Sapienza Università di Roma
Corso di Laurea in Ingegneria Informatica e Automatica A.A. 2023-24

Esercitazione Python n. 5 -- 31 Ottobre 2023

Obiettivo dell'esercitazione è prendere confidenza con l'uso delle funzioni in Python, ripassare le istruzioni **for**, **if-then-else, while,** ripassare le funzioni ed esercitarsi con le istruzioni **break** e **continue**.

In laboratorio, per avviare la macchina virtuale, dovete selezionarla all'interno della schermata di VirtualBox (schermata visualizzata all'accensione del PC), cliccando sull'opzione **LXLE-BIAR-4.5**. Una volta avviata la macchina virtuale, svolgete gli esercizi così come indicato nel testo. Usate l'ambiente Spyder per svolgere gli esercizi. Ovviamente è possibile consultare il materiale didattico disponibile sulla pagina web del corso (https://classroom.google.com/u/1/c/NjIwOTY0ODk3MDAx). Si ricorda che le note relative alle lezioni Python possono essere lette con l'applicazione JupyterLab.

La consegna deve essere effettuata entro l'orario di fine dell'esercitazione.

LE ESERCITAZIONI SVOLTE CONSEGNATE OLTRE QUESTO TERMINE, O CHE NON RISPETTANO IL FORMATO INDICATO PER LA CONSEGNA, NON VERRANNO CONSIDERATE. In particolare, vi chiediamo di NON caricare un esercizio svolto per volta, di NON usare formati di compressione diversi da .zip, di NON rinominare i file o metterli in sottocartelle.

Fate attenzione che gli input siano richiesti all'utente UNO PER VOLTA e NELL'ORDINE RIPORTATO nell'esercizio e che le vostre stampe riportino a video i messaggi ESATTAMENTE nel formato atteso.

Ogni esercizio richiede che sia completata una funzione all'interno del file predisposto con lo stesso nome e, come visto a lezione, può essere testata eseguendo il file stesso, con i casi di test forniti. Notate infatti che ogni file incorpora del codice python per eseguire alcuni test sulla funzione. **NON modificate questo codice, ma SCRIVETE SOLO il contenuto della funzione. Non spostate i file dalla loro posizione e non create nuovi file. Si noti che, come in sede di esame, per la correzione verranno usati insiemi di dati di test diversi.**

Esercizi

1) **A_Ex1(s, n)** Scrivere una funzione che prende in ingresso una stringa **e** un intero **n** e restituisce la sottosequenza di **s** di lunghezza **n** tale per cui la somma dei codici unicode dei caratteri che la costituiscono è massima. Se la stringa ha una lunghezza inferiore ad **n** restituisce la stringa vuota.

Esempio:

- Se **s** è 'scacco matto' e **n** è 3, la funzione deve restituire la stringa 'tto'
- Se **s** è 'scacco' e **n** è 3 la funzione deve restituire la stringa 'sca'
- Se, **s** è 'scacco' e **n** è 7 la funzione deve restituire la stringa vuota ".

2) **A_Ex2(s)** Scrivere una funzione che prende in ingresso una stringa **s** e restituisce una stringa composta da tutti e soli i caratteri alfabetici maiuscoli contenuti in s, senza ripetizioni. La stringa restituita deve essere ordinata in ordine alfabetico (cioè secondo l'ordine UNICODE crescente).
Esempio:

- Se **s** è 'cIAo MAmMa' e **n** è 3, la funzione deve restituire la stringa 'AIM'.

3) **A_Ex3(s)** Scrivere una funzione che prende in ingresso una stringa **s** e restituisce una stringa ottenuta "slittando circolarmente" tutti i caratteri di **s** di un numero di posizioni tali per cui la somma $S=i*c_i$ è massima, dove c_i è il codice UNICODE del carattere di **s** in posizione i-esima, con i=0,1,…n-1, dove n è la lunghezza di **s** e 0 è la posizione più a sinistra.
Esempio:

- Se **s** è 'mamma', la funzione restituisce 'amamm'; infatti, slittando circolarmente tutti i caratteri di **s** di 1 posizione verso destra, otteniamo la stringa 'amamm' per la quale S= 1066; slittandoli di 2 posizioni, otteniamo la stringa 'mamam' per la quale S= 1042; slittandoli di 3 posizioni, otteniamo la stringa 'mmama' per la quale S= 1018; slittandoli di 4 posizioni, otteniamo la stringa 'ammam' per la quale S= 1054; slittandoli di 5 posizioni, otteniamo la stringa 'mamma' per la quale S= 1030.

4) **A_Ex4(n)** Scrivere una funzione che prende in ingresso un numero intero positivo **n** e restituisce True se le sue cifre sono tutti numeri pari, False altrimenti. Per risolvere l'esercizio usare l'istruzione **break**.
Esempio:

- Se **n** è 2416, la funzione restituisce False;
- Se **n** è 4860, la funzione restituisce True.

5) **A_Ex5(n)** Scrivere una funzione che prende in ingresso un numero intero **n** e restituisce una stringa che rappresenta una scacchiera nXn su **n** righe di **n** caratteri ognuna che contiene il carattere ' ' (spazio) per le caselle bianche, il carattere '*' per le caselle nere e comincia con una casella bianca. Controllare che **n** sia compatibile con il requisito che la funzione crei sempre una scacchiera con almeno una casella nera e una bianca, restituendo 'ERRORE' se ciò non è possibile.
Esempio:

- Se **n** è 2, la funzione restituisce la stringa ' *\n* \n';
- Se **n** è 3, la funzione restituisce la stringa ' * \n* *\n * \n'
- Se **n** è 4, la funzione restituisce la stringa:
' * *\n* * \n * *\n* * \n'

6) **A_Ex6(v1)** Scrivere una funzione che prende in input due interi **v1** e **v2**, superiori o uguali a 1, che rappresentano rispettivamente la velocità di partenza di due viaggiatori, e restituisce il numero di giorni necessario al secondo viaggiatore per raggiungere il primo, assumendo che il primo viaggi a velocità costante mentre il secondo viaggi alla velocità di partenza il primo giorno, al doppio della velocità di partenza il secondo giorno, al triplo il terzo giorno e così via (problema posto da Fibonacci nel 1200).

Esempio:

- Se **v1**=20 e **v2**=1, la funzione restituisce 39;
- Se **v1**=10 e **v2**=5, la funzione restituisce 3;
- Se **v1**=1 e **v2**=1, la funzione restituisce 1.

7) **A_Ex7(b1,b2)** Scrivere una funzione che prende in input due stringhe **b1** e **b2** contenenti solo caratteri '0' ed '1' (interpretati come bit) e restituisce una stringa di '0' ed '1' che corrisponde all'AND bit a bit di **b1** e **b2** letti da sinistra a destra. Si noti che l'AND bit a bit effettua l'AND solo dei bit che occupano la stessa posizione nelle stringhe **b1** e **b2**, e per ogni posizione i produce 1 solo se entrambi i bit in posizione i in **b1** e **b2** sono pari ad 1. Per i bit di b1 che non hanno un corrispondente bit in **b2** (in questo caso **b1** è più lunga di **b2**) la funzione deve calcolare l'AND con lo '0'. Analogamente nel caso in cui **b2** sia più lunga di **b1**.

Esempio:

- Se **b1** vale '100' e **b2** vale '1011', la funzione deve restituire la stringa '1000'
- Se **b1** vale '' e **b2** vale '111' la funzione deve restituire '000'
- Se entrambe **b1** e **b2** sono vuote, la funzione deve restituire la stringa vuota.

8) **A_Ex8(n)** Scrivere una funzione che prende in ingresso un numero intero **n** e restituisce una stringa che rappresenta un quadrato nXn dove vengono disegnati pieni solo i bordi e le due diagonali, tutte le altre caselle sono vuote. Usate il carattere ' ' (spazio) per le caselle vuote ed il carattere '*' per le caselle piene. Controllare che il numero n sia strettamente positivo, restituendo 'ERRORE' in caso contrario.

Esempio:

- Se **n** è 2, la funzione restituisce la stringa '**\n**', che corrisponde al disegno:

```
**
**
```

- Se **n** è 5, la funzione restituisce la stringa '*****\n** **\n* * *\n** **\n*****\n', che corrisponde al disegno:

```
*****
** **
* * *
** **
*****
```

- Se **n** è 6, la funzione restituisce la stringa '******\n** **\n* ** *\n* ** *\n** **\n******\n', che corrisponde al disegno:

```
******
**  **
* ** *
* ** *
**  **
******
```

9) **A_Ex9(s1,s2)** Scrivere una funzione che riceve in ingresso due stringhe **s1** ed **s2** e calcola la lunghezza della più lunga sottosequenza comune alle due stringhe.

Esempi:

- Se **s1** vale 'casa' ed **s2** vale 'cappello' la funzione deve restituire 2.
- Se **s1** vale ' carcassa' ed **s2** vale 'cappello' la funzione deve restituire 2.

10) **A_Ex10(s)** Scrivere una funzione che riceve in ingresso una stringa **s** (composta da caratteri qualunque) e restituisce una stringa composta da tutti i caratteri che compaiono in **s**, in ordine crescente di codice UNICODE e senza ripetizioni.

Esempi:

- Se **s** vale 'casale' la funzione deve restituire 'acels';
- Se **s** vale 'Tanto va la Gatta' la funzione deve restituire ' GTalnotv';