A hospital uses spreadsheet software to generate health reports for each patient who have participated in health screening.

You are required to finish setting up the spreadsheet to evaluate the test results for a patient.

Open the file **HEALTHREPORT.xlsx**. You will see the following data.

Save the file as **MYHEALTH\_**<your name>\_<class>\_<index number>.xlsx

	А	В	С	D	E	F
1		Temase	ek Gei	neral Hospital		
2						
3	Health R	eport Year:	1900	Date:		
4						
5	Name:	Lily Tan				
6	ID:	S7824752E				
7	Date of Birth:	11/11/1978		Age:	-78	
8						
9			Basic	Tests		
10						
11	Res	ults		Eval	uation	
12	Height:	1.61	m	BMI:		kg/m <sup>2</sup>
13	Weight:	48	kg	BMI Category:		
14	BP (Systolic):	98	mmHg	Blood Pressure		
15	BP (Diastolic):	74	mmHg	Category:		
16						
17			Bloo	d Test		
18						
19	Res	ults		Eval		
20	Cholesterol (LDL):	100	mg/dL	Total Cholesterol:	: mg/d	
21	Cholesterol (HDL):	69	mg/dL	%HDL:		
22						
23	Refer	ence Charts:		BMI Lev		
24				kg/m <sup>2</sup>	Category	
25				0	Underweight	
26				18.5	Normal	
27				25	Overweight	
28						
29	Blood Pressure					
30		S	ystolic,	Diastolic	Category	
31		<120	&	<80	Normal	
32		≥120	or	≥80	Elevated	
33		>140	&	>90	High	

1 In cell **E3**, use an appropriate function to show the current date.

- [1]
- 2 In cell E12, enter a formula to calculate the BMI value. The BMI value is calculated by taking the weight of the patient, divided by the square of the patient's height. This value must be shown in 1 decimal place. [2]
- 3 In cell **E13**, use an appropriate function to search for BMI category based on the reference chart at the bottom of the spreadsheet. [1]
- 4 In cell **E15**, use a conditional statement to identify the blood pressure category based on the patient's systolic and diastolic value. The categories can be found in the reference chart at the bottom of the spreadsheet. [2]

- 5 In cell **E20**, use an appropriate function to calculate the total cholesterol. The total cholesterol consists of both the LDL and the HDL [1]
- 6 In cell **E21**, enter a formula to calculate percentage of HDL cholesterol in the total cholesterol. This value must be shown as a percentage in **whole numbers**. [2]
- 7 In cell **E21**, apply conditional formatting such that the font color turns red when the cell value is less than 50 percent. [1]

The following program prints out the factors of a positive integer, other than 1 and itself.

```
number = int(input("Enter a positive integer: "))
```

```
print("Factors of this integer are: ")
```

for factor in range(2, number):

```
if number % factor == 0:
```

print(factor)

Open the file **FINDFACTOR.py** 

Save the file as **FINDFACTOR**\_<your name>\_<class>\_<index number>.py

- 8 Edit the program so that it:
  - (a) Prints out the message "There are a total of X factors." at the end of the program, where X represents the total number of factors. [2]
  - (b) Checks if the input is a positive integer. If it is not, ask the user for input again as necessary. [2]

Save your program.

9 Save your program as FINDMULTI\_<your name>\_<class>\_<index number>.py

Edit the program such that it:

(a) Prints out each factor multiplied by its corresponding factor instead. [3]

For example, if the input is 12, the output should be:  $2 \times 6$  $3 \times 4$ 

(b) Has no repetitions in the output. For example, the outputs 2×6 and 6×2 are considered as the same. [3]

Save your program.

No part of the paper is to be reproduced without the approval of the Principal of Temasek Secondary School.

The following program runs on a vending machine that sells drinks. The program does the following:

- Allows the user to select a drink to order.
- Accepts cash as payment. Accepted denominations are 10 cents, 20 cents, 50 cents, \$1, \$2, \$5 and \$10.
- Calculates the amount of money to be returned if
  - the user paid more than the cost of the drink, or
  - the user decides to cancel the order.

There are several syntax errors and logical errors in the program.

```
print(" ~~Drinks Menu~~ ")
drinks=["Hot Coffee", "Hot Tea", "Canned Drink", "Bottled Drink"]
prices=[4,3,2.5,3.5]
for option in range(drinks):
    print(str(option+1)+"-"+drinks[option]+"\t$"+prices[option]))
choice=input("\nEnter your choice: ")
owe=prices[choice]
print("\n"+drinks[choice]+" costs $"+str(owe)+".")
valid=["0.1","0.2","0.5","1","2","5","10"]
while paid>owe:
    cash=input("Please insert cash(\"x\" to cancel): ")
    if cash in valid:
        paid+=int(cash)
    if paid=="x":
        print("Your order is cancelled.")
        owe=0
    if not cash.isdigit():
        print("Invalid currency!")
    print("\nBalance payment is $"+str(owe-paid)+".")
if paid>owe:
    print("\n$"+str(paid-owe)+" has been returned to you.")
if cash!="x":
    print("Your "+drinks[choice]+" has been served. Enjoy!")
```

Open the file VENDDRINKS.py

Save the file as MYVENDING\_<your name>\_<class>\_<index number>.py

**10** Identify and correct the errors in the program so that it works according to the description given. [10]

Save your program.

[Turn over

The modern day calendar used worldwide is known as the Gregorian Calendar. It was started on 15 October 1582.

The Gregorian Calendar has 12 months and each month has different number of days:

- January 31 days
- February 28 days (29 days during a leap year)
- March 31 days
- April 30 days
- May 31 days
- June 30 days
- July 31 days
- August 31 days
- September 30 days
- October 31 days
- November 30 days
- December 31 days

Each year in the Gregorian Calendar has 365 days, except for leap years which have 366 days. A year that is exactly divisible by 4 is a leap year, except for those divisible by 100. For years that are divisible by 100, only those divisible by 400 are leap years.

You have been asked to create a program that identifies leap years in the Gregorian Calendar and determine if the input date is valid or not.

The program should allow you to:

- Enter a date in the DD/MM/YYYY format.
- Check that the input meets the format such that D, M and Y must be digits. If the input does not match the format, ask the user to input again
- Determine if the input year is a leap year.
- Hence, check if the combination of the input year, month and day makes up a valid date.
- Output "This date is valid." if the combination is a valid date, otherwise output "This date is not valid."
- 11 Write your program and test that it works.

Save your program as **VALIDDATE** \_<your name>\_<class>\_<index number>.py

12 When your program is complete, use today's date as the test data.

Take a screenshot of the outputs displayed on the screen and save as: [2] **TESTTODAY\_**<your name>\_<class>\_<index number>

Save your files in either .png or .jpg format.

[8]

**13** Save your program as **COUNTLEAP**\_<your name>\_<class>\_<index number>.py

Extend your program to count the number of leap years that occurred before the input year, since year 1582. Your program should output this result with an appropriate statement. [4]

Save your program.

- 14 Each week of the Gregorian Calendar has 7 days, with the order as shown:
  - Monday
  - Tuesday
  - Wednesday
  - Thursday
  - Friday
  - Saturday
  - Sunday

Save your program as **WEEKDAY**\_<your name>\_<class>\_<index number>.py

Extend your program to determine the day of the week from the input date, given that 31 December 1582 is a Friday. Your program should output this result with an appropriate statement. [6]

Save your program.

### **END OF PAPER**

# QUICK REFERENCE FOR PYTHON

### 1 Identifiers

When naming variables, functions and modules, the following rules must be observed:

• Names should begin with character 'a'-'z' or 'A'-'Z' or '

and followed by alphanumeric characters or '\_'.

- Reserved words should not be used.
- User-defined identifiers are case sensitive.

# 2 Comments and Documentation Strings

# This is a comment

.....

This is a documentation string over multiple lines

.....

# 3 Input/Output

print ("This is a string")

s = input ("Instructions to prompt for data entry.")

### 4 Import

import <module>

e.g. import math

### 5 Data Type

Data Type	Notes
int	integer
float	real number
bool	boolean
str	string (immutable)
list	Series of values

### 6 Assignment

Assignment Statement	Notes
a = 1	integer
b = c	variable
d = "this is a string"	string
mylist = [1, 2, 3, 4,5]	list or array

# 7 Arithmetic Operators

Notes
plus, subtract
multiply, divide
remainder or modulus
exponential or power
quotient of the floor division

### 8 Relational Operators

Operator	Notes
==	equality
!=	not equal to
> >=	greater than, greater than or equal to
< <=	less than, less than or equal to

### 9 Boolean Expression

<b>Boolean Expression</b>	Notes
a and b	logical and
a or b	logical or
not a	logical not

### 10 Iteration

while loop	for loop		
while	for i in range(n):		
conditions(s):	<statement(s)></statement(s)>		
<statement(s)></statement(s)>	for record in records:		
	<statement(s)></statement(s)>		

#### 11 Selection

Туре 1	Туре 2	Туре 3
if condition(s): <statement(s)></statement(s)>	<pre>if condition(s):     <statement(s)> else:     <statement(s)></statement(s)></statement(s)></pre>	<pre>if condition(s):     <statement(s)> elif condition(s):     <statement(s)> else:     <statement(s)></statement(s)></statement(s)></statement(s)></pre>

#### 12 Built-in Functions

#### (a) Basic Functions

abs()	chr()	float()	input()	int()	
ord()	print()	range()	round()	str()	
format()					

# (a) Mathematical Functions

ceil()	exp()	fabs()	floor()	log()	
max()	min( )	pow( )	sqrt()	trunc()	

## (a) String Functions

endswith()	find()	isalnum()	isalpha()	isdigit()
islower()	isspace()	isupper()	len()	lower()
startswith()	upper()			

### 13 Reserved Words

Reserved words cannot be used as identifiers. They are part of the syntax of the language.

False	None	True	and	as
assert	break	class	continue	def
del	elif	else	except	finally
for	from	global	if	import
in	is	lambda	nonlocal	not try
or	path	raise	return	
while	with	yield		