NAME:

NUMBER:

Date: 21.11.2023, Tuesday
Time: 09:00 (90 minutes)
Place: D1, D2
Lecturer: İrem Ülkü

ANKARA UNIVERSITY, COMPUTER ENGINEERING DEPARTMENT

COM1001: Computer Programming I (Fall 2023-24) - MIDTERM EXAM (100P)

(32P) (8P*4) QUESTION 1:

Below Python function "*Selection_Sort*" performs selection sort operation. Consider a list of 10 elements list[0], list[1,……list[N-1]. First you will search for the position of the smallest element from **list[0] to list[N-1]** and then interchange the smallest element with list[0]. Now you will search for the position of the second smallest element from **list[1] to list[n-1]** and then interchange that smallest element with list[1]. This process continues till the end and finally we obtain the sorted list. The whole process of selection sort will be as shown:

Consider the unsorted list of 8 elements as: [74, 34, 42, 13, 87, 24, 64, 57]

Iteration 1

1. Search the smallest element from list[0] to list[N-1].

2. Interchange list[0] with the smallest element.

Result: list[0] is sorted: [**13**, 34, 42, **74**, 87, 24, 64, 57]

Iteration N-1

1. Search the smallest element from list[N-1] to list[N-1].

2. Interchange list[N-1] with the smallest element.

Result: list[0]……list[N-1]: Final Sorted List is: [13, 24, 34, 42, 57, 64, 74, 87]

Hint: Selection sort repeatedly selects the smallest element and swaps it with the first element in the remaining list.

Fill the 4 blank spaces in this code.

```
def Selection_Sort(MyList):
    for i in range(len(MyList)-1):
        k=i #ith element is assumed to be smallest
        for j in ----------range(i+1,len(MyList))----------:
            if(----------MyList[j]<MyList[k]----------):
                k=j
        if (k!=i):
            temp=MyList[i]
            MyList[i]=---------- MyList[k]----------
            MyList[k]---------- temp----------
```

(20P) (10P*2) QUESTION 2:

A sublist is a list that makes up part of a larger list. A sublist may be a list containing a single element, multiple elements, or even no elements at all. Below Python function "*allSublists*" returns a list containing every possible sublist of a list. For example, the sublists of [1, 2, 3] are [], [1], [2], [3], [1, 2], [2, 3] and [1, 2, 3]. Function always returns a list containing at least the empty list because the empty list is a sublist of every list.

Fill the 2 blank spaces in this code.

```
def allSublists(data):
    sublists = [[]]
    for length in range(----------1, len(data) + 1-------------):
        for i in range(---------0, len(data) - length + 1------------):
            sublists.append(data[i : i + length])
    return sublists

def main():
    data = [1, 2, 3]
    print("The sublists of data are included in the following list:")
    print(allSublists(data))

main()
```

(30P) (10P*3) QUESTION 3:

Below Python program prints reverse mirrored right triangle number as given in the figure:

Fill the 3 blank spaces in this code.

```
num = int (input("Enter the total number of rows:"))
for i in range(1, num + 1):
    for j in ------------range(1, num +1)-------------:
        if --------------(j<i)-----------:
            print(" ", end=" ")
        else:
            --------------print(j, end=" ")------------
    print()
```

(18P) (6P*3) QUESTION 4:

Below Python function capitalizes the appropriate characters in a string. The rules are:

- A lowercase "i" should be replaced with an uppercase "I" if it is both preceded and followed by a space.
- The first character in the string should also be capitalized, as well as the first non-space character after **".", "!"** or **"?"**.

For example, if the function is provided with the string **"what time do i have to be there? what's the address?"** then the function "*capitalize*" should return the string **"What time do I have to be there? What's the address?"**.

There are 3 errors (can be syntax, runtime, or logical) in this code. Please find and correct them.

```
def capitalize(s):
    # Correct the capitalization for i
    result = s.replace(" i ", " I ")

    # Capitalize the first character of the string
    if len(s) > 0:
        result = result[0].upper() + result[1:len(result)]

    # Capitalize the first letter that follows ".", "!" or "?"
    pos = 0
    while pos < len(s):
        if result[pos] == "." or result[pos] == "!" or result[pos] == "?":
            pos +=1
            while pos < len(s) and result[pos] == " ":
                pos += 1
            # If we haven't reach the end of the string then replace the current
            # character with its uppercase equivalent
            if pos < len(s):
                result = result[0:pos] + result[pos].upper() + result[pos+1:len(result)]
        # Move to the next character
        pos += 1
    return result
```