```
DEBUG_MODE = True

def read_text(prompt):
    '''
    Displays a prompt and reads in a string of text.
    Keyboard interrupts (CTRL+C) are ignored
    returns a string containing the string input by the user
    '''
    while True:  # repeat forever
        try:
            result=input(prompt) # read the input
            # if we get here no exception was raised
            if result=='':
                #don't accept empty lines
                print('Please enter text')
            else:
                # break out of the loop
                break
        except KeyboardInterrupt:
            # if we get here the user pressed CTRL+C
            print('Please enter text')
            if DEBUG_MODE:
                raise Exception('Keyboard interrupt')

    # return the result
    return result


def read_number(prompt,function):
    '''
    Displays a prompt and reads in a floating point number.
    Keyboard interrupts (CTRL+C) are ignored
    Invalid numbers are rejected
    returns a float containing the value input by the user
    '''
    while True:  # repeat forever
        try:
            number_text=read_text(prompt)
            result=function(number_text) # read the input
            # if we get here no exception was raised
            # break out of the loop
            break
        except ValueError:
            # if we get here the user entered an invalid number
            print('Please enter a number')

    # return the result
    return result

def read_number_ranged(prompt, function, min_value, max_value):
    '''
    Displays a prompt and reads in a number.
    min_value gives the inclusive minimum value
    max_value gives the inclusive maximum value
    Raises an exception if max and min are the wrong way round
    Keyboard interrupts (CTRL+C) are ignored
    Invalid numbers are rejected
    returns a number containing the value input by the user
    '''
    if min_value>max_value:
        # If we get here the min and the max
        # are wrong way round
        raise Exception('Min value is greater than max value')
    while True:  # repeat forever
        result=read_number(prompt,function)
        if result<min_value:
            # Value entered is too low
            print('That number is too low')
            print('Minimum value is:',min_value)
            # Repeat the number reading loop
            continue
        if result>max_value:
            # Value entered is too high
            print('That number is too high')
            print('Maximum value is:',max_value)
            # Repeat the number reading loop
            continue
        # If we get here the number is valid
        # break out of the loop
        break
    # return the result
    return result

def read_float(prompt):
    '''
    Displays a prompt and reads in a floating point number.
    Keyboard interrupts (CTRL+C) are ignored
    Invalid numbers are rejected
    returns a float containing the value input by the user
    '''
    return read_number(prompt,float)

def read_int(prompt):
    '''
    Displays a prompt and reads in an integer number.
    Keyboard interrupts (CTRL+C) are ignored
    Invalid numbers are rejected
    returns an int containing the value input by the user
    '''
    return read_number(prompt,int)

def read_float_ranged(prompt, min_value, max_value):
    '''
    Displays a prompt and reads in a floating point number.
    min_value gives the inclusive minimum value
    max_value gives the inclusive maximum value
    Raises an exception if max and min are the wrong way round
    Keyboard interrupts (CTRL+C) are ignored
    Invalid numbers are rejected
    returns a number containing the value input by the user
    '''
    return read_number_ranged(prompt,float,min_value,max_value)

def read_int_ranged(prompt, min_value, max_value):
    '''
    Displays a prompt and reads in an integer point number.
    min_value gives the inclusive minimum value
    max_value gives the inclusive maximum value
    Raises an exception if max and min are the wrong way round
    Keyboard interrupts (CTRL+C) are ignored
    Invalid numbers are rejected
    returns a number containing the value input by the user
    '''
    return read_number_ranged(prompt,int,min_value,max_value)
```