[NAME		CLASS	INDEX NO.		
SPS S75	ST. PATRICK'S SCHOOL PRELIMINARY EXAMINATION 2022				
SUBJECT :	Computing Paper 2 (7155/02)	DATE	: 22 AUG 2022		
LEVEL :	Secondary 4 Express	DURATION	: 2 hours 30 minutes		
ADDITIONAL MATERIALS	Electronic version of APPL Electronic version of BLOO Electronic version of HASH Insert Quick Reference Glo	ICATION.xlsx file DPRESSURE.py file TAG.py file ssary			

READ THESE INSTRUCTIONS FIRST

Write your name, register number and class at the top of this page.

Answer **all** questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

Programs are to be written in Python. Save your work using the file name given in the question as and when necessary.

The number of marks is given in brackets [] at the end of each question or part question. The total number of marks for this paper is 50.

For Examiner's Use		
Score	/50	

This paper contains 11 printed pages including this cover page.

Quick Reference for Python

This quick reference shows some examples of the Python language constructs. The complete Python language is not limited to these examples.

1. Identifiers

When naming functions, variables 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

Operator	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

Boolean Expression	Notes
a and b	logical and
a or b	logical or
not a	logical not

10. Iteration

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

11. Selection



12. Built-in Functions

(a) Basic functions

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

(b) Mathematical functions

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

(c) 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	Тгие	and	as
assert	break	class	continue	def
del	elif	else	except	finally
for	from	global	if	import
in	is	lambda	nonlocal	not
or	pass	raise	return	try
while	with	yield		

A government agency uses spreadsheet software to calculate the one-time cash payout to help applicants residing in the country tide through the Covid-19 pandemic. The spreadsheet provides the application reference number, monthly family income and number of family members in the applicant's household.

Each Application Reference Number in Column A comprises the following:

- 1. The first alphabet which represents the citizenship status of the applicant as follows:
 - "C" represents "Citizen"
 - "P" represents "Permanent Resident"
 - "F" represents "Foreigner"
- 2. The next 4 digits of the Application Reference Number represents the year that the applicant was born.
- 3. The last 3 digits are random.

You are required to finish setting up the spreadsheet to provide useful information and calculate the payout to each applicant.

	A	В	С	D	E	F	G
1			2020 Appli	ication for Covi	id-19 Crisis Fund	l	
2							
3							
	Application Ref.		olular and la	Monthly Family	Number of Family	Monthly Per Capita	Cost Down
4	No.	Age	Citizenship	Income	Members	Income	Cash Payout
5	C1950893			\$ 3,200	6		
6	C1959252			\$ 1,800	3		
7	P1971158			\$ 4,500	3		
8	C1999772			\$ 6,200	2		
9	C1947585			\$ 1,200	7		
10	C1949500			\$ 1,600	5		
11	C1978166			\$ 5,600	6		
12	C2006054			\$ 8,000	4		
13	F1988665			\$ 10,000	1		
14	P1944818			\$ 2,300	7		
15	C1980912			\$ 1,400	6		
16	C2005380			\$ 1,250	2		
17	C1977433			\$ 900	4		
18	F1959907			\$ 1,400	4		
19	F1990499			\$ 3,600	2		
20	P1971008			\$ 4,000	1		
21	C1983766			\$ 11,500	1		
22	F2006903			\$ 3,600	6		
23	C1959344			\$ 4,800	3		
24	C1978220			\$ 2,400	6		
25							
26							
27	Monthly Per Capit	a Income			<u>Pe</u>	Per Capita Income Level	
					Monthly Per Capita	Description	Payout
28	Median:				Income	Description	Tuyout
					\$0	Between \$0	
29	70th percentile:				֥	(inclusive) to \$3,100	\$ 800
					\$3,100	Between \$3,100	
30						(inclusive) to \$5,100	\$ 500
-					\$5,100	\$5,100 or more	
31						-	\$ 100

Open the file **APPLICATION.xIsx**. You will see the following data. Save the file as **PAYOUT**_<your name>_<centre number>_<index number>.xlsx

- 1 Enter a formula to calculate the current age of the applicants in the **Age** column at the time of application in the year 2020. [2]
- Use a conditional statement to identify the citizenship status (Citizen, Permanent Resident or Foreigner) in the Citizenship column. [2]
- **3** The monthly per capita income is the monthly family income divided by the number of family members in the applicant's household.

Enter a formula to calculate each applicant's **Monthly Per Capita Income**, rounded down to the nearest dollar(\$). [2]

4 All applicants except **Foreigner** citizenship status will receive the cash payout according to their **Monthly Per Capita Income**.

Use a conditional statement to identify an applicant's cash payout based on Citizenship status and search for the corresponding Payout in the Per Capital Income Level table to complete the Cash Payout column. [2]

- 5 In cell **B28** enter a formula to find the median monthly per capita income. [1]
- 6 In cell **B29** enter a formula to find the 70th percentile monthly per capita income. [1]

Save and close your file.

Blood pressure (BP) readings are expressed as a ratio of the systolic pressure over the diastolic pressure. The program below takes in the patient's systolic and diastolic pressure and outputs the diagnosed outcome of the patient.

```
systolic = int(input("Enter your systolic pressure (mmHg): "))
diastolic = int(input("Enter your diastolic pressure (mmHg): "))
cat = None
result = ["Normal BP", "High-normal BP"]
if systolic < 120 and diastolic < 80:
    cat = 0
if (systolic >= 120 and systolic <= 139) or (diastolic >= 80 and
diastolic <= 89):
    cat = 1
print("Diagnosis:", result[cat])
```

Open the file **BLOODPRESSURE.py**

Save the file as **MYBP**_<your name>_<centre number>_<index number>.py

7 Edit the program so that it prints the patient's blood pressure reading in the following form after the patient's data has been entered:

```
Enter your systolic pressure (mmHg): 120
Enter your diastolic pressure (mmHg): 75
Your BP is 120 / 75 mmHg.
```

Save your program.

[1]

8 Save your program as MYBP2_<your name>_<centre number>_<index number>.py

Categories for Blood Pressure Levels in Adults (Aged 18 Years and Older)								
	Blood Pressure Level (mmHg)							
Category	Systolic		Diastolic					
Normal BP	< 120	and	< 80					
High-Normal BP	120 - 139	or	80 - 89					
Stage 1 Hypertension	140 - 159	or	90 - 99					
Stage 2 Hypertension	160 or more	or	100 or more					
* Isolated Systolic Hypertension	> 140	and	< 90					

A more complete categorisation is shown in the table below.

When systolic and diastolic blood pressures fall into different categories, the higher category should be used to classify blood pressure level.

For example, 160/95 mmHg would be classified as stage 2 hypertension.

Edit the program so that it can also correctly output the additional 3 cases of hypertension.

Save your program.

9 Save your program as **MYBP3**_<your name>_<centre number>_<index number>.py

Edit the program so that it will request user to enter the number of sets of data to be entered and perform the categorisation for the entered number of times.

Your program should prompt the user with the following message:

"Enter the number of sets of data to be cateogorised: "

Save your program.

[7]

A program is used to process data relating to the number of hashtags linked to blog posts and the number of views of the posts in its first week of posting.

The hashtags of 5 posts are combined into a string and consolidated in a list. A second list consolidates the number of views of each post after 1 week. The post titles are consolidated in a third list.

The program eventually outputs

- 1) the post with the highest number of hashtags
- 2) the average number of hashtags among the 5 posts
- 3) the post with the most number of views

There are several syntax and logic errors in the program.

```
all hashes = ["#gogreen#recycling#upcycling",''
          "#coolgames#experiment#expertmode#awesomegames",''
          "#diycook#ironchef",''
          "#outdoor#fauna#flora#nature#scenery#sunset",''
          "#magic#howcanitbe"]
views = [230, 683, 388, 597, 127]
post titles = ["My Upcycling Project", ''
               "Stanley's Awesome Games", ''
               "Chef @ Home", ''
               "An afternoon at Bukit Timah Reserve", ''
               "Card Tricks"
num hashes = []
for i in range(5)
    num hashes += len(all hashes[i].split("#")) - 1
highest hash = max(num hashes)
index = None
for j in range(5):
    if num hashes[j] = highest hash:
        index = j
print(post titles[j], "has", highest hash, "hashtags which is the
highest among all posts.")
average = sum(num hashes) % 5
print("The average number of hashtags is", sum)
```

```
highest_view = max(view)
index = None
for k in range(4):
    if views[k] == highest_view:
        k = index
print(post_titles[index],"has",highest_view,"views which is the
highest among all posts.")
```

Open the file HASHTAG.py

Save the file as **MYHASHTAG_**<your name>_<centre number> <index number>.py

10 Identify and correct the errors in the program so that it works according to the rules given.

Save your program.

[10]

SpendMax shopping mall is having its annual lucky draw for all shoppers. Every shopper is entitled to a lucky draw chance. Shoppers have to enter the receipt number of their purchase followed by their mobile phone number. Each receipt is entitled to one chance.

Lucky draw outcomes include:

- 1) a 65-inch television (1 top prize only)
- 2) a \$50 shopping voucher (20 sets only)
- 3) a \$10 shopping voucher (50 sets only)
- 4) a recylable bag (no limit)

Develop a program that will prompt the user to enter his receipt number and mobile phone number.

Your program must check that the user enters

- 1) a valid receipt number (starting with the capital letters "DRAW" followed by a 5-digit number)
 - Eg. DRAW56789
- 2) a valid mobile phone number (8-digit number)

Your program should then generate a random lucky draw outcome from one of the four outcomes stated above. The number of prizes given out should be recorded to track their availability. In the event that the prize or vouchers have been given out, the program should randomly generate another lucky draw outcome until the chosen outcome is available.

Your output must look like this:

Enter your receipt number: DRAW12345 Enter your mobile phone number: 98765432 Congratulations! You have won a recylable bag

11 Write your program and test that it works.

Save your program as **MYDRAW**_<your name>_<centre number>_<index number>.**py**

12 When your program is working, use the following test data to show your test results:

Test Output 1: Enter your receipt number: DR12345 Enter your receipt number: DRAW12345 Enter your mobile phone number: 12349876 Congratulations! You have won a <depending on random prize>

Test Output 2: Enter your receipt number: DRAW23012 Enter your mobile phone number: 98123 Enter your mobile phone number: 98123456 Congratulations! You have won a <depending on random prize>

Take a screen shot of your results and save it as MYDRAW_<your name>_<centre number>_<index number>

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

[4]

13 Save your program as MYDRAW2_<your name>_<centre number>_<index number>.py

Extend your program to remind the customer to double-check his mobile number details if he has won the 65-inch television.

Your output must look like this:

Please double-check if your mobile number is 12345678. Enter 1 to confirm and 0 to re-input your mobile number: 1 You will be informed shortly through your mobile number.

Allow the user to re-enter his mobile number only once if he enters 0 and end the program.

Save your program.

[6]