < strlen(secondaParola)-lung; i++)
            printf("%c", secondaParola[i]);
         printf("\n");
   }
   return 1;
} // end main
```