

Please note: this exam contains the questions in Swedish *and* English!

- Inga hjälpmedel tillåtna.
- **Skriv tydligt.** Svårlästa svar riskerar 0 poäng.
- Skriv bara på en sida av varje papper!
- Motivera alla svar (om inte annat anges)!
- Man måste bli godkänd på del A (5 rätt på 10 frågor) för att del B ska rättas.
- **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 svarspapper.

1. Vilket av dessa alternativ är *inte* en typ i Python?

- A. bit
- B. bool
- C. float
- D. int
- E. str

2. Vilket påstående är falskt?

- A. Filer måste öppnas för att kunna läsas.
- B. Man måste läsa *hela* filen innan den stängs.
- C. Det blir ett särfall (*exception*) om man försöker läsa en rad för mycket.
- D. Du kan ha två filer öppna samtidigt.
- E. Det är två olika anrop till **open** för att läsa och skriva.

3. Vad blir effekten av koden till höger?

- A. Det blir ett särfall.
- B. Värdet `None` skrivs ut.
- C. Tomma strängen skrivs ut.
- D. Värdet `7.5` skrivs ut.
- E. Värdet `'DA2005'` skrivs ut.

```
a = {'DA2005': 7.5, 'MM2001': 30}
b = 'ProgTek'
print(a[b])
```

4. Vad blir skrivs ut av koden till höger?

- A. Inget, det blir ett särfall.
- B. `[False, False, False, True]`
- C. `'Silverstreckad pärlemorfjäril'`
- D. `['Silverstreckad pärlemorfjäril']`
- E. `[6, 10, 8, 29]`

```
a = lambda x: len(x) > 10
b = ['Amiral', 'Sorgmantel', 'Blåvinge',
     'Silverstreckad pärlemorfjäril']
print(list(map(a, b)))
```

5. När ska man använda **raise**? (Ett svar.)

- A. Aldrig, det är en intern Python-instruktion.
- B. När man vill fånga sårfall.
- C. När man vill signalera att ett fel har uppstått.
- D. När man vill skriva ut ett delresultat.
- E. När man vill avsluta programmet.

6. Hur många punkter skrivs ut om man definierar funktionen till höger och sen gör anropet $f(4)$?

- A. 4
- B. 6
- C. 8
- D. 9
- E. 16

```
def f(n):  
    for i in range(n):  
        for j in range(i):  
            print('.')
```

7. Vilket påstående är sant om strängar?

- A. Man kan inte använda grekiska bokstäver i dem.
- B. Man kan inte ändra på en sträng, bara skapa nya strängar.
- C. Man kan inte ha radbrytningar i strängar.
- D. Man kan inte komma åt enskilda bokstäver i strängar.
- E. Man kan inte avgöra om två strängar är likadana.

8. Vad returnerar anropet $f(2)$ om koden till höger är definierad?

- A. 2
- B. 3
- C. 4
- D. 5
- E. 6

```
a = 1  
def f(x):  
    a = x  
    return a + g(a)  
  
def g(x):  
    return a + x
```

9. Vad returnerar anropet $h([1, 2, 3])$ givet funktionen till höger?

- A. $[3, 2, 1]$
- B. $[1, 2, 3]$
- C. $[(1, 2), (1, 3), (2, 3)]$
- D. $[(2, 1), (3, 1), (3, 2)]$
- E. $[(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)]$

```
def h(lst):  
    return [(a, b) for a in lst  
            for b in lst if a < b]
```

10. Vad returnerar anropet $f(123, 0)$ om f är definierad som till höger?

- A. 1
- B. 6
- C. 12
- D. 14
- E. 36

```
def f(x, y):  
    if x == 0:  
        return y  
    else:  
        a = x // 10 # Kvot  
        b = x % 10 # Rest  
        return f(a, y + b**2)
```

Del B: kodfrågor

Var snäll använd ett papper till varje fråga i del B.

11. Förklara varför uttryck som

```
if x + y == 0.3:  
    ...
```

är problematiska. (1p)

12. Funktionen `interact(menu, valid_choices)` ska presentera en menytext och sedan fråga användaren efter ett val. Giltiga svar finns i listan `valid_choices` och om man inte ger ett giltigt svar ska funktionen fortsätta fråga efter ett giltigt menyval. Om det är ett giltigt svar så ska det returneras.

A. Det finns ett fel i koden nedan. Vad är felet och hur rättar man till det? (1p)

- B. Man kan avbryta en körning av `interact` genom att trycka CTRL-c (control-tangenten och c) eller CTRL-d. CTRL-c avbryter körningen och CTRL-d skickar "end of file" till programmet, vilket betyder "nu kommer det inte mer data". Resultatet för ett Python-program blir att särfallen `KeyboardInterrupt` respektive `EOFError` uppstår. Skriv om `interact` till att fånga dessa särfall och skriva ut "Bye!" samt avsluta. (1p)

```
def interact(menu, valid_choices):  
    '''  
    Get an answer from a user.  
    menu - a string presenting the choices.  
    valid_choices - a list of strings containing the alternatives  
                  for the user.  
    '''  
    while True:  
        print(menu)  
        answer = input('What is your choice? ')  
        if answer in valid_choices:  
            return  
        else:  
            print('That is not a valid answer. Try again.')
```

13. Funktionen `rec_fcn(lst, x)`, nedan, gör en rekursiv beräkning på listan `lst`. Förklara vad funktionen beräknar och skriv om den till att *inte* använda rekursion (men göra samma beräkning). (2p)

```
def rec_fcn(lst, x):  
    if lst == []:  
        return 0  
    elif lst[0] == x:  
        return 1 + rec_fcn(lst[1:], x)  
    else:  
        return rec_fcn(lst[1:], x)
```

14. I den här uppgiften tänker vi oss att vi har data i en uppslagstabell, med fjärilsnamn associerade med par, där parets första element anger om fjärilen övervintrar i Sverige och dess andra element är fjärilens skönhet givet på skalan 0 till 10. Till exempel:

```
butterflies = {  
    'Sorgmantel': (True, 10),  
    'Nässelfjäril': (True, 10),  
    'Grönsnabbvinge': (False, 9),  
    'Amiral': (False, 10),  
    'Citronfjäril': (True, 9),  
}
```

Här kan man utläsa att sorgmantel övervintrar och har perfekt skönhet, samt att grönsnabbvinge inte övervintrar och har skönhetsvärde 9.

- A. Skriv funktionen `beauty_stats(data)` som beräknar medelskönheten för övervintrande respektive icke-övervintrande fjärilar och skriver ut resultatet. (2p)

Exempelkörning:

```
[In: ] beauty_stats(butterflies)
[Out:]
Hibernating Mean beauty
True          9.666666666666666
False         9.5
```

Det är inte viktigt att presentationen har raka marginaler, men det ska vara en liten tabell.

- B. Skriv funktionen `keep_perfect(data)` för att konvertera en uppslagstabell på ovanstående format till en lista med par (med tupler), där första elementet är fjärlens namn och det andra elementet anger om fjärlen övervintrar eller inte. För full poäng (2p) *måste* du använda funktionell programmering: for- och while-loopar är inte accepterade i lösningen. Observera att resultatet ska returneras, inte skrivas ut. (2p)

Exempelkörning:

```
[In: ] keep_perfect(butterflies)
[Out:] [('Sorgmantel', True), ('Nässelfjäril', True), ('Amiral', False)]
```

- C. Skriv en klass `Butterfly` som har tre attribut som initieras med tre parametrar vid instantiering: artnamn (som "Citronfjäril"), dess skönhetsvärde (exempelvis 9), och ett boolskt värde som är sant om och endast om arten övervintrar. Visa hur ett man instantierar ett objekt för citronfjäril. (1p)

English translation

- This exam has multiple choice questions where at least one answer is correct. If your answer is incorrect or you do not include all correct answers, you will receive 0 points on that question.
 - **Write clearly.** Answers that are difficult to read may receive 0 points.
 - Write only on one side of each paper!
 - You must pass part A (5 correct out of 10 questions) to have your part B graded.
 - **Aids:** An A4 with as much information as you want. You can write on both sides.
 - **Grade thresholds:** E: 10, D: 12, C: 14, B: 16, A: 18, of maximum 20.
-

Part A: Multiple choice

Please collect your answers to part A on a single piece of paper.

1. Which of these alternatives is *not* a type in Python?

- A. bit
- B. bool
- C. float
- D. int
- E. str

2. Which claim is false?

- A. Files must be opened to be read.
- B. You have to read the *whole* file before closing it.
- C. There is an exception if you try to read one line too much.
- D. You can have two files opened at the same time.
- E. There are different calls to **open** for reading and writing.

3. What is the effect of the code on the right?

- A. There will be an exception.
- B. The value None is printed.
- C. The empty string is printed.
- D. The value 7.5 is printed.
- E. The value 'DA2005' is printed.

```
a = {'DA2005': 7.5, 'MM2001': 30}
b = 'ProgTek'
print(a[b])
```

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

- A. Inget, det blir ett sårfall.
- B. [False, False, False, True]
- C. 'Silverstreckad pärlemorfjäril'
- D. ['Silverstreckad pärlemorfjäril']
- E. [6, 10, 8, 29]

```
a = lambda x: len(x) > 10
b = ['Amiral', 'Sorgmantel', 'Blåvinge',
     'Silverstreckad pärlemorfjäril']
print(list(map(a, b)))
```

5. When are you supposed to use **raise**? (One correct answer.)

- A. Never, it is an internal Python instruction.
- B. When catching exceptions.
- C. When signalling that an error has occurred.
- D. When you want to print a intermediate result.
- E. When terminating the program.

6. How many points (periods) are printed if you define the code on the right and call `f(4)`?

- A. 4
- B. 6
- C. 8
- D. 9
- E. 16

```
def f(n):  
    for i in range(n):  
        for j in range(i):  
            print('.')
```

7. Which claim about strings is true?

- A. You cannot use Greek letters in them.
- B. You cannot change a string, only create new ones.
- C. You cannot have newlines in strings.
- D. You cannot access single characters in strings.
- E. You cannot determine whether two strings are equal.

8. What does the call `f(2)` return if the code on the right is defined?

- A. 2
- B. 3
- C. 4
- D. 5
- E. 6

```
a = 1  
def f(x):  
    a = x  
    return a + g(a)  
  
def g(x):  
    return a + x
```

9. What does the call to `h([1, 2, 3])` return given the function on the right?

- A. `[3, 2, 1]`
- B. `[1, 2, 3]`
- C. `[(1, 2), (1, 3), (2, 3)]`
- D. `[(2, 1), (3, 1), (3, 2)]`
- E. `[(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)]`

```
def h(lst):  
    return [(a, b) for a in lst  
            for b in lst if a < b]
```

10. What is returned by `f(123, 0)` if `f` is defined as on the right?

- A. 1
- B. 6
- C. 12
- D. 14
- E. 36

```
def f(x, y):  
    if x == 0:  
        return y  
    else:  
        a = x // 10 # Kvot  
        b = x % 10  # Rest  
        return f(a, y + b**2)
```

Part B: Coding questions

Please use a separate piece of paper (or several) for each question in part B.

11. Explain why expressions like

```
if x + y == 0.3:
    ...
```

are problematic.

(1p)

12. The function `interact(menu, valid_choices)` is supposed to present a menu and then ask the user for a valid menu choice. Valid answers are given in the list `valid_choices` and the function should ask again if an invalid answer is given. A valid answer should be returned.

- A. There is an error in the code below. What is the error and how to you correct it? (1p)

- B. You can interrupt a call to `interact` by pressing CTRL-c (the control key and c) or CTRL-d. CTRL-c aborts the execution and CTRL-d sends "end of file" to the program, which means "there is no more data". For a Python program, this results in the exceptions `KeyboardInterrupt` and `EOFError`, respectively. Rewrite `interact` to catch these exceptions, with the result that "Bye!" is printed, and the program quits. (1p)

```
def interact(menu, valid_choices):
    '''
    Get an answer from a user.
    menu - a string presenting the choices.
    valid_choices - a list of strings containing the alternatives
                  for the user.
    '''
    while True:
        print(menu)
        answer = input('What is your choice? ')
        if answer in valid_choices:
            return
        else:
            print('That is not a valid answer. Try again.')
```

13. The function `rec_fcn(lst, x)`, below, makes a recursive computation on the list `lst`. Explain what the function computes and rewrite it to *not* use recursion (but making the same computation). (2p)

```
def rec_fcn(lst, x):
    if lst == []:
        return 0
    elif lst[0] == x:
        return 1 + rec_fcn(lst[1:], x)
    else:
        return rec_fcn(lst[1:], x)
```

14. In this assignment we assume data is given in a dictionary, with Swedish butterfly names associated with pairs, where the first element of the pair indicates whether the butterfly hibernates in Sweden, and its second element rates the beauty of the butterfly on a scale from 0 to 10. For example:

```
butterflies = {
    'Sorgmantel': (True, 10),
    'Nässelfjäril': (True, 10),
    'Grönsnabbvinge': (False, 9),
    'Amiral': (False, 10),
    'Citronfjäril': (True, 9),
}
```

Here you can read that the *sorgmantel* hibernates and has perfect beauty, and *grönsnabbvinge* does not hibernate and has beauty 9.

- A. Write the function `beauty_stats(data)` that computes the average beauty for hibernating and non-hibernating butterflies, respectively, and prints the result. (2p)

Sample execution:

```
[In: ] beauty_stats(butterflies)
[Out:]
Hibernating Mean beauty
True         9.666666666666666
False        9.5
```

It is not important with straight margins, but there should be a small table.

- B. Write the function `keep_perfect(data)` to convert a dictionary in the format above to a list of pairs (tuples), where the first element is the species name and the second element indicates whether the butterfly hibernates or not. For full points (2p) you *have* to use functional programming: for and while loops are not accepted in the solution.

Please note that the result should be returned, not printed. (2p)

Sample execution:

```
[In: ] keep_perfect(butterflies)
[Out:] [('Sorgmantel', True), ('Nässelfjäril', True), ('Amiral', False)]
```

- C. Write a class `Butterfly` that has three attributes that are initialized at instantiation: species name (like "Citronfjäril"), its beauty (for example 9), and a boolean value for indicating whether the species hibernates. Show how to instantiate an object for *Citronfjäril*. (1p)