

# University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 4 MODULE, AUTUMN SEMESTER 2020-2021

Programming

Time allowed: 3 Hours

---

## *Open-book examination*

Answer ALL FOUR questions

*This open-book examination will be marked out of 100.*

You may write/draw by hand your answers on paper and then scan them to a PDF file, or you may type/draw your answers into electronic form directly and generate a PDF file. Guidance on scanning can be found through the [Faculty of Science Moodle Page Guidance for Remote Learning](#).

Submit your answers containing all the work you wish to have marked as a single PDF file, with each page in the correct orientation, to the appropriate dropbox on the module's Moodle page.

Use the standard naming convention for your document:

StudentID COMP4008. Write your student ID number at the top of each page of your answers. Do not include your name.

Although you may use any notes or resources you wish to help you complete this open-book examination, the academic misconduct policy that applies to your coursework also applies here. You must be careful to avoid plagiarism, collusion or false authorship. You must produce the answers by yourself only. Please familiarise yourself with the [Faculty of Science Statement on Academic Integrity](#). This statement refers to, and does not replace, the University policy which stipulates severe penalties for academic misconduct. Please check the box indicated on Moodle to confirm that you have read this statement and that you understand it.

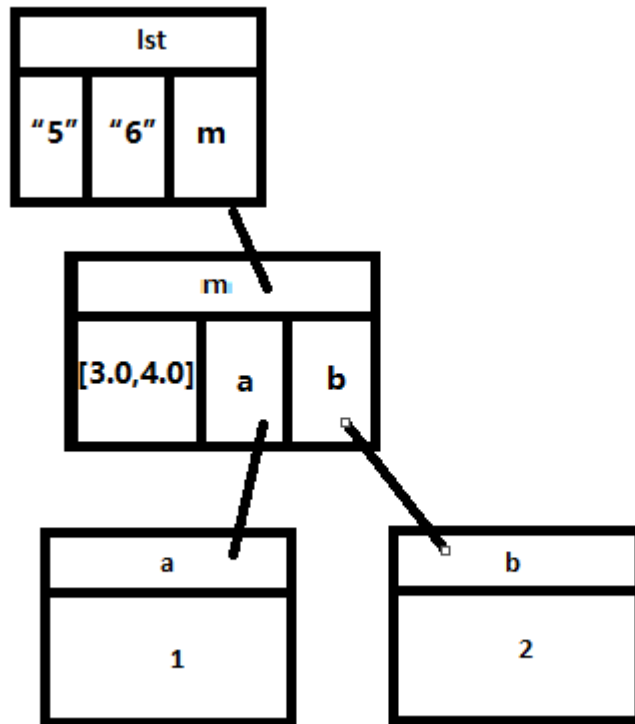
Staff are not permitted to answer assessment or teaching queries during the period in which your open-book examination is live. If you spot what you think may be an error on the exam paper, note this in your submission but answer the question as written.

---

**Question 1**

a.

(i)

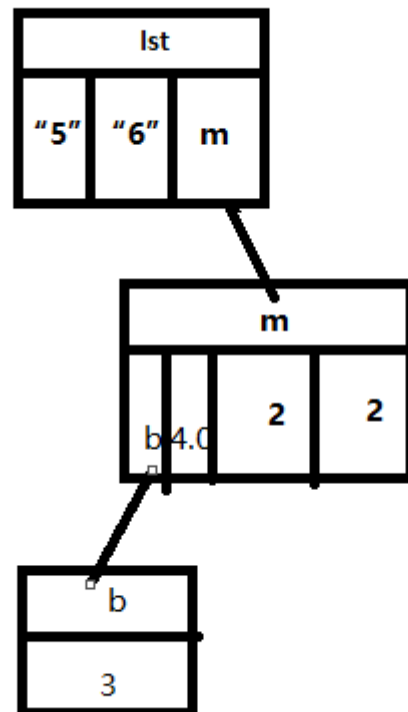


(ii)

```
a = 1
b = 2
m = [[3.0, 4.0], a, b]
lst = ["5", "6", m]

print(m[1:]+m[0]+lst[0:-1])
```

(iii)



iv.

```

8   a = 1
9   b = 2
10  m = [[3.0,4.0],a,b]
11  lst = ["5","6",m]
12
13  print(m[1:]+m[0]+lst[0:-1])
14  my_list = [m[0]+m[0]]+[m[0]+m[0]]+[m[0]+m[0]]

```

v.

```

8   m = [[3.0,4.0],2,2]
9   def foo(l):
10      m[:] = [m[1:]+m[0]]
11      return m
12   print(foo(m))
13

```

b.

```

8   def split_and_merge(a,b):
9       return a[:3]+b[:3]+a[3:]+b[3:]
10

```

Yes

Slicing can be used on lists.

c.

```

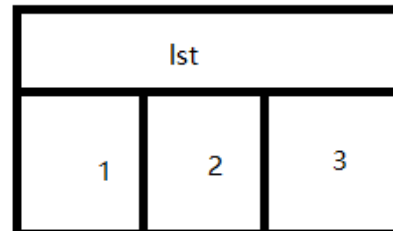
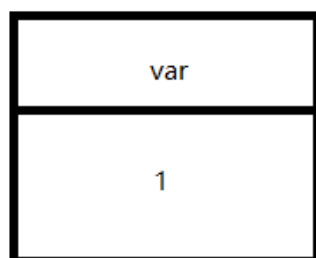
def split_and_merge(lista,word):
    word = list(word)
    lista[:] = lista[:3] + word[:3] + lista[3:] + word[3:]

```

yes,

The contents of a string based on the original address cannot be changed, however string can be sliced as same as list, because slicing could provide a new copy (new address).

d.



output:

3

[3,2,1]

1

[1,2,3]

The program does not modify the original variables, it used the local variables.

---

**Question 2**

a.

(i)

```
b = True
c = not b
d = b or c
if b and c or d:
    print("Option 1")
else:
    print("Option 2")
```

(ii)

```
try:
    x = "x"
    y = "a4"
    print(y)
    z = int(x)
    print("End")
except ValueError:
    print("Ups! there was an error")
```

(iii)

```
8     count = 0
9     while count<10:
10        if count%2==0 and count%4==0 :
11            print(".",end="")
12        else:
13            print("-",end="")
14        count +=1
```

(iv)

```
lst = [1,2,3,4,5,6,7,8,9,10]
for i in range(len(lst)-1,-1,-2):
    print(lst[i],end=",")
print()
for i in range(len(lst)-2,-1,-2):
    print(lst[i],end=",")
```

(v)

```
7
8   for i in range(0,5):
9       for j in range(0,5):
10          if j == i:
11              print("*",end=" ")
12          else:
13              print("_",end=" ")
14          print("\n")
```

b.

```
def sum_divisors(n):
    divisors = [1]
    for i in range(2, n):
        if (n % i) == 0:
            divisors += [i]
    return sum(divisors)

print(sum_divisors(220))
```

c.

```
def sum_divisors(n):
    divisors = [1]
    for i in range(2, n):
        if (n % i) == 0:
            divisors += [i]
    return sum(divisors)
print(sum_divisors(220))

def amigos(a,b):
    try:
        if sum_divisors(a) == b:
            return True
        else:
            return False
    except TypeError:
        print("Error: Wrong input")

print(amigos(220,284))
print(amigos(15,20))
print(amigos("Isaac","Thorsten"))
```

d.

```
29 def list_amigos(n,m):
30     for i in range(n,m):
31         if i == 1:
32             pass
33         else:
34             z = sum_divisors(i)
35             if amigos(z,i):
36                 print(i,z)
37
38 list_amigos(1,2000)
```

### Question 3

a.

```
def rev(l) :
    if len(l) == 0:
        return []
    return rev(l[1:])+[l[0]]
def fib(n) :
    if n < 2:
        return n
    return fib(n-1)+fib(n-2)

rev([1,2,3])

fib(5)
```

b.

```
8 def rev(l) :
9     if len(l) == 0:
10         return []
11     return rev(l[1:])+[l[0]]
12
13 def fib(n):
14     if n == 0:
15         return 1
16     elif n == 1:
17         return 1
18     else:
19         return fib(n-1)+fib(n-2)
20
21 rev([1,2,3])
22
23 fib(5)
24
```

c.

**Question 4:**

a.

(i) `fs.contents[0].name`

(ii) `fs.contents[1].contents[0].name`

(iii) `fs.contents[1].contents[0].contents[1].name = "dogs.jpg"`