Prova Scritta del 21-01-2021

Esercizio 1.

All'interno di una schema risolto di parole crociate è stata nascosta una frase da individuare della quale si conosce l'elenco ordinato $\{L_1 \ldots L_n\}$ delle lunghezze di ogni parola. Le n parole che compongono la frase si trovano orizzontalmente o verticalmente nello schema e possono essere facilmente individuate se si conosce per ognuna di esse la posizione della prima lettera e l'orientamento (orizzontale o verticale). Sfortunatamente, anche queste informazioni sono state nascoste, questa volta all'interno di un testo (una stringa). In particolare, all'interno del testo, in mezzo ad altri caratteri, si trovano n sequenze $S_1 \ldots S_n$ di tre caratteri consecutivi, che chiamiamo ***valide***; ogni sequenza S_i consente di identificare la corrispondente parola L_i e l'ordine in cui si trovano le sequenze nel testo è lo stesso dell'ordine in cui si trovano le corrispondenti parole nella frase (si trova prima la sequenza S_1, poi la sequenza S_2, e così via). Ogni sequenza valida è formata da tre caratteri consecutivi del tipo $x_i y_i c_i$, dove 1) $x_i y_i$ sono cifre che rappresentano una posizione valida all'interno dello schema dove x_i è la riga e y_i la colonna, 2) $x_i y_i$ indica l'inizio di una parola dello schema (non una lettera centrale), 3) la parola che inizia in $x_i y_i$ secondo l'orientamento c_i (che può essere o se orizzontale, v se verticale) ha una lunghezza uguale a L_i. Si noti che all'interno della stringa potrebbero essere presenti sequenze di tre caratteri consecutivi non valide, ossia che non rispettano le condizioni 1), 2) e 3) indicate sopra, e che pertanto non possono essere usate per ricostruire la parola.

Si scriva un programma Python, opportunamente modularizzato in funzioni, che prese da input una matrice di caratteri che rappresenta lo schema (già risolto) di parole crociate, una lista di interi che rappresenta l'elenco delle lunghezze delle parole ed una stringa di caratteri contenente le sequenze valide stampi la parola nascosta nello schema. Si può supporre che l'input sia corretto e che per ogni parola esista una sola sequenza valida che la identifica. Inoltre, si può supporre che la matrice dello schema contenga caratteri '*' per rappresentare le caselle nere.

ESEMPIO

Si consideri lo schema di parole crociate riportato a destra e si supponga che l'elenco con le lunghezze delle parole sia {2,5,2,4} e il testo nel quale sono nascoste le sequenze valide sia "01o.gs*13o11vcasa01v15oooo10oxs:130osalto". La frase nascosta nello schema è UN LIBRO DI ARTE e le sequenze valide che identificano le parole che la compongono sono "13o", "01v", "10o" e "30o". Si noti che "01o" non è una sequenza valida perché la parola corrispondente (LUNA) ha una lunghezza diversa da 2 che è la lunghezza della prima parola. Inoltre, "11v" non è una

	0	1	2	3	4
0	■	L	U	N	A
1	D	I	■	U	N
2	■	B	E	T	A
3	A	R	T	E	■
4	D	O	■	L	■
5	E	■	G	L	I
6	■	G	R	A	N

CAMPUS DI ARCAVACATA http://www.unical.it/portale/strutture/dipartimenti_240/matinf/
87036 Arcavacata di Rende (CS) - Ponte Pietro Bucci Cubo 30B
tel. (+39) 0984 496402 - fax (+39) 0984 496410

Prova Scritta del 21-01-2021

sequenza valida perché la posizione 1,1 non corrisponde all'inizio di una parola e "15v" non è una sequenza valida perché 1,5 non è una posizione interna allo schema.

SOLUZIONE

```
def leggiTabella(r, c):
  tabella = []
  for i in range(r):
    tabella.append([])
    for j in range(c):
      tabella[i].append(input("Inserire carattere "))
  for i in range(r):
    print(tabella[i])
  return tabella

def leggiLunghezze(l):
  lunghezze = []
  for i in range(l):
    lunghezze.append(int(input("Inserire la lunghezza della prossima
parola ")))
  return lunghezze

def trovaSequenzeValide(sequenza,tabella):
  sv = []
  for i in range(len(sequenza)-2):
    if    sequenza[i].isdigit()    and    sequenza[i+1].isdigit()    and
(sequenza[i+2] == 'o' or sequenza[i+2] == 'v'):
      r=int(sequenza[i])
      c=int(sequenza[i+1])
      if (r<len(tabella) and c<len(tabella[0])):
        if sequenza[i+2] == 'o' and (c==0 or tabella[r][c-1]=='*'):
          sv.append([r,c,sequenza[i+2]])
        elif sequenza[i+2] == 'v' and (r==0 or tabella[r-1][c]=='*'):
          sv.append([r,c,sequenza[i+2]])
  return sv

def paroleNascoste(tabella, lunghezze, sequenzeValide):
  parole = []
  n = 0
  for i in range(len(sequenzeValide)):
    if n == len(lunghezze):
      break

    r = sequenzeValide[i][0]
    c = sequenzeValide[i][1]

    if sequenzeValide[i][2] == 'o':
```


CAMPUS DI ARCAVACATA http://www.unical.it/portale/strutture/dipartimenti_240/matinf/
87036 Arcavacata di Rende (CS) - Ponte Pietro Bucci Cubo 30B
tel. (+39) 0984 496402 - fax (+39) 0984 496410

Prova Scritta del 21-01-2021

```
      if ((c + lunghezze[n]) > len(tabella[0])) or ((c + lunghezze[n]) <
len(tabella[0]) and tabella[r][(c+lunghezze[n])]!='*'):
        continue
      parola = ""
      j=0
      while (c+j)<len(tabella[0])and tabella[r][c+j]!='*':
        parola += tabella[r][c+j]
        j+=1

      if len(parola)==lunghezze[n]:
        parole.append(parola)
        n += 1

    elif sequenzeValide[i][2] == 'v':
      if ((r + lunghezze[n]) > len(tabella)) or ((r + lunghezze[n]) <
len(tabella) and tabella[(r+lunghezze[n])][c]!='*'):
        continue
      parola = ""
      j=0
      while ((r+j)<len(tabella) and tabella[r+j][c]!='*'):
        parola += tabella[r+j][c]
        j+=1

      if len(parola)==lunghezze[n]:
        parole.append(parola)
        n += 1

  return parole

def main():
  # Lettura tabella caratteri
  r = int(input("Inserire il numero di righe della tabella: "))
  c = int(input("Inserire il numero di colonne della tabella: "))
  tabella = leggiTabella(r, c)

  # Lettura lista lunghezze parole
  n = int(input("Inserire il numero di parole per le quali leggere le
lunghezze: "))
  lunghezze=leggiLunghezze(n)

  # Lettura testo posizioni nascoste
  pn = input("Inserire la stringa che codifica le posizioni delle parole
nascoste ")

  sequenzeValide = trovaSequenzeValide(pn,tabella)

  print(sequenzeValide)
```


CAMPUS DI ARCAVACATA http://www.unical.it/portale/strutture/dipartimenti_240/matinf/
87036 Arcavacata di Rende (CS) - Ponte Pietro Bucci Cubo 30B
tel. (+39) 0984 496402 - fax (+39) 0984 496410

Prova Scritta del 21-01-2021

```
  frase=paroleNascoste(tabella, lunghezze, sequenzeValide)

  if (len(frase)==len(lunghezze)):
    print("La frase nascosta e': ", end='')
    for i in range(len(frase)):
      print (frase[i],' ',end='')
  else:
    print("Nessuna frase nascosta")

main()
```

Esercizio 2.

Scrivere in Python una funzione ricorsiva booleana che dati due numeri N ed M restituisce ***True*** se almeno una delle rotazioni delle cifre di N genera un numero divisibile per M e restituisce ***False*** altrimenti. Oltre a N e M, è possibile aggiungere alla funzione ulteriori parametri se necessario. Si può supporre che il numero N non contenga alcuna cifra pari a 0.

N.B. L'implementazione corretta della funzione senza la ricorsione vale metà punteggio.

ESEMPIO

Se N fosse 324 e M fosse 8 la funzione restituirebbe True. Infatti, i numeri che si ottengono dalla rotazione delle cifre di N sono, oltre a 324, 243 e 432 e tra questi 432 è divisibile per 8. Se N fosse 421 e M fosse 8 la funzione restituirebbe False. Infatti le rotazioni delle cifre di N generano i numeri 421, 214 e 142 e nessuno di questi è divisibile per 8.

Bonus: Si modifichi la funzione affinchè sia possibile gestire casi in cui il numero N contiene almeno una cifra pari a 0 (ad esempio N=420, per il quale le rotazioni sono 420, 204 e 042).

SOLUZIONE 1

I numeri sono trattati come interi, e la rotazione avviene scomponendo in cifre e ricomponendo dopo la rotazione.

```
def verificaAggiuntaZeri(N,Cifre):
    X=N
    CifreN=[]
    while X>0:
        c=X%10
```


CAMPUS DI ARCAVACATA http://www.unical.it/portale/strutture/dipartimenti_240/matinf/
87036 Arcavacata di Rende (CS) - Ponte Pietro Bucci Cubo 30B
tel. (+39) 0984 496402 – fax (+39) 0984 496410

```
        X=int(X/10)
        CifreN.append(c)
    if len(CifreN)!=len(Cifre):
        Cifre.append(0)

def ruota(N,M,CN):
    if CN%M==0:
        return True
    Cifre=[]
    while CN>0:
        c=CN%10
        CN=int(CN/10)
        Cifre.append(c)
    verificaAggiuntaZeri(N,Cifre)
    Cifre.insert(0,Cifre[-1])
    Cifre.pop()
    CN=0
    for i in range(len(Cifre)):
        CN+=Cifre[i]*10**i
    if N==CN:
        return False
    return ruota(N,M,CN)

print(ruota(324,8,324))
print(ruota(420,8,420))
```

SOLUZIONE 2

I numeri sono trattati come stringhe.

```
def ruota(N,M,cont):

    if cont == len(N):
        return False
    if int(N)%M==0:
        return True

    N=N[1:]+N[0]

    return ruota(N,M,cont+1)

N=input()
M=int(input())
print(ruota(N,M,0))
```


CAMPUS DI ARCAVACATA http://www.unical.it/portale/strutture/dipartimenti_240/matinf/
87036 Arcavacata di Rende (CS) - Ponte Pietro Bucci Cubo 30B
tel. (+39) 0984 496402 - fax (+39) 0984 496410