

# Tentamen Programmeren IK

donderdag 22 oktober 12:15—13:30

Schrijf je naam op de regel hieronder.

---

Sla deze pagina niet om tot de surveillant vertelt dat het tentamen begonnen is.

~

Leg je collegekaart of identiteitsbewijs klaar op je tafel.

~

Dit tentamen is “gesloten boek.” Je mag echter tijdens het tentamen één tweezijdig beschreven of bedrukt A4'tje gebruiken. Daarnaast mag je een pen of potlood gebruiken, maar verder niets.

~

Je moet je antwoorden invullen in dit document. Scheur de bladzijden niet af. De opgaven worden t.z.t. gepubliceerd op de website van het vak.

~

Kladpapier kun je vinden op de achterkant van dit boekje. Tenzij anders vermeld mag je alle functies gebruiken die in opgaven en leeswerk van de cursus zijn behandeld. Je hoeft bij code geen commentaar te schrijven, maar het kan helpen in het geval je programma niet correct blijkt te zijn. Als je te weinig tijd hebt kun je uitwijken naar het schrijven van pseudocode om een deel van de punten te halen.

SUCCES :-)



### Multiple Choice.

For each of the following questions or statements, circle the letter (a, b, c or d) of the one response that best answers the question or completes the statement; you need not explain your answers.

1. (1 point.) If a dependent child is a person under 18 years of age who does not earn \$10,000 or more a year, which expression would define a dependent child?

a. `(age < 18) and (salary < 10000)`  
b. `(age < 18) or (salary < 10000)`  
c. `(age <= 18) and (salary <= 10000)`  
d. `(age <= 18) or (salary <= 10000)`

2. (1 point.) What are the values of `girls`, `boys`, and `children` after the following code has been executed?

```
girls = 0
boys = 0
children = girls + boys
girls = 15
boys = 12
```

a. 0, 0, 0  
b. 0, 0, 27  
c. 15, 12, 0  
d. 15, 12, 27

3. (2 points.) Consider the following block of code, where variables `a`, `b`, `c`, and `answer` each store integer values:

```
if a > b:
    if b > c:
        answer = c
    else:
        answer = b
elif a > c:
    answer = c
else:
    answer = a
```

Which of the following sets of values for `a`, `b`, and `c` will cause `answer` to be assigned the value in variable `b`?

a. `a = 1, b = 2, c = 3`  
b. `a = 1, b = 3, c = 2`  
c. `a = 2, b = 1, c = 3`  
d. `a = 3, b = 2, c = 1`

### Looping back to Scratch.

4. (4 points.) Consider the Scratch script below.



In the space below, complete the translation of this Scratch script to a Python program in such a way that its output is equivalent. (Your program's structure needn't be equivalent.) Assume that **say** is `print`, and **change** means to add or subtract.

```
counter = 0
```

5. (4 points.) Consider the Scratch script below.



In the space below, complete the translation of this Scratch script to a Python program with two functions, `main` and `yell`, in such a way that its output is equivalent; it needn't be structurally the same. Assume that Scratch's **say** block translates to `print`.

```
def yell():
```

```
def main():
```

```
if __name__ == '__main__':  
    main()
```

### Real problems.

6. (2 points.) Consider the program below.

```
cents = 50
dollars = cents / 100
print dollars
```

When executed, this program prints

0

which is not how much money we have! In no more than three sentences, explain why this program thinks that 50 cents divided by 100 is something other than 0.5.

### Quick statistics.

7. (2 points.) What will be the value of `result` after the following code statements are executed?

```
nums1 = [1, -5, 2, 0, 4, 2, -3]
nums2 = [1, -5, 2, 4, 4, 2, 7]
result = 0
j = 0
while j < len(nums1):
    if nums1[j] != nums2[j]:
        result = result + 1
    j = j + 1
```

## Real dictionaries.

8. (4 points.) Say you're given the following variable to help you write a word game. Each element in the dictionary associates one letter of the (lowercase) alphabet with a score.

```
SCRABBLE_LETTER_VALUES = { 'a': 1, 'b': 3, 'c': 3, 'd': 2, 'e': 1,
    'f': 4, 'g': 2, 'h': 4, 'i': 1, 'j': 8, 'k': 5, 'l': 1, 'm': 3,
    'n': 1, 'o': 1, 'p': 3, 'q': 10, 'r': 1, 's': 1, 't': 1, 'u': 1,
    'v': 4, 'w': 4, 'x': 8, 'y': 4, 'z': 10 }
```

Now, write a function `word_score` that computes the total value of a word, given the scores for each letter. It should take a string `word` as input:

```
def word_score(word):
```

## Program Analysis.

9. (1 points.) What is the outcome or likely purpose of the following piece of code?

```
result = 0
for j in range(0, len(number)):
    if number[j] < 0:
        result = result + 1
```

- a. to find the smallest number in the list
- b. to count the negative numbers in the list
- c. to sum the negative numbers in the list
- d. to add 1 to each of the negative numbers in the list
- e. to find the index of the first negative number in the list

10. (2 points.) In a short sentence, not unlike the answers in question 9, describe the purpose of the following piece of code.

```
list1 = ['a','b','c','d','e','f']
list2 = list1[:]
length = len(list1)
for i in range(0, length ):
    print list1[i]
    list2[ length -1 -i ] = list1[i]
```

11. (2 points.) The following function has a big problem. It never stops and never gives an answer!

```
def factorial(n):
    answer = 1
    while n > 0:
        answer = answer * n
    return answer
```

Explain in no more than three sentences why the program never stops.



### Program transformation.

12. (4 points.) The following program returns a new list in which each element is double the value of its corresponding element in the input list, `aList`.

```
def listDoubler(aList):  
    doubledList = []  
    for elem in aList:  
        # append the value 2*elem to doubledList  
        doubledList.append(2*elem)  
    return doubledList  
print (listDoubler([20, 21, 22]))
```

Now, write a function that has the same input and output, but uses a list comprehension instead of a for-loop.

```
def listDoublerComprehension(aList):
```

**Scrap Paper.**

*Nothing on this page will be examined by the staff unless otherwise directed.*