

Question 1. [6 MARKS]

Complete the following function.

```
def find_partner(partners, student):
    '''Given partners (a list of lists) and student (a string containing a
    student's username), return the username of the student's partner, or
    the empty string if the student does not have a partner. Each element
    of partners is either a one-element list containing a username (a string),
    to represent a student without a partner, or a two-element list containing
    two usernames (both strings), for partners. The student will appear
    in partners exactly once.'''

    for pair in partners:
        if len(pair) == 2:
            if student == pair[0]:
                return pair[1]
            elif student == pair[1]:
                return pair[0]
        else:
            if student == pair[0]:
                return ''
```

Question 2. [8 MARKS]**Part (a)** [4 MARKS]

```
def mystery(a, b=0):
    i = len(a) - 1
    c = False

    while i >= 0 and not c:
        if a[i] == b:
            c = True
        else:
            i -= 1
    return i
```

Show variable values after each line has executed:				
a: [6, 1, 4, 1, 2]		b: 1		
i: 4				
c: False				
During 1st iteration	During 2nd iteration	...	During last iteration	
i: 4 c: False	i: 3 c: False		i: 3 c: True	
a[i]: 2 b: 1	a[i]: 1 b: 1		a[i]: b:	
c: (not reached)	c: True		c:	
i: 3	i: (not reached)		i:	
value returned: 3				

What is returned by the function call `mystery([6, 1, 4, 1, 2], 1)`?

In the table above, trace the variable values during execution of the function. For each blank in the table, fill in the value of the variable specified or write “not reached” if that line was not executed. You do **not** need to complete the section of the table labeled ‘...’. You only need to complete the section for the last iteration if the loop iterates 3 or more times.

Part (b) [4 MARKS]

```
def mystery(a, b=0):
    i = len(a) - 1
    c = False

    while i >= 0 and not c:
        if a[i] == b:
            c = True
        else:
            i -= 1
    return i
```

Show variable values after each line has executed:				
a: [3, 5]		b: 0		
i: 1				
c: False				
During 1st iteration	During 2nd iteration	...	During last iteration	
i: 1 c: False	i: 0 c: False		i: -1 c: False	
a[i]: 5 b: 0	a[i]: 3 b: 0		a[i]: b:	
c: (not reached)	c: (not reached)		c:	
i: 0	i: -1		i:	
value returned: -1				

Note: the code in Part(b) is the same as the code in Part(a).

What is returned by the function call `mystery([3, 5])`?

In the table above, trace the variable values during execution of the function. For each blank in the table, fill in the value of the variable specified or write “not reached” if that line was not executed. You do **not** need to complete the section of the table labeled ‘...’. You only need to complete the section for the last iteration if the loop iterates 3 or more times.

Question 3. [6 MARKS]

Complete the following function.

```
def extract_character(pixel_list):
    '''Given a list, pixel_list, containing 3 pixels, return the character
    hidden in the list as a string. One digit of the ASCII value representing
    the character is hidden in the one's place of the green component of each
    pixel in the list. The first pixel in the list contains the hundred's
    component of the number. The second pixel contains the ten's component,
    and the third contains the one's component.'''

    char = ""
    for pixel in pixel_list:
        char += str(get_green(pixel) % 10)
    return chr(int(char))
```