GERMAN-MALAYSIAN INSTITUTE

UNIVERSITI TEKNOLOGI PETRONAS

FINAL EXAMINATION SEPTEMBER 2023 SEMESTER

COURSE	:	FBT0015 - STRUCTURED ALGORITHM & PROGRAMMING
DATE	:	21 DECEMBER 2023 (THURSDAY)
TIME	:	9:00 AM - 12:00 NOON (3 HOURS)

INSTRUCTIONS TO CANDIDATES

SECTION A :

1. Answer **ALL** questions in the **OMR** sheet.
2. Use **2B pencil** only.

SECTION B :

1. Answer **ALL** questions in the Answer Booklet.
2. Begin **EACH** answer on a new page in the Answer Booklet given.
3. Indicate clearly answers that are cancelled, if any.
4. Where applicable, show clearly steps taken in arriving at the solutions and indicate **ALL** assumptions, if any.
5. **DO NOT** open this Question Booklet until instructed.

Note :

i. There are **TWENTY-TWO (22)** pages in this Question Booklet including the cover page .

ii. **DOUBLE-SIDED** Question Booklet.

SECTION A
[40 MARKS]

1. ______ repetition in programming is used when the number of iterations in a loop cannot be determined in advance.

   A. Fixed
   B. Variable
   C. Dynamic
   D. Conditional

2. The primary effect of using the `break` keyword within a loop in a programming language is to ______.

   A. reverse the order of iteration
   B. immediately terminates the loop
   C. pause the execution of the loop until user input is received
   D. skip the current iteration and proceeds to the next iteration

3. Identify the **CORRECT** statement for tuples and lists in Python.

   A. Tuples are ordered collections, but lists are not.
   B. Tuples support reordering elements, while lists do not.
   C. Tuples are created using square brackets, whereas lists use parentheses.
   D. Lists allow for dynamic elements manipulation, while tuples have fixed elements.

4. In Python, the **CORRECT** syntax to create a `while` loop is ______.

   A. `do n times :`
   B. `while X < 3 :`
   C. `for x in range (-n) :`
   D. `while x in range (n) :`

5. Given a list, `list1 = [4,5,6]`.
   `list4 = 2 * list1` will give the output of ______.

   A. `[4,5,6]`
   B. `[8,10,12]`
   C. `[4,5,6,4,5,6]`
   D.
   ```
   8
   10
   12
   ```

6. In the following Python code, identify the number of iterations that will occur in the loop.

```
count = 8
while count > 2:
    count -= 1
```

   A. 2 iterations
   B. 3 iterations
   C. 8 iterations
   D. no iteration

7. Identify the **CORRECT** Python syntax example that demonstrates the usage of a `for` loop.

   A. `if x > 0 :`
   B. `while False :`
   C. `for i = 10 :`
   D. `for j in range (10,1,-5) :`

8. Identify the error from the following Python code.

```
my_list = [6, 32, 4, 1]
for i in range(len(my_list)):
    print(my_list[i])
```

A. There is no error in the code.
B. The range function uses `len(my_list)` as an argument.
C. The list was defined by using square brackets instead of parentheses.
D. The loop counter, `i` was initialized to `0` and could lead to index out of range error.

9. The following syntax for tuple produces no error **EXCEPT** ______.

A. `sample = (12,)`
B. `sample = ('Apple',Cherry',3)`
C. `sample = ('1','2','3','4',5)`
D. `sample = ('apple' ,' rose' ,)`

10. Identify for any error from the following Python code.

```
List1 = [1, 2, 3, 4, 5]
for i in range(5):
    if List1[i] % 2 == 0:
        print("Even:", List1[i])
    else
        print("Odd:", List1[i])
```

A. There is no error in the code.
B. The range function should include a step value.
C. There should be a colon (:) after the else statement.
D. `for` loop should be replaced with a `while` loop instead.

Question 11 and **Question 12** are referring to **FIGURE Q11** below.

```
i = 1
x = 2
while i <= 2:
    x = x * x
    print (x, end = ' ')
i += 1
```

FIGURE Q11: Sample coding

11. The Python code in **FIGURE Q11** will result in an infinite loop. Identify the **CORRECT** action to make sure the code is not producing an infinite loop.

    A. Remove all `i` variables.

    B. Re-indent `i += 1` into the `while` loop.

    C. Change `while` loop to `for i in range(5).`

    D. Reposition `print(x)` to the outside of the `while` loop.

12. Assuming the Python code in **FIGURE Q11** already being corrected, trace the output of the code.

    A.
    ```
    2 4
    ```
    B.
    ```
    4 16
    ```
    C.
    ```
    2
    4
    ```
    D.
    ```
    4
    16
    ```

13. Trace the output of the following Python code.

```
numbers = [1, 2, 3, 4, 5]
for num in numbers:
    print(num * 3, end = " ")
```

A. `1 2 3 4 5`

B. `3 6 9 12 15`

C.
```
1
2
3
4
5
```

D.
```
3
6
9
12
15
```

14. Identify the **CORRECT** output of the following Python code.

```
my_tuple = (2 , 1 , 4 , 5)
for item in my_tuple:
    print(item, end = '#')
    print('')
```

A. `1234#`

B. `2145#`

C. `1#2#4#5#`

D.
```
2#
1#
4#
5#
```

15. Given the Python code below.

```
numbers = [1, 2, 3, 4, 5]
total = 0
for num in numbers:
    total += num
total = num
print("Total: ", total)
```

The output of this code is `Total :` ______.

A. 0
B. 5
C. 15
D. 120

16. ______ keyword is used to define a function in Python.

A. `def`
B. `func`
C. `define`
D. `function`

17. In Python, the purpose of a function's `return` statement is to ______.

A. define a function's name and parameters
B. execute a specific block of code within a function
C. provide a comment or documentation string for the function
D. pass data or a value from the function to the function's caller

18. The purpose of arguments in function are ______.

A. to serve as the function's name

B. to define the function's return value

C. optional statements within the function body

D. placeholders for value to be passed to the function

19. Choose the **CORRECT** syntax to define a Python function called `add` that takes two parameters, `a` and `b`, and returns their product.

A.
```
add(a, b):
 a + b
```

B.
```
def add ():
 return a + b
```

C.
```
def add(a, b):
 return a + b
```

D.
```
function add(a, b)
 return a + b
```

20. The following are usable function names in Python **EXCEPT** ______.

A. `My_function`

B. `420function`

C. `_function1234`

D. All names can be used as function names.

21. Recursive function is ______.

A. more memory-efficient than iterative solutions

B. implemented only in high-level programming languages

C. always faster and more efficient than their iterative counterparts

D. a programming technique where a function calls itself to solve a problem

22. The Python code below can be executed. However, there is no output from the code. Identify the issue that causes the situation.

```
def calculate_average(x, y, z) :
    result = (x + y + z) / 3
    return result
```

A. The function name is unsuitable.
B. The initialization of function is incorrect.
C. The function call outside of the function is missing.
D. The parameters are enclosed in parentheses, not square brackets.

23. Identify the error in the following Python code.

```
def outer ():
    def inner ():
        print("This is an inner function")
    print("This is an outer function")

inner ()
```

A. There is no error in the code.
B. Nested functions are not allowed in Python.
C. The inner function should be called within the inner function.
D. The inner function should be defined before the outer function.

24. In the following Python code, identify the error, and how can it be corrected.

```
def add_numbers(a, b):
    result = a + b
    return result


number1 = 1
number2 = "2"
sum_result = add_numbers(number1, number2)
print("The sum is:", sum_result)
```

A. There is no error in the code.

B. The error is a missing data type declaration for the `number2` variable, and it can be corrected by specifying `number2` as an integer.

C. The error is a missing return statement, and it can be corrected by adding return result at the end of the `add_numbers` function.

D. The error is in the function call, and it can be corrected by changing `add_numbers(number1,number2)` to `add_numbers(number1, int(number2))`.

25. The output of the code is `None`. Identify the **CORRECT** solution that will change the output of the program to `10`.

```
def c(a,b): #line1
    if a > b: #line2
        r = a #line3
    else:       #line4
        r = b #line5
print(c(10, 5)) #line 6
```

A. Replace `a` and `b` with `a = 10` and `b = 5` at line 1

B. Add `return r` after line 5 outside `else` indentation.

C. Add `print (r)` after line 5 outside `else` indentation.

D. Assign function call `c(10,5)` to a variable before line 6.

26. Identify the **CORRECT** output of the following Python code.

```
def power(x, n):
    if n == 0:
        return 1
    else:
        return x * power(x, n - 1)

def my_function(a, b):
    result = power(a, b)

output = my_function(2, 3)
print(output)
```

A. 2

B. 6

C. 8

D. None

27. Consider the following Python code.

```
def add_numbers(a, b):
    result = a + b
    return result

x = 5
y = 2
print(add_numbers(x, y))
```

Trace the output of the Python code.

A. 2

B. 5

C. 7

D. 12

28. Find the **CORRECT** output of the following Python code:

```
def multiply(num):
    result = num * 2
    return result


value = 5
print(multiply(value))
```

A. 5

B. 10

C. 15

D. No output is displayed.

29. Trace the output of the following Python code.

```
def faa(x, y):
    x, y[0] = 12, 1 + 3


def fbb():
    a, b = 5,[1,2,4]
    faa (a, b)
    print ("Value 1 is", a , "and value 2 is", b)
fbb()
```

A. Value 1 is 5 and value 2 is [1,2,4]

B. Value 1 is 5 and value 2 is [4,2,4]

C. Value 1 is 12 and value 2 is [1,2,4]

D. Value 1 is 12 and value 2 is [4,2,4]

30. Identify the output of the following Python code.

```
n = 2
def outer (n):
    def inner (x):
        if x == 0:
            return 0
        elif x == 1:
            return 1
        else:
            return inner (x-1) + inner (x-2)

    if n <= 0:
        return None
    elif n == 1:
        return 0
    elif n == 2:
        return 1
    else:
        result = 0
        for k in range(1, n):
            if k % 2 == 0:
                result += inner (n)
            else:
                result -= inner (n)
k = outer (5)
print(n)
```

A. 0
B. 1
C. 2
D. None

31. The primary purpose of using a data file in a program is to ______.

A. view or create program results
B. temporarily store data during program execution
C. create an easily readable and editable file for human users
D. permanently save data that can be read or written by the program

32. ______ file mode in Python opens an existing file for writing. If the file does not exist yet, an error will appear.

A. `'r'`
B. `'w'`
C. `'a'`
D. `'wb'`

33. After opening a file for reading, ______ Python method is used to read only one line from the file.

A. `read()`
B. `readchar()`
C. `readline()`
D. `readlines()`

34. Consider the following Python code that attempts to read a text file.

```
file = open ("Intro.txt", "r")
content = file.read()
print("File content:", content)
file.close()
```

Identify for any syntax error in the code.

A. There is no error in the code.
B. The file mode should be `w` instead of `r`.
C. There should be a `try` and `except` block for error handling.
D. The file should be opened using `file.open()` instead of `open()`.

35. Examine the following Python code designed to write data to a text file.

```
file = open ("listNum.txt", "w")
data = ["1", "2", "3"]
file.write(data)
file.close()
```

Find the syntax error in the code.

A. The file mode is incorrect.
B. There is no error in the code.
C. The `file.stop()` statement is missing.
D. The `write()` method should accept a string, not a list.

36. The following Python code will read a text file and display all of its content.

```
file = open ("student_details.txt", "r")
lines = file.readln()
for line in lines:
    print(line, end = "")
file.close()
```

The code should have displayed a list of names from `student_details.txt`. However, when the code is executed, syntax errors keep on appearing. Identify the best solution to solve the error in the code.

A. The file mode is incorrect.
B. The file mode should be `w` instead of `r`.
C. The `readln()` method should be replaced with `read()`.
D. `lines = file.readln()` should be placed inside of the loop.

37. Consider the following Python code that reads a text file named `data.txt`:

```
file = open("data.txt", "r")
lines = file.readlines()
file.close()

count = 0
for line in lines:
    count += 1
print("The number of lines in the file is:", count)
```

Trace the **CORRECT** output of this code when it reads a file containing **five (5)** names on **five (5)** different lines.

A. `The number of lines in the file is: 0`
B. `The number of lines in the file is: 1`
C. `The number of lines in the file is: 5`
D. Error

38. Identify the **CORRECT** output of the following Python code, where the `Intro.txt` contains the text "`Hello, World!`" in one line.

```
file = open("Intro.txt", "r")
content = "Mayday."
content = file.read()
file.close()

print("File content: ", content)
```

A. `Mayday.`

B. `File content: Mayday.`

C. `File content: Intro.txt`

D. `File content: Hello, World!`

39. Trace the output of the following Python program, where the `Statement.txt` contains the text "`sunny skies bring joy`" in one line.

```
file = open("Statement.txt", "r")
lines = file.readlines()
for line in lines:
    words = line.split()
    for word in words:
        if len(word) < 4:
            print(word)
file.close()
```

A. `joy`

B. `bring`

C. `sunny skies`

D.
```
sunny
skies
bring
joy
```

40. The following Python code reads a text file and processes its content:

```
file = open("Message.txt", "r")
lines = file.readlines()
file.close()

result = ""
for line in lines:
    result += line

print(result)
```

Find the **CORRECT** output of this code program when it reads a text file, `Message.txt` with the following content:

```
Hi,
Are you ok?
I feel great!
```

A.
```
Hi,
Are you ok?
I feel great!
```

B.
```
Hi,

Are you ok?

I feel great!
```

C. `Hi,Are you ok?I feel great!`

D. `Hi, Are you ok? I feel great!`

SECTION B
[60 MARKS]

1. a. Identify the differences between global and local variables in Python by providing an example of each.

[6 marks]

b. Explain the concept of a recursive function in Python by providing an example.

[4 marks]

c. Identify the key steps to open and write a new text file in Python by providing a code example.

[6 marks]

d. Compare the differences between reading a file by using `readline()` and by using `read()`.

[4 marks]

2. Trace the output of the following Python codes:

a.

```
n = 5
i = 1
while i <= n:
    j = 1
    while j <= i:
        print(j, end=" ")
        j += 1
    print()
    i += 1
```

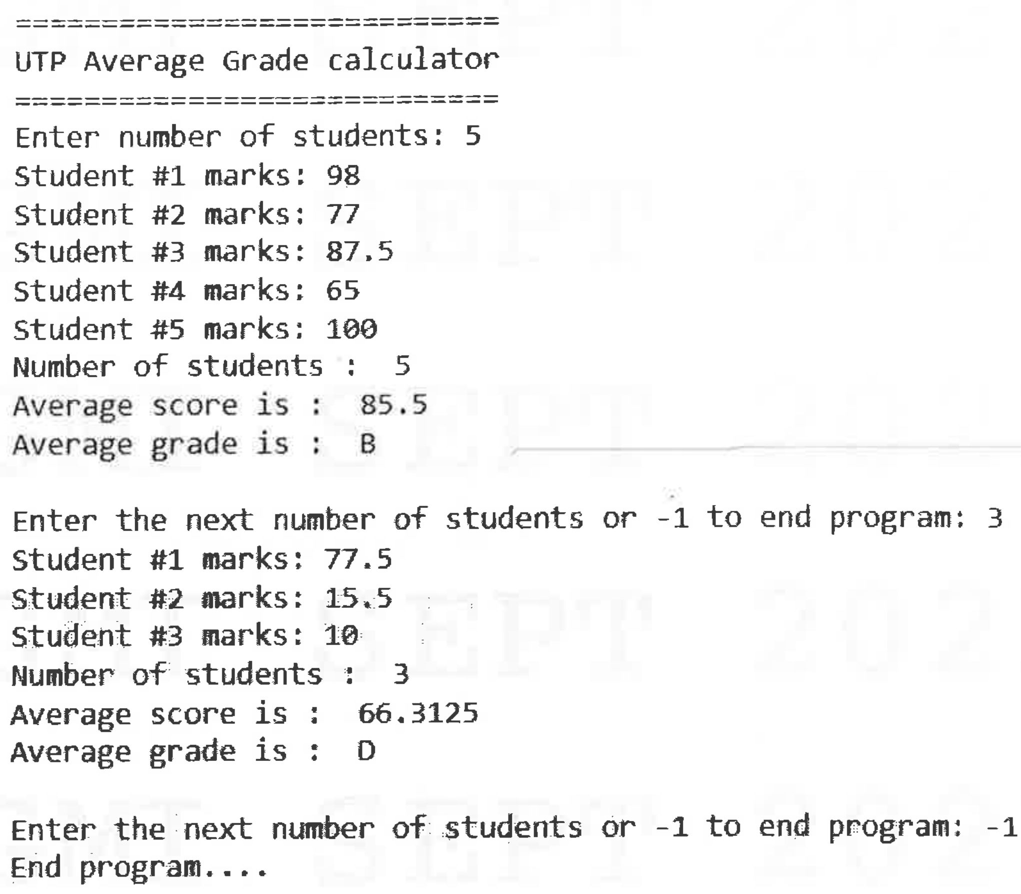
[10 marks]

b.

```
for i in range(1, 10):
    for j in range(i, 10):
        print("u", end="#")
    print()
```

[10 marks]

3. You are tasked by UTP to create a Python program to calculate the average grade of a group of students based on their test scores. The program should also assign a letter grade to the calculated average. Below is the sample interface of the program:



```
============================
UTP Average Grade calculator
============================
Enter number of students: 5
Student #1 marks: 98
Student #2 marks: 77
Student #3 marks: 87.5
Student #4 marks: 65
Student #5 marks: 100
Number of students :  5
Average score is :  85.5
Average grade is :  B

Enter the next number of students or -1 to end program: 3
Student #1 marks: 77.5
Student #2 marks: 15.5
Student #3 marks: 10
Number of students :  3
Average score is :  66.3125
Average grade is :  D

Enter the next number of students or -1 to end program: -1
End program....
```

FIGURE Q3: Sample of output program

a. Write a function called `calc_average`. This function takes in **one** (1) list called `list_scores` as the parameter. It will calculate the average value of `list_scores` and save it to a variable called `average`. The function requires to return the result of the calculation.

[4 marks]

b. Write a Python function called `assign_grades` that takes in **one** (1) variable, `score` as the parameter. The function will return a grade if the condition is true. The conditions and respective grades are as follows:

Conditions	**Grade**
`score` is 90 and above	A
`score` is between 80 and 89	B
`score` is between 70 and 79	C
`score` is between 60 and 69	D
`score` is 59 and lower	F

[8 marks]

c. Referring to **FIGURE Q3** as the sample outputs, write the Python code for the main program that will execute the following:

- Ask the user to enter the number of students and store it in variable `num`.
- Ask the user to enter the marks for each student into variable `marks`. All student marks will be saved into a list named `main_list`.
- Display the average score of the marks entered, by calling the function `calc_average` from **part a** with `main_list` as the function argument. Assign the function call to a variable named `average`.
- Display the average grade of the average score by calling the function `assign_grades` from **part b** with `average` as the function argument.
- Ask the user to enter the next number of students or -1 to end the program. An end message will be displayed after the user enters -1.

[8 marks]

-END OF PAPER-