

- Tentan har flervalsfrågor där minst ett svarsalternativ är korrekt. Om man svarar fel eller inte har exakt antal rätta alternativ får man noll poäng på frågan.
- Man måste bli godkänd på del A (4 rätt på 8 frågor) för att del B ska rättas.
- Del B består av frågor med varierande poäng (totalt 12 poäng).
- Inga import (Pythons standardbibliotek eller externa bibliotek) får användas om de inte nämns eller finns med i uppgiften. Man får använda inbyggda funktioner som len, range och map.
- All kod avser **Python 3**, dvs *inte* t.ex. Python 2.7
- **Hjälpmedel:** Ett A4 med så mycket information du vill. Du får skriva på båda sidorna.
- **Betygsgränser:** E: 10, D: 12, C: 14, B: 16, A: 18, av maximala 20.

Del A: flervalsfrågor

Var snäll samla svaren på del A på ett svarpapper.

1. Vad innehåller `s` efter att man kört kodsnutten till höger?

- A. ''
- B. 'tillLycka'
- C. 'Lyckatill'
- D. 'akcyLllit'
- E. 'llitakcyL'

```
s = ''  
  
for x in ['Lycka', 'till']:  
    for y in x:  
        s = y + s
```

2. Vilka av följande nyckelord i Python används för felhantering?

- A. raise
- B. yield
- C. try
- D. except
- E. break

3. Givet `xs = [0, 1, 2, 3, 4, 5]`, vilket eller vilka av följande **print**-uttryck skriver ut listan `[1, 3, 5]`?

- A. `print([x for x in xs if x % 2 != 0])`
- B. `print(xs[1][3][5])`
- C. `print(xs[0:1:3])`
- D. `print([xs[1], xs[3], xs[5]])`
- E. `print([xs[1]] + [xs[3]] + [xs[5]])`

4. Vad skrivs ut av koden till höger?

- A. 3
- B. 4
- C. 5
- D. 6
- E. Inget då det blir ett särfall eftersom man inte får skriva `x += x`

```
x = 3  
  
def f(x, y=2):  
    x += x  
    print(x + y)  
  
f(1)
```

5. Om `d = ('e', [4, 'e'], 'tenta', { 'a': 1, 2 : 'e' })`, vilket eller vilka av följande alternativ kan man skriva för att komma åt ett 'e'?

- A. `d[0]`
- B. `d[1][2]`
- C. `d[2][1]`
- D. `d[3]['e']`
- E. `d[3][2]`

6. Vad blir resultatet av koden till höger?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

```
def f(x):  
    if x == 0:  
        return x  
    elif x % 2 == 0:  
        return x + f(x-1)  
    else:  
        return x - f(x-1)  
  
print(f(5))
```

7. Hur många gånger anropas `print` när man kör koden till höger?

- A. 1 gång
- B. 2 gånger
- C. 3 gånger
- D. 4 gånger
- E. 5 gånger

```
i = 1  
j = 4  
  
while i < 4 and j < 8:  
    print(i, ", ", j)  
    i += 1  
    j += 1
```

8. Hur många kombinationer av tilldelningar av True/False för variablerna `x` och `y` finns det som gör uttrycket `(x and y) or (not x and not y)` sant?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

Del B: kodfrågor

Var snäll använd ett papper till varje fråga i del B. Delfrågor får gärna lösas på samma papper.

9.

A. Skriv en funktion `sum_keys(d)` som tar in en uppslagstabell `d` där alla *nycklar* är tal och returnerar summan av nycklarna. (1p)

Exempelanvändning:

```
[In: ] ex_dict1 = { 1 : 2 , 3 : 4, 5 : 6 }  
[In: ] ex_dict2 = { 2 : 0 , 7 : 5, 8 : 3 }  
[In: ] print(sum_keys(ex_dict1))  
[Out:] 9  
[In: ] print(sum_keys(ex_dict2))  
[Out:] 17
```

B. Skriv en funktion `max_value(d)` som tar in en uppslagstabell `d` där alla *värden* är tal och returnerar det största av dem. (1p)

Exempelanvändning (med samma `ex_dict1` och `ex_dict2` som ovan):

```
[In: ] print(max_value(ex_dict1))
[Out:] 6
[In: ] print(max_value(ex_dict2))
[Out:] 5
```

10. När man läser en text spelar det ofta inte så stor roll om bokstäver i mitten av ord är i fel ordning. Exempelvis ser man lätt att "progammreing" egentligen borde vara "programmering". Det är i regel lättare att läsa ord där bokstäver är omkastade så länge bokstäverna i början och slutet är rätt. Ni ska här skriva en funktion som kastar om bokstäver i mitten av strängar, men behåller ett antal bokstäver i början och slutet i rätt ordning.

- A. Skriv först en funktion `shuffle_letters(s)` som kastar om bokstäverna i strängen `s`. (2p)

Algoritmidé:

1. Skapa en tom utdatasträng `out`.
2. Generera ett slumpmässigt tal `x` mellan 0 och `len(s)-1`.
3. Plocka ut och ta bort bokstaven på position `x` i `s` och lägg den till `out`.
4. Fortsätt på samma sätt tills du plockat ut alla bokstäver i `s` och lagt dem till `out`.
5. Returnera `out`.

Tips: för att få ett slumpmässigt tal `x` mellan 0 och `len(s)-1` kan man skriva `x = randrange(len(s))`.

Obs: man får inte använda någon annan funktion från `random` biblioteket än `randrange`.

Exempelanvändning:

```
[In: ] print(shuffle_letters("programmering"))
[Out:] grpiaroemnrg
```

- B. Skriv en funktion `shuffle_letters_in_middle(s,n)` som tar in en sträng `s` och ett tal `n` och kastar om bokstäverna `n` steg från början och slutet av strängen. De `n` första och sista bokstäverna ska alltså inte kastas om medan de i mitten ska det. Man kan anta att `len(s) > 2 * n` så att det är några bokstäver som ska kastas om. (1p)

Exempelanvändning:

```
[In: ] print(shuffle_letters_in_middle("programmering",3))
[Out:] proagmerrmring
[In: ] print(shuffle_letters_in_middle("programmering",4))
[Out:] progmmaerring
[In: ] print(shuffle_letters_in_middle("programmering",5))
[Out:] progrmamering
```

11.

- A. Skriv en högre ordningens funktion `iterate(f,x,n)` som returnerar en lista av längd `n` där elementet på index `i` är `f` applicerad `i` gånger på `x`. Listan ska alltså ha formen `[x, f(x), f(f(x)), ...]` där det sista elementet är `f` applicerad `n-1` gånger på `x`. (2p)

Exempelanvändning:

```
[In: ] def double(x): return 2 * x
[In: ] print(iterate(double,1,10))
[Out:] [1, 2, 4, 8, 16, 32, 64, 128, 256, 512]
```

- B. Istället för att ge `double` ett namn ovan skulle man kunna använda sig av en anonym funktion (d.v.s. ett lambda-uttryck) istället. Vad ska man byta ut ??? mot i koden nedan för att uppnå detta? (1p)

```
[In: ] print(iterate(???,1,10))
[Out:] [1, 2, 4, 8, 16, 32, 64, 128, 256, 512]
```

12. Du har nyligen fått jobb på ett svenskt IT-bolag inom musikbranschen och din första uppgift på ditt nya jobb är att skriva klasser för att representera delar av deras musikkatalog.

- A. Skriv en klass `Song` som representerar en låt. Klassen ska ha en konstruktor som låter användaren sätta instansattributen `artist`, `name`, och `length`. Artistens namn (`artist`) och låtnamn (`name`), representeras m.h.a. strängar. Låtens längd (`length`) representeras som ett par (m, s) där m och s är låtens längd i minuter och sekunder, så $(4, 59)$ är en 4 minuter och 59 sekunder lång låt. (1p)

Klassen ska även ha en `__str__` metod som returnerar en sträng på formen `artist - name` (se nedan för exempel på hur det ska se ut). (1p)

Exempelanvändning:

```
[In: ] s1 = Song("Bruce Springsteen", "The Ties That Bind", (3, 33))
[In: ] s2 = Song("Bruce Springsteen", "Hungry Heart", (3, 19))
[In: ] s3 = Song("Bruce Springsteen", "The River", (4, 59))
[In: ] print(s1)
[Out:] Bruce Springsteen - The Ties That Bind
[In: ] print(s2)
[Out:] Bruce Springsteen - Hungry Heart
[In: ] print(s3.length)
[Out:] (4, 59)
```

- B. Skriv en klass `Album` som representerar ett album. Konstruktorn ska låta användaren sätta instansattributen `artist`, `name`, och `songs` vilka innehåller artistens namn (`artist`), albumets namn (`name`) och en lista `songs` med låtarna på albumet (d.v.s. en lista med element av typ `Song`). Konstruktorn ska även kontrollera att `artist` matchar artistnamnet på alla låtar som man lägger till och om detta inte är fallet ska ett `ValueError` lyftas (för enkelhets skull kan man anta att det bara är en artist på skivan, så inga samarbeten med andra artister). (2p)

Exempelanvändning:

```
[In: ] a = Album("Bruce Springsteen", "The River", [s1, s2, s3])
[In: ] print(a.artist)
[Out:] Bruce Springsteen
[In: ] b = Album("Johnny Cash", "I Walk The Line", [s1, s2, s3])
[Out:] Traceback (most recent call last):
...
ValueError: All songs must have the same artist as the artist on the album
```

- The exam contains multiple-choice questions where at least one option is correct. If you answer incorrectly or do not select the exact number of correct options, you get zero points for that question.
- You must pass part A (4 correct answers out of 8) for part B to be graded.
- Part B consists of questions with varying points (a total of 12 points).
- No import (Python's standard library or external libraries) may be used unless specified or included in the task. You may use built-in functions like len, range, and map.
- All code refers to **Python 3**, i.e., *not* e.g., Python 2.7
- **Help:** One A4 sheet with as much information as you wish. You may write on both sides.
- **Grading:** E: 10, D: 12, C: 14, B: 16, A: 18, out of a total of 20.

Part A: Multiple choice questions

Please collect the answers to part A on one answer sheet.

1. What does `s` contain after running the code snippet on the right?

- A. ''
- B. 'tillLycka'
- C. 'Lyckatill'
- D. 'akcyLllit'
- E. 'llitakcyL'

```
s = ''  
  
for x in ['Lycka', 'till']:  
    for y in x:  
        s = y + s
```

2. Which of the following keywords in Python are used for error handling?

- A. raise
- B. yield
- C. try
- D. except
- E. break

3. Given `xs = [0, 1, 2, 3, 4, 5]`, which of the following **print** expressions will output the list `[1, 3, 5]`?

- A. `print([x for x in xs if x % 2 != 0])`
- B. `print(xs[1][3][5])`
- C. `print(xs[0:1:3])`
- D. `print([xs[1], xs[3], xs[5]])`
- E. `print([xs[1]] + [xs[3]] + [xs[5]])`

4. What is printed by the code to the right?

- A. 3
- B. 4
- C. 5
- D. 6
- E. Nothing, as you get an exception as you cannot write `x += x`

```
x = 3  
  
def f(x, y=2):  
    x += x  
    print(x + y)  
  
f(1)
```

5. If `d = ('e', [4, 'e'], 'exam', { 'a': 1, 2 : 'e' })`, which of the following options can be used get an 'e'?

- A. `d[0]`
- B. `d[1][2]`
- C. `d[2][1]`
- D. `d[3]['e']`
- E. `d[3][2]`

6. What is the result of the code to the right?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

```
def f(x):  
    if x == 0:  
        return x  
    elif x % 2 == 0:  
        return x + f(x-1)  
    else:  
        return x - f(x-1)  
  
print(f(5))
```

7. How many times is `print` called when the code to the right is run?

- A. 1 time
- B. 2 times
- C. 3 times
- D. 4 times
- E. 5 times

```
i = 1  
j = 4  
  
while i < 4 and j < 8:  
    print(i, ", ", j)  
    i += 1  
    j += 1
```

8. How many combinations of assignments of True/False to the variables `x` and `y` make the expression `(x and y) or (not x and not y)` true?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

Part B: Code questions

Please use one piece of paper for each question in Part B. Subquestions can be solved on the same paper.

9.

A. Write a function `sum_keys(d)` that takes a dictionary `d` where all *keys* are numbers and returns the sum of the keys. (1p)

Example:

```
[In: ] ex_dict1 = { 1 : 2 , 3 : 4, 5 : 6 }  
[In: ] ex_dict2 = { 2 : 0 , 7 : 5, 8 : 3 }  
[In: ] print(sum_keys(ex_dict1))  
[Out:] 9  
[In: ] print(sum_keys(ex_dict2))  
[Out:] 17
```

B. Write a function `max_value(d)` that takes a dictionary `d` where all *values* are numbers and returns the largest of them. (1p)

Example (using the same `ex_dict1` and `ex_dict2` as above):

```
[In: ] print(max_value(ex_dict1))
[Out:] 6
[In: ] print(max_value(ex_dict2))
[Out:] 5
```

10. When reading a text, it often doesn't matter if letters in the middle of words are in the wrong order. For example, it's easy to see that "prorgammreing" should actually be "programming" ("programming" in Swedish). It is usually easier to read words where the letters are scrambled as long as the letters at the beginning and end are correct. You should write a function that scrambles the letters in the middle of strings, but keeps a certain number of letters at the beginning and end in the correct order.

- A. First, write a function `shuffle_letters(s)` that shuffles the letters in the string `s`. (2p)

Algorithm idea:

1. Create an empty output string `out`.
2. Generate a random number `x` between 0 and `len(s)-1`.
3. Extract and remove the letter at position `x` in `s` and add it to `out`.
4. Continue in the same way until you have extracted all the letters in `s` and added them to `out`.
5. Return `out`.

Tip: To get a random number `x` between 0 and `len(s)-1`, you can write `x = randrange(len(s))`.

Note: you may not use any other function from the `random` library than `randrange`.

Example:

```
[In: ] print(shuffle_letters("programming"))
[Out:] gpiaroemmnrg
```

- B. Write a function `shuffle_letters_in_middle(s,n)` that takes a string `s` and a number `n` and scrambles the letters `n` steps from the beginning and end of the string. The `n` first and last letters should hence not be scrambled while the ones in the middle should be. You can assume that `len(s) > 2 * n` so that there are some letters to scramble. (1p)

Example:

```
[In: ] print(shuffle_letters_in_middle("programming",3))
[Out:] proagmermring
[In: ] print(shuffle_letters_in_middle("programming",4))
[Out:] progmmaerring
[In: ] print(shuffle_letters_in_middle("programming",5))
[Out:] progrmaming
```

11.

- A. Write a higher-order function `iterate(f,x,n)` that returns a list of length `n` where the element at index `i` is `f` applied `i` times to `x`. The list should thus have the form `[x, f(x), f(f(x)), ...]` where the last element is `f` applied `n-1` times to `x`. (2p)

Example:

```
[In: ] def double(x): return 2 * x
[In: ] print(iterate(double,1,10))
[Out:] [1, 2, 4, 8, 16, 32, 64, 128, 256, 512]
```

- B. Instead of giving `double` a name as above, one could use an anonymous function (i.e. a lambda expression) instead. What should you replace `???` with in the code below to achieve this? (1p)

```
[In: ] print(iterate(???,1,10))
[Out:] [1, 2, 4, 8, 16, 32, 64, 128, 256, 512]
```

12. You have recently started working at a Swedish IT company in the music industry, and your first task is to write classes to represent parts of their music catalog.

- A. Write a class `Song` that represents a song. The class should have a constructor that allows the user to set the instance attributes `artist`, `name`, and `length`. The artist's name (`artist`) and song name (`name`) are represented as strings. The song length (`length`) is represented as a tuple `(m, s)` where `m` and `s` are the song's length in minutes and seconds, so `(4, 59)` is a song that is 4 minutes and 59 seconds long. (1p)

The class should also have a `__str__` method that returns a string of the form `artist - name` (see below for an example of what this should look like). (1p)

Example:

```
[In: ] s1 = Song("Bruce Springsteen","The Ties That Bind", (3,33))
[In: ] s2 = Song("Bruce Springsteen","Hungry Heart", (3,19))
[In: ] s3 = Song("Bruce Springsteen","The River", (4,59))
[In: ] print(s1)
[Out:] Bruce Springsteen - The Ties That Bind
[In: ] print(s2)
[Out:] Bruce Springsteen - Hungry Heart
[In: ] print(s3.length)
[Out:] (4, 59)
```

- B. Write a class `Album` that represents an album. The constructor should allow the user to set the instance attributes `artist`, `name`, and `songs`, which contain the artist's name (`artist`), the album's name (`name`), and a list of the songs on the album (`songs`) which is a list of `Song` objects. The constructor should also check that the `artist` matches the artist name on all songs added, and if this is not the case, a `ValueError` should be raised (for simplicity, you can assume there is only one artist on the album, so no collaborations with other artists). (2p)

Example:

```
[In: ] a = Album("Bruce Springsteen","The River",[s1,s2,s3])
[In: ] print(a.artist)
[Out:] Bruce Springsteen
[In: ] b = Album("Johnny Cash","I Walk The Line",[s1,s2,s3])
[Out:] Traceback (most recent call last):
      ...
ValueError: All songs must have the same artist as the artist on the album
```