Cognome: **Nome**: **# Matricola**: **Riga**: **Col**:

Algoritmi e Strutture Dati - 19/12/14

Esercizio 0 Scrivere correttamente nome, cognome, numero di matricola, riga e colonna.

Esercizio 1 – Punti ≥ 6

Dato un vettore V di n numeri reali, bisogna calcolare il numero minimo di intervalli di lunghezza *unitaria* in grado di ricoprire tutti i valori di V. Un intervallo di lunghezza unitaria è definito da una coppia $[x, x+1]$, estremi inclusi. Ad esempio se $n = 5$ e $V = \{2.5, 3.8, 1.5, 3.1, 1.8\}$, allora il numero minimo di intervalli unitari è 2 e una possibile soluzione è $[1.5, 2.5], [3, 4]$. Descrivere un algoritmo **integer** minIntervals(**real**[ ] V, **integer** n) che risolve il problema, discuterne la correttezza e valutare la complessità.

Esercizio 2 – Punti ≥ 6

Non contenti del difficile passaggio al Regolamento 14/15, i docenti del DISI hanno realizzato il nuovo Regolamento 15/16, così complicato che già solo capire quali corsi seguire è prova che lo studente è pronto per laurearsi.
Il regolamento è così descritto. Esiste un insieme C di n corsi; esistono k requisiti, dati dalle coppie $(m_1, S_1), (m_2, S_2), \ldots, (m_k, S_k)$ dove m_i è un intero e S_i è un sottoinsieme di C. Per soddisfare il requisito (m_i, S_i), lo studente deve inserire nel proprio piano di studi m_i corsi presi dal sottoinsieme S_i. Notate che un corso può comparire in più requisiti, nel qual caso può essere utilizzato per soddisfare al massimo un requisito (e non conta per gli altri). Lo studente deve seguire in totale t corsi per laurearsi.
Ad esempio, con $t = 5$ e la lista di $n = 7$ corsi rappresentata dalle lettere `A,B,C,D,E,F,G`, i requisiti potrebbero essere:

```
2  A B C             "base"
2  B C D E           "caratterizzanti"
1  A B C D E F G     "scelta libera"
```

è possibile soddisfare i requisiti facendo i corsi `A,B` per il primo requisito, `C,D` per il secondo, `F` per il terzo. Notate che i corsi `B,C` possono soddisfare tutti i requisiti, ma abbiamo scelto che `B` sia contato nei base e `C` nei caratterizzanti.
Avete il dubbio che il regolamento sia così assurdo che non sia possibile soddisfarlo; descrivete un algoritmo per determinare se è possibile soddisfare le richieste del regolamento, valutandone poi la complessità.

Esercizio 3 – Punti ≥ 9

Si consideri una stringa S composta da n caratteri nell'alfabeto $A - Z$. Una sottosequenza di S è *ordinata-distinta* se è composta solo da caratteri ordinati alfabeticamente e tutti distinti. Vi ricordo che una sottosequenza è ottenuta rimuovendo caratteri dalla stringa originale, lasciando i restanti nell'ordine originale. Siamo interessati alle più lunghe sottosequenze ordinate-distinte. Come esempi, si considerino le seguenti stringhe, dove alcune delle sottosequenze ordinate-distinte più lunghe sono evidenziate in grassetto.

- `RSTU`**`A`**`RSTU`**`B`**`RSTU`**`C`**`RSTU`**`DRSTU`**`E` → **ABCDRSTU**
- **C**`BA` → **C**

Data una stringa S di lunghezza n, scrivere un algoritmo maxOrdinataDistinta(**integer**[ ] S, **integer** n) che restituisca la lunghezza delle sottosequenza ordinata-distinta di S più lunga. Discuterne correttezza e complessità.
Suggerimento: se vi è comodo, potete assumere che i caratteri $A - Z$ siano rappresentanti dai codici $1, \ldots, 26$. Notate che esiste una soluzione di 3 righe di codice che utilizza codice esistente.

Esercizio 4 – Punti ≥ 9

Data due stringhe, è possibile "allinearle" inserendo dei caratteri dash "–" in una o nell'altra, in modo tale che a caratteri uguali corrispondano caratteri uguali, oppure ad un carattere di una stringa corrisponda un dash nell'altra. Ad esempio, le stringhe `ABC` e `ACE` possono essere allineate con 2 dash, mentre le stringhe `questoèunoscempio` e `unesempio` possono essere allineate con 10 dash:

```
ABC-      questoèun-oscempio
A-CE      -------une-s-empio
```

Date due stringhe $S[1 \ldots n]$ e $P[1 \ldots m]$, scrivere un algoritmo **integer** minAllineamento(ITEM[ ] P, ITEM[ ] S, **integer** n, **integer** m) che restituisca il minimo numero di caratteri dash necessari per allineare le due stringhe. Discuterne correttezza e complessità.