Question 1. [4 MARKS]

Dietitians consider a food to be *not a source* of iron if a serving of this food contains less than 0.7 mg of iron. If a serving of a food contains at least 3.5 mg of iron, then this food is considered to be a *very good source* of iron. Similarly, a *good source* of iron contains at least 2.1 mg of iron per serving (if it is not a very good source of iron). And finally, a food is considered to be a *source* of iron if it contains at least 0.7 mg of iron per serving (if is not a good or a very good source of iron).

Your task is to implement the following function:

```
def iron_source(amount: float) -> str:
    '''Return a classification of a food that contains amount mg of iron
    per serving, as an iron source.

    >>> iron_source(0.5)
    'not a source of iron'
    >>> iron_source(2.0)
    'source of iron'
    >>> iron_source(3.0)
    'good source of iron'
    >>> iron_source(3.7)
    'very good source of iron'
    '''

    desc = 'source of iron'
    if amount >= 3.5:
        return 'very good ' + desc
    elif amount >= 2.1:
        return 'good ' + desc
    elif amount >= 0.7:
        return desc
    else:
        return 'not a ' + desc
```

Question 2. [4 MARKS]

A *palindrome* is a word that reads the same forwards and backwards. For example, `civic` and `redder` are palindromes, while `csca08` is not.

Yesterday morning I implemented a function `reverse` in Python that takes a `str` and returns the reverse of it. Some example uses of my function:

```
    >>> reverse('')
    ''
    >>> reverse('x')
    'x'
    >>> reverse('csca08')
    '80acsc'
```

Implement the function below. You **must use the function reverse** in your solution.

```
def is_palindrome(text: str) -> bool:
    """Return True if and only if text is a palindrome.

    >>> is_palindrome('')
    True
    >>> is_palindrome('tattarrattat')
    True
    >>> is_palindrome('notatall')
    False

    """

      return text == reverse(text)
```

Question 3. [12 MARKS]

Implement the following functions.

```
def no_vowels(text: str) -> bool:
    """Return True if and only if text does not contain any vowels. Vowels
    are a, e, i, o, and u.

    >>> no_vowels('')
    True
    >>> no_vowels('thrrvowels')
    False
    """

    for char in text:
        if char in 'aeiou':
            return False
    return True
```

We will now consider a more general definition of a palindrome than the one we used Question 2. In this definition spaces, case, and punctuation are ignored. For example, `A Man, A Plan, A Canal: Panama!` is now considered a palindrome.

We will consider only the following characters as punctuation: `. , : ; ? !` as well as a space character.

Implement the function below. You **must use the function is_palindrome** in your solution, *even if you did not complete Question 2.*

```
def is_real_palindrome(text: str) -> bool:
    """Return True if and only if text is a palindrome, for a more general
    definition of a palindrome: case, whitespace, and punctuation are
    ignored.

    >>> is_real_palindrome('')
    True
    >>> is_real_palindrome('A Man, A Plan, A Canal: Panama!')
    True
    >>> is_real_palindrome('not at all')
    False

    """

    text = text.lower()            # ignore case

    filtered = ''
    for char in text:              # remove punctuation and spaces
        if char not in ' .,:;!?':
            filtered += char

    return is_palindrome(filtered)
```

Question 4. [5 MARKS]

Carefully examine the code below and complete the docstring for the mystery function `midterm`.

```
def midterm(message: str, x: str, y: str) -> str:

    """Return a copy of message, but with every occurrence of x replaced
    by y.

    Precondition: x is a single character string.

    >>> midterm('Anya Tafliovich', 'c', 'x')   # one occurrence of x
    'Anya Tafliovixh'
    >>> midterm('Kaveh Mahdaviani', 'a', 'o')  # several occurrences of x
    'Koveh Mohdovioni'
    >>> midterm('Anya Tafliovich', 't', 'e')   # no occurrences of x
    'Anya Tafliovich'
    """

    result = ''
    for ch in message:
        if ch == x:
            result = result + y
        else:
            result = result + ch
    return result
```

Question 5. [10 MARKS]

Use the bubble sheet on the next page to answer the multiple choice questions from the **extra sheet** you have received, not attached to this midterm.

Use only the first 10 lines on the bubble sheet. Everything else will be ignored when marking.

Note that the purpose of this question is NOT to test whether you remember built-in Python methods! Instead, we expect that you will use the "Built-in functions and methods" help sheet to answer the following questions.

You may use the rest of this page if you need more space for your solutions.