

Esame 20250616

Esercizio 2

(1) Esercizio 2 v1

Una keypad é un blocco o un pad di pulsanti impostato con una disposizione di cifre, simboli o lettere alfabetiche. I keypad contenenti principalmente numeri sono utilizzati con i vecchi cellulari sono keypad numerici. Le keypad si trovano su dispositivi che richiedono principalmente input numerici come calcolatrici, telecomandi televisivi, telefoni a pulsanti, distributori automatici, bancomat, terminali di punti vendita, serrature a combinazione, casseforti e serrature digitali. La disposizione delle cifre su un keypad numerico segue tipicamente lo standard E.161, come illustrato nella figura seguente:



Secondo questo layout, per ottenere sullo schermo il carattere 'y' si deve premere il tasto 9 quattro (4) volte. Per ottenere il carattere 'z' si deve premere il tasto 9 cinque (5) volte. Per ottenere il carattere '1' si deve premere il tasto 1 una (1) volta. Per ottenere il carattere '0' si deve premere il tasto 0 una (1) volta. Se devo inserire due caratteri che sono associati allo stesso tasto numerico, in pratica devo aspettare qualche istante prima di premere di nuovo il tasto. I caratteri associati sono considerati in modo ciclico. Ad esempio se premo il tasto 9 sei (6) volte ottengo il carattere '9', se premo il tasto 7 sette (7) volte ottengo il carattere 'w'.

Assumendo che la mappa dei tasti sia memorizzata in un array di 10 elementi, dove ogni elemento in posizione i contiene una struttura dati Key composta da un intero e da una lista concatenata List * di caratteri associati al tasto numerico i (l'intero nella struttura Key rappresenta la lunghezza della lista di caratteri, il terminatore di lista é nullptr), si vuole scrivere un programma che, dato una stringa di caratteri numerici e spazi (usati per separare due caratteri che appartengono allo stesso tasto, simulando il passare di un istante di tempo), restituisce il testo corrispondente.

Il programma in `esercizio2.cpp` prende come argomento a) una stringa che rappresenta la sequenza di caratteri numerici e spazi e stampa a video il testo corrispondente. La mappa dei tasti é rappresentata in memoria con un array di 10 elementi in cui ogni elemento é di tipo Key che contiene un intero e una lista concatenata di caratteri. La stringa di input é rappresentata in memoria con un array di caratteri.

Il `main` del programma é già implementato e non deve essere modificato, e chiama la funzione `decode` (**da definire**) che prende come argomento una stringa di caratteri, un intero che rappresenta la dimensione della stringa di caratteri, e l'array che rappresenta la mappa dei tasti e stampa a video il testo corrispondente. **La funzione decode stampa a video la decodifica della stringa rispetto alla mappa data.**

La funzione `decode` **deve essere ricorsiva** e **NON deve contenere iteratori esplicativi** (`for`, `while`, `do-while`). La funzione `decode` può ovviamente contenere codice sequenziale o

condizionale. Sono consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non contengano iterazioni esplicite** (`for`, `while`, `do-while`).

Il file `esercizio2.cpp` contiene tutto quanto necessario tranne la dichiarazione e la definizione della procedura `decode`.

Sia data la seguente mappa dei tasti:

```
0 -> "0abc",
1 -> "1def",
2 -> "2ghi",
3 -> "3jkl",
4 -> "4mno",
5 -> "5pqrs",
6 -> "6tuv",
7 -> "7wxyz",
8 -> "8 .,@*=-",
9 -> "9"
```

di seguito sono riprotati due esempi di esecuzione del programma.

```
computer > ./a.out "00000 000988 44 44499997 7777777"
Decoding string: "00000 000988 44 44499997 7777777"
Decoded string: "0b9 mn97w"
computer > ./a.out "440055550000444444485555444666611155552222222"
Decoding string: "440055550000444444485555444666611155552222222"
Decoded string: "marco roveri"
```

Note:

- Scaricare il file `esercizio2.cpp`, modificarlo per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- Si può assumere che la stringa di input contenga solo caratteri numerici e spazi.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `fstream`, `cstdlib`, `sstream`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).
- Si ricorda che la soluzione deve essere implementata in C++ **NON usando `string` e altri elementi della C++ standard template library, anche se il file compila senza cambiare gli header!**

esercizio2.cpp

Information for graders:

(2) Esercizio 2 v2

marked out of 10 penalty 0 File picker

Una keypad é un blocco o un pad di pulsanti impostato con una disposizione di cifre, simboli o lettere alfabetiche. I keypad contenenti principalmente numeri sono utilizzati con i vecchi cellulari sono keypad numerici. Le keypad si trovano su dispositivi che richiedono principalmente input numerici come calcolatrici, telecomandi televisivi, telefoni a pulsanti, distributori automatici, bancomat, terminali di punti vendita, serrature a combinazione, casseforti e serrature digitali. La disposizione delle cifre su un keypad numerico segue tipicamente lo standard E.161, come illustrato nella figura seguente:



Secondo questo layout, per ottenere sullo schermo il carattere 'y' si deve premere il tasto 9 quattro (4) volte. Per ottenere il carattere 'z' si deve premere il tasto 9 cinque (5) volte. Per ottenere il carattere '1' si deve premere il tasto 1 una (1) volta. Per ottenere il carattere '0' si deve premere il tasto 0 una (1) volta. Se devo inserire due caratteri che sono associati allo stesso tasto numerico, in pratica devo aspettare qualche istante prima di premere di nuovo il tasto. I caratteri associati sono considerati in modo ciclico. Ad esempio se premo il tasto 9 sei (6) volte ottengo il carattere '9', se premo il tasto 7 sette (7) volte ottengo il carattere 'w'.

Assumendo che la mappa dei tasti sia memorizzata in un array di 10 elementi, dove ogni elemento in posizione i contiene una struttura dati Key composta da un intero e da una lista concatenata List * di caratteri associati al tasto numerico i (l'intero nella struttura Key rappresenta la lunghezza della lista di caratteri, il terminatore di lista é nullptr), si vuole scrivere un programma che, dato una stringa di caratteri numerici e spazi (usati per separare due caratteri che appartengono allo stesso tasto, simulando il passare di un istante di tempo), restituisce il testo corrispondente.

Il programma in `esercizio2.cpp` prende come argomento a) una stringa che rappresenta la sequenza di caratteri numerici e spazi e stampa a video il testo corrispondente. La mappa dei tasti é rappresentata in memoria con un array di 10 elementi in cui ogni elemento é di tipo Key che contiene un intero e una lista concatenata di caratteri. La stringa di input é rappresentata in memoria con un array di caratteri.

Il `main` del programma é già implementato e non deve essere modificato, e chiama la funzione `decode` (**da definire**) che prende come argomento una stringa di caratteri, un intero che rappresenta la dimensione della stringa di caratteri, e l'array che rappresenta la mappa dei tasti e stampa a video il testo corrispondente. **La funzione decode stampa a video la decodifica della stringa rispetto alla mappa data.**

La funzione `decode` **deve essere ricorsiva e NON deve contenere iteratori esplicativi** (`for`, `while`, `do-while`). La funzione `decode` può ovviamente contenere codice sequenziale o condizionale. Sono consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non contengano iterazioni esplicite** (`for`, `while`, `do-while`).

Il file `esercizio2.cpp` contiene tutto quanto necessario tranne la dichiarazione e la definizione della procedura `decode`.

Sia data la seguente mappa dei tasti:

```
0 -> "0jkl",
1 -> "1def",
2 -> "2ghi",
3 -> "3abc",
4 -> "4pqrs",
5 -> "5mno",
6 -> "6tuv",
7 -> "7wxyz",
8 -> "8",
9 -> "9 .,@*+=-"
```

di seguito sono riportati due esempi di esecuzione del programma.

```
computer > ./a.out "00000 000988 44 44499997 7777777"
Decoding string: "00000 000988 44 44499997 7777777"
Decoded string: "0k98pq,7w"
computer > ./a.out "55334444444443333555599444455556666111111111144442222"
Decoding string: "55334444444443333555599444455556666111111111144442222"
Decoded string: "marco roveri"
```

Note:

- Scaricare il file `esercizio2.cpp`, modificarlo per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- Si può assumere che la stringa di input contenga solo caratteri numerici e spazi.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `fstream`, `cstdlib`, `sstream`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).
- Si ricorda che la soluzione deve essere implementata in C++ **NON usando `string` e altri elementi della C++ standard template library, anche se il file compila senza cambiare gli header!**

`esercizio2.cpp`

Information for graders:

Total of marks: 20