9618 Paper 42 Practical – AS & A Level Computer Science **evidence document**

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

Insert your centre number, name and candidate number into the header above.

Save this evidence document as **evidence_** followed by your **centre number_ candidate number**.
For example: **evidence_ zz999_9999**

Screenshots or program listings must be copied into appropriate cells in the following table.

Examiners must be able to read the contents including any screenshots without the use of a magnifying glass. Answers that are not readable or missing will not be awarded any marks.

Save this evidence document at regular intervals, for example every 10 minutes.

Question 1

Part 1(a)

2

```
StackData = [] # 1D Array of 10 integer elements
StackPointer = 0
```

Part 1(b)

2

```
def PrintStack():
    global StackData, StackPointer

    print("Stack values: ", end="")
    for i in range(StackPointer): # Loop over the Stack
        print(StackData[i], end=" ") # Print value

    print("\nPointer Value:", StackPointer) # Print pointer value
```

Part 1(c)

6

```
def PushToStack(value):
    global StackData, StackPointer
    if StackPointer == 10: # Check if stack is full
        return False
    else:
        StackData[StackPointer] = value # Push the value to the stack
        StackPointer += 1 # Move the pointer to the next free space
        return True
```

Part 1(d)(i)

5

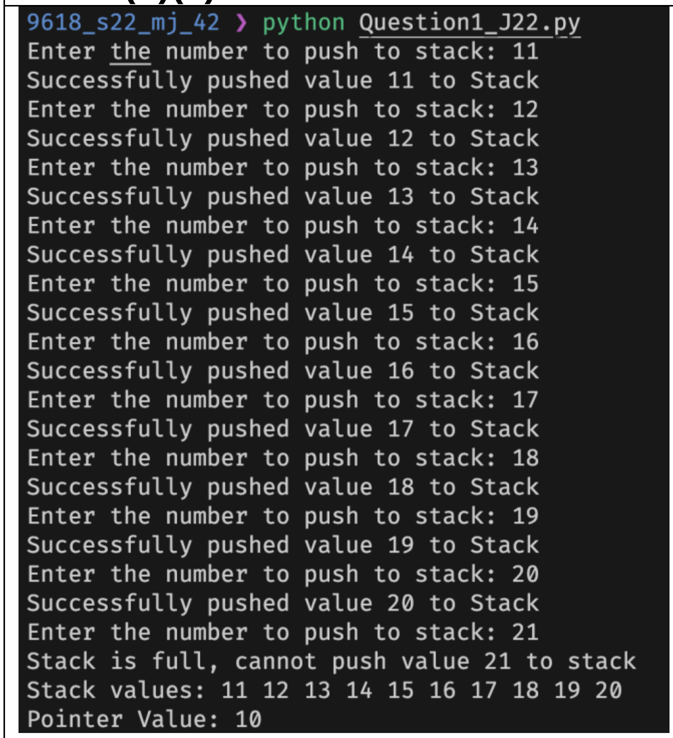
```
for i in range(11): # run 11 times
    InputValue = int(input("Enter the number to push to stack: "))

    PushResponse = PushToStack(InputValue)

    if PushResponse == True: # Check if value is pushed to stack
        print("Successfully pushed value", InputValue, "to Stack") #
```

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

```
Output sucessful message
    else:
        print("Stack is full, cannot push value", InputValue, "to stack")
# Output failed message

PrintStack()
```

Part 1(d)(ii)

1



```
9618_s22_mj_42 ❯ python Question1_J22.py
Enter the number to push to stack: 11
Successfully pushed value 11 to Stack
Enter the number to push to stack: 12
Successfully pushed value 12 to Stack
Enter the number to push to stack: 13
Successfully pushed value 13 to Stack
Enter the number to push to stack: 14
Successfully pushed value 14 to Stack
Enter the number to push to stack: 15
Successfully pushed value 15 to Stack
Enter the number to push to stack: 16
Successfully pushed value 16 to Stack
Enter the number to push to stack: 17
Successfully pushed value 17 to Stack
Enter the number to push to stack: 18
Successfully pushed value 18 to Stack
Enter the number to push to stack: 19
Successfully pushed value 19 to Stack
Enter the number to push to stack: 20
Successfully pushed value 20 to Stack
Enter the number to push to stack: 21
Stack is full, cannot push value 21 to stack
Stack values: 11 12 13 14 15 16 17 18 19 20
Pointer Value: 10
```

Part 1(e)(i)

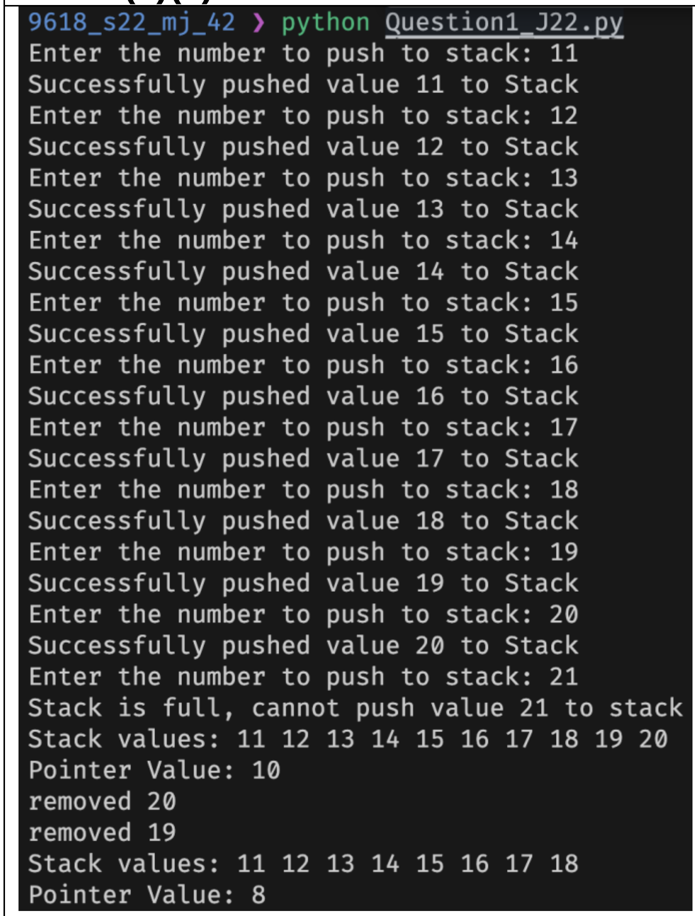
5

```
def Pop():
    global StackData, StackPointer

    if StackPointer == 0: # Check if stack is empty
        return -1

    TopValue = StackData[StackPointer-1] # Get the top value
    StackData.pop() # Remove the value from list
    StackPointer = StackPointer - 1 # Move the pointer to the previous
free space
    return TopValue
```

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

Part 1(e)(ii)

2



```
9618_s22_mj_42 ❯ python Question1_J22.py
Enter the number to push to stack: 11
Successfully pushed value 11 to Stack
Enter the number to push to stack: 12
Successfully pushed value 12 to Stack
Enter the number to push to stack: 13
Successfully pushed value 13 to Stack
Enter the number to push to stack: 14
Successfully pushed value 14 to Stack
Enter the number to push to stack: 15
Successfully pushed value 15 to Stack
Enter the number to push to stack: 16
Successfully pushed value 16 to Stack
Enter the number to push to stack: 17
Successfully pushed value 17 to Stack
Enter the number to push to stack: 18
Successfully pushed value 18 to Stack
Enter the number to push to stack: 19
Successfully pushed value 19 to Stack
Enter the number to push to stack: 20
Successfully pushed value 20 to Stack
Enter the number to push to stack: 21
Stack is full, cannot push value 21 to stack
Stack values: 11 12 13 14 15 16 17 18 19 20
Pointer Value: 10
removed 20
removed 19
Stack values: 11 12 13 14 15 16 17 18
Pointer Value: 8
```

Question 2

Part 2(a)

3

```
import random

ArrayData = [] # 2D Array of 10 by 10 integer elements

for i in range(10):
    InsideArray = [] # Blank array for 10 integer elements
    for j in range(10):
        RandomNumber = random.randint(1,100) # Generate a random number
between 1 and 100
        InsideArray.append(RandomNumber) # Add to the subarray

    ArrayData.append(InsideArray) # Append to main array
```

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

Part 2(b)(i)

5

```
ArrayLength = 10

# Increase all index search by 1 as index starts at 0 and ends at (n-1)
for X in range(ArrayLength):
    for Y in range(ArrayLength-1):
        for Z in range(ArrayLength-Y-1):
            if ArrayData[X][Z] > ArrayData[X][Z+1]:
                TempValue = ArrayData[X][Z]
                ArrayData[X][Z] = ArrayData[X][Z+1]
                ArrayData[X][Z+1] = TempValue
```

Part 2(b)(ii)

2

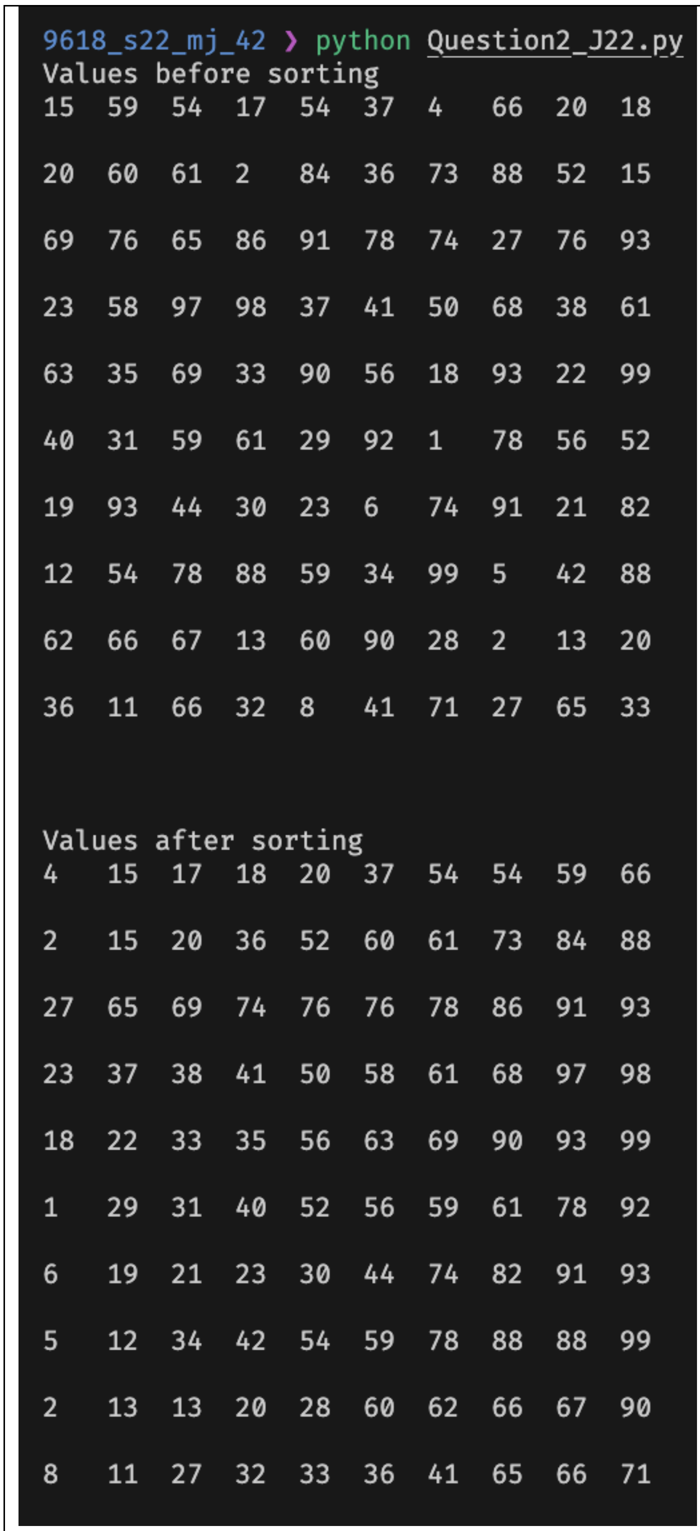
```
def printArrayData():
    global ArrayData
    for i in range(10): # Loop over first dimension
        for j in range(10): # Loop over second dimension
            print(ArrayData[i][j], end="\t") # Output value [row][column]
        print("\n") # Create new line (new row)
```

make sure copy all in here

Part 2(b)(iii)

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

1



```
9618_s22_mj_42 ❯ python Question2_J22.py
Values before sorting
15  59  54  17  54  37  4   66  20  18

20  60  61  2   84  36  73  88  52  15

69  76  65  86  91  78  74  27  76  93

23  58  97  98  37  41  50  68  38  61

63  35  69  33  90  56  18  93  22  99

40  31  59  61  29  92  1   78  56  52

19  93  44  30  23  6   74  91  21  82

12  54  78  88  59  34  99  5   42  88

62  66  67  13  60  90  28  2   13  20

36  11  66  32  8   41  71  27  65  33


Values after sorting
4   15  17  18  20  37  54  54  59  66

2   15  20  36  52  60  61  73  84  88

27  65  69  74  76  76  78  86  91  93

23  37  38  41  50  58  61  68  97  98

18  22  33  35  56  63  69  90  93  99

1   29  31  40  52  56  59  61  78  92

6   19  21  23  30  44  74  82  91  93

5   12  34  42  54  59  78  88  88  99

2   13  13  20  28  60  62  66  67  90

8   11  27  32  33  36  41  65  66  71
```

Part 2(c)(i)

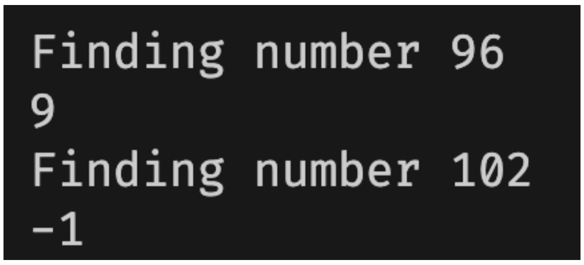
```
def BinarySearch(SearchArray, Lower, Upper, SearchValue) -> int:
```

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

7

```
    global ArrayData
    if Upper > Lower:
        Mid = (Lower + (Upper -1)) // 2
        if SearchArray[0][Mid] == SearchValue:
            return Mid
        else:
            if SearchArray[0][Mid] > SearchValue:
                return BinarySearch(SearchArray, 0, Mid - 1, SearchValue)
            else:
                return BinarySearch(SearchArray, Mid+1, Upper, 
SearchValue)

    return -1
```

Part 2(c)(ii)



```
Finding number 96
9
Finding number 102
-1
```

2

Question 3

Part 3(a)

5

```
class Card:
    # private __number: INTEGER number of the card
    # private __colour: STRING colour of the card
    def __init__(self, number, colour):
        self.__number = number # Create a private attribute to store 
number
        self.__colour = colour # Create a private attribute to store 
colour
```

Part 3(b)

3

```
    def GetNumber(self):
        return self.__number

    def GetColour(self):
        return self.__colour
```

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

Part 3(c)

6

```
CardArray = [] # Array of 30 elements with datatype Card

CardValuesFile = open('CardValues.txt', 'r') # Open CardValues.txt
CardValuesFileData = CardValuesFile.read().split("\n") # Read the data 
line by line
NumberOfLines = 60
CardValuesFile.close() # Close the file

for line in range(0, NumberOfLines, 2): # Loop through 2 lines at a time
    CardNumber = int(CardValuesFileData[line]) # Get card number
    CardColour = CardValuesFileData[line+1] # Get card colour

    CardArray.append(Card(CardNumber, CardColour)) # Add the Card to the 
array
```

Part 3(d)

6

```
def ChooseCard():
    global ChosenCards
    CardAvailible = False

    while CardAvailible == False:
        CardChoice = int(input("Select the card index you want: "))
        if CardChoice >= 1 and CardChoice <= 30: # Check if valid index
            if CardChoice not in ChosenCards: # Check if card is already 
chosen
                CardAvailible = True # Set valid varible to True
                ChosenCards.append(CardChoice) # Add choice to chosen 
cards
            else:
                print("Card already chosen, please choose again") # 
Display message asking user to choose again
        else:
            print("Not valid index, please choose from 1 to 30 
(inclusive)") # Display message asking user to choose again

    return CardChoice # return index of chosen card
```

Part 3(e)(i)

5

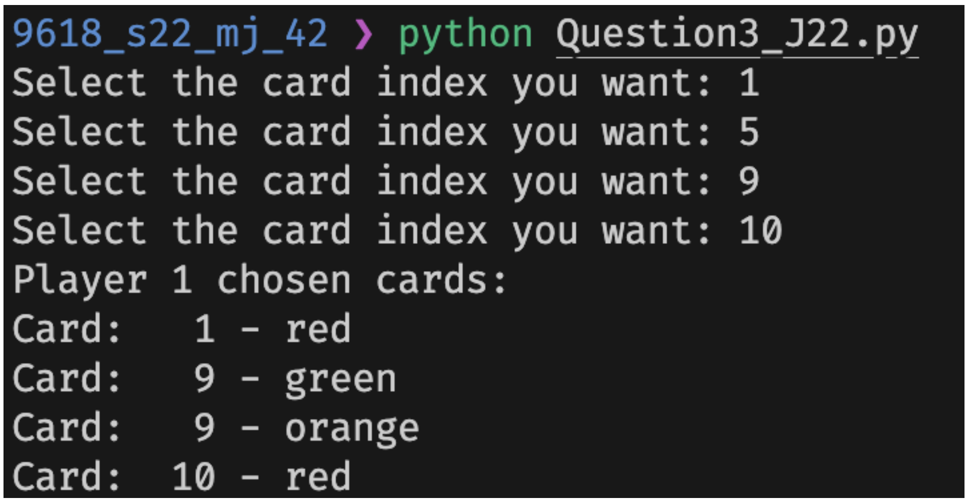
```
Player1 = [] # Array for player 1 chosen cards of type Card

for i in range(4):
    Player1CardChoice = ChooseCard() # Ask player to choose card
    Player1.append(CardArray[Player1CardChoice-1]) # Add the chosen card 
to Player1's array minus 1 because index starts at 0

print("Player 1 chosen cards:")
for i in range(4): # Loop over Player1 chosen cards
```

Centre Number:
Candidate Name: NGO THO QUANG
Candidate Number:

```
    P1Card = Player1[i]
    P1CardNumber = P1Card.GetNumber() # Get card number
    P1CardColour = P1Card.GetColour() # Get card colour
    print("Card:", P1CardNumber, "-", P1CardColour) # Output card number
and colour
```

Part 3(e)(ii)

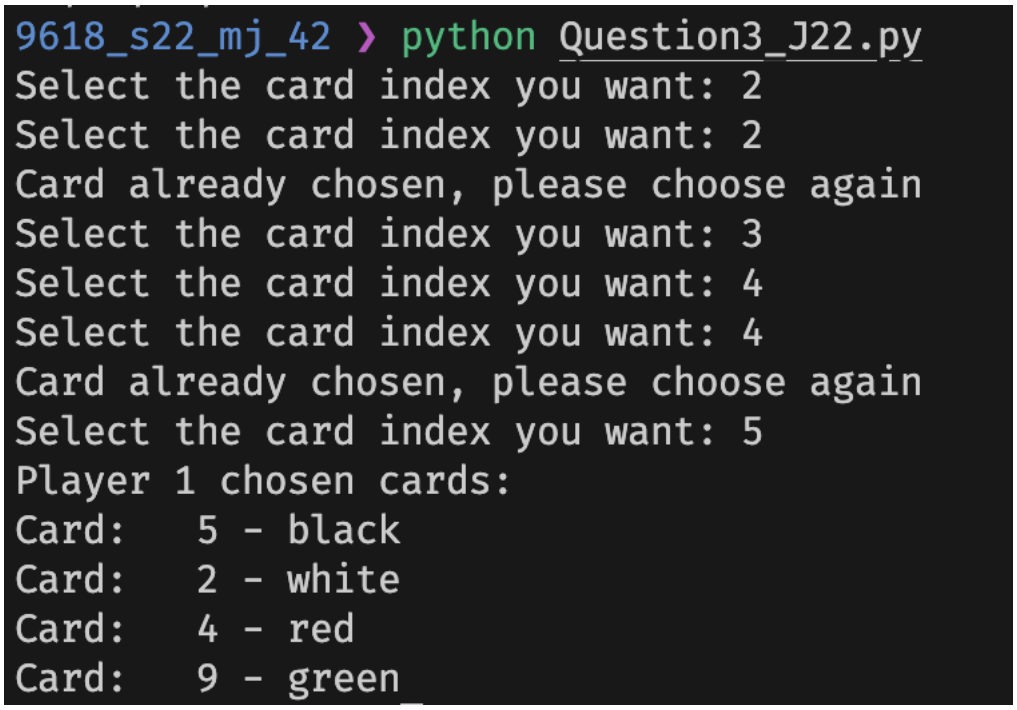
TEST 1:



```
9618_s22_mj_42 ❯ python Question3_J22.py
Select the card index you want: 1
Select the card index you want: 5
Select the card index you want: 9
Select the card index you want: 10
Player 1 chosen cards:
Card:   1 - red
Card:   9 - green
Card:   9 - orange
Card:  10 - red
```

TEST 2:

1



```
9618_s22_mj_42 ❯ python Question3_J22.py
Select the card index you want: 2
Select the card index you want: 2
Card already chosen, please choose again
Select the card index you want: 3
Select the card index you want: 4
Select the card index you want: 4
Card already chosen, please choose again
Select the card index you want: 5
Player 1 chosen cards:
Card:   5 - black
Card:   2 - white
Card:   4 - red
Card:   9 - green
```

Time taken: 2:01:50

69/75