

Nome: _____ Cognome: _____ Matr.: _____
Fila: ____ Posto: ____

Parte C

C/Esercizio 1 (4 punti)

Scrivere una funzione **match** con due puntatori a stringa come parametri. Il primo è il testo che si vuole riconoscere, il secondo è il pattern. La funzione ritorna 1 se il testo rispetta il pattern altrimenti 0. Il pattern è una sequenza di coppie: carattere, quantificatore. Il quantificatore dice quante volte il carattere può comparire di seguito nel testo (in quel punto): '+' significa 1 o più volte, '?' significa nessuna volta oppure una. Il pattern viene valutato da sinistra a destra; i quantificatori consumano tutti i caratteri del testo compatibili con la definizione. Ad esempio: "a?b+" significa che il testo può cominciare con una 'a' e poi contiene almeno una 'b'; le stringhe "bbbb" e "ab" sono valide mentre "aabb", "a" e "abc" non lo sono.

C/Esercizio 2 (4 punti)

Si scriva una funzione **merge** con due puntatori ad una stessa stringa come parametri: il primo punta all'inizio e il secondo a metà +1. Le due metà della stringa sono ordinate in modo crescente e la funzione non restituisce nulla, modifica la stringa stessa rendendola completamente ordinata. Ad esempio data la stringa **"bdefacxy"** (con i due puntatori che puntano rispettivamente alla b e alla a; la funzione modifica la stringa come segue: **"abcdefxy"** .

Nome: _____ Cognome: _____ Matr.: _____
Fila: ____ Posto: ____

Parte C

C/Esercizio 1 (4 punti)

Scrivere una funzione **match** con due puntatori a stringa come parametri. Il primo è il testo che si vuole riconoscere, il secondo è il pattern. La funzione ritorna 1 se il testo rispetta il pattern altrimenti 0. Il pattern è una sequenza di coppie: carattere, quantificatore. Il quantificatore dice quante volte il carattere può comparire di seguito nel testo (in quel punto): '*' significa 0 o più volte, '!' significa una volta. Il pattern viene valutato da sinistra a destra; i quantificatori consumano tutti i caratteri del testo compatibili con la definizione. Ad esempio: "a!b*" significa che il testo comincia con una 'a' e poi può contenere un numero a piacere di 'b'; le stringhe "abbbb" e "a" sono valide mentre "aabbb", "aa" e "abc" non lo sono.

C/Esercizio 2 (4 punti)

Si scriva una funzione **merge** con due puntatori ad una stessa stringa come parametri: il primo punta a metà e il secondo alla fine. Le due metà della stringa sono ordinate in modo crescente e la funzione non restituisce nulla, modifica la stringa stessa rendendola completamente ordinata. Ad esempio data la stringa **"bdefacxy"** (con i due puntatori che puntano rispettivamente alla f e alla y; la funzione modifica la stringa come segue: **"abcdefxy"** .

Nome: _____ Cognome: _____ Matr.: _____
Fila: ____ Posto: ____

Parte C

C/Esercizio 1 (4 punti)

Scrivere una funzione **match** con due puntatori a stringa come parametri. Il primo è il testo che si vuole riconoscere, il secondo è il pattern. La funzione ritorna 1 se il testo rispetta il pattern altrimenti 0. Il pattern è una sequenza di coppie: carattere, quantificatore. Il quantificatore dice quante volte il carattere può comparire di seguito nel testo (in quel punto): '*' significa 0 o più volte, '?' significa nessuna volta oppure una. Il pattern viene valutato da sinistra a destra; i quantificatori consumano tutti i caratteri del testo compatibili con la definizione. Ad esempio: "a?b*" significa che il testo può cominciare con una 'a' e poi può contenere un numero a piacere di 'b'; le stringhe "abb", "bb" e "" sono valide mentre "aabb", "aa" e "bc" non lo sono.

C/Esercizio 2 (4 punti)

Si scriva una funzione **merge** con due puntatori ad una stessa stringa come parametri: il primo punta all'inizio e il secondo a metà +1. Le due metà della stringa sono ordinate in modo decrescente e la funzione non restituisce nulla, modifica la stringa stessa rendendola completamente ordinata. Ad esempio data la stringa **"fedbyxca"** (con i due puntatori che puntano rispettivamente alla f e alla y; la funzione modifica la stringa come segue: **"yxfedcba"**).