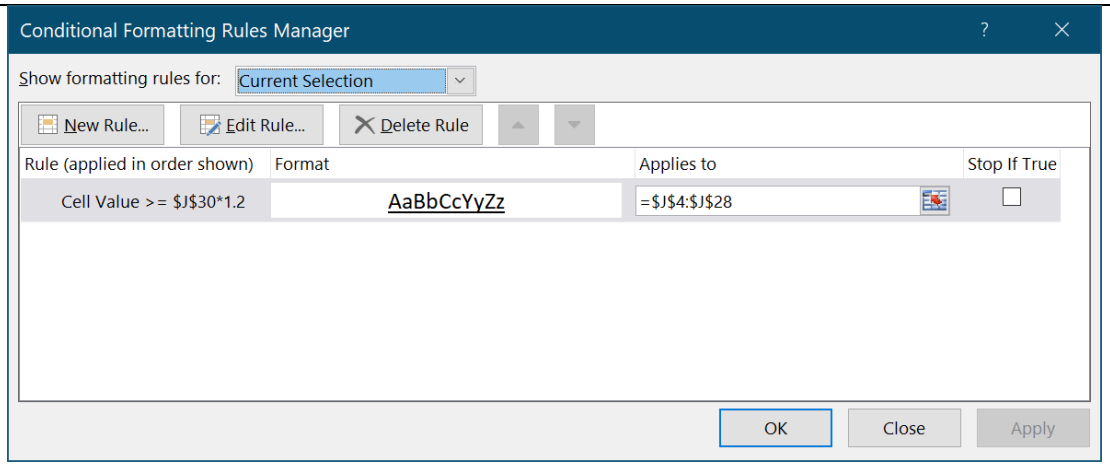
Sec 4 Computing Prelim 2021 P2 Ans Scheme

TASK 1

1	One mark: **D4**=VLOOKUP(C4,$L$4:$N$30,3,FALSE) One mark: Completing **D4:D28** and **F4:F28** columns	2m
2	**G4**=IF(LEN(G4)>0,VLOOKUP(G4,$L$4:$N$30,3,FALSE),0) Note: **G4:D28** must be completed appropriately	1m
3	One mark for logical condition, one mark for calculation: **I4**=IF(AND(LEFT(C4)<>LEFT(E4),LEFT(G4)<>LEFT(E4),LEFT(C4)<>LEFT(G4),LEN(G4)>0) ,0.1*SUM(D4,F4,H4),0) Note: **I4:I28** must be completed appropriately	2m
4	**J4**=SUM(D4,F4,H4)-I4 Note: **J4:J28** must be completed appropriately	2m
5	One mark: **O4**=COUNTIF($C$4:$H$28,L4) One mark: Completing **O4:O28** column	1m
6	**J30**=AVERAGE(J4:J28)	2m
7		1m

TASK 2

8a	`print("The solutions to this quadratic equation are {} and {}.".format(sol1,sol2))`	1m

8b	`if sol1==sol2:`
	`  print("The only solution to this quadratic equation is {}.".format(sol1)`
	`else:`
	`  print("The solutions to this quadratic equation are {} and {}".format(sol1,sol2))`
	
	1 mark for checking if-else, 1 mark for appropriate output
		2m

8c	`if (b**2-4*a*c)<0:`
	`  print("There are no real solutions to this quadratic equation.")`
	`else:`
	`  sol1 = (-b+(b**2-4*a*c)**0.5)/(2*a)`
	`  sol2 = (-b-(b**2-4*a*c)**0.5)/(2*a)`
	`  if sol1==sol2:`
	`  print("The only solution to this quadratic equation is {}.".format(sol1))`
	`  else:`
	`  print("The solutions to this quadratic equation are {} and {}".format(sol1,sol2))`
	
	1 mark for checking if-else, 1 mark for appropriate output
		2m

9a	`while confirm == "N":`
	`  :`
	`  confirm = input("The quadratic equation is {}x^2 + {}x + C = 0, correct? (Y/N):".format(a,b,c))`
	`  if confirm == "N":`
	`  print("Please re-enter the values again.")`
	
	1 mark for loop control, 1 mark for formatting of input prompt and output
		2m

9b	1 mark for each variable **a**, **b** and **c**, i.e. checking and correcting the equation
	`if a == 1:`
	`  eqn = "x^2 "`
	`elif a == -1:`
	`  eqn = "-x^2 "`
	`else:`
	`  eqn = str(a) + "x^2 "`
	
	`if b == 1:`
	`  eqn = eqn + "+ x "`
	`elif b == -1:`
	`  eqn = eqn + "- x "`
	`elif b < 0:`
	`  eqn = eqn + "- " + str(-b) + "x "`
	`else:`
	`  eqn = eqn + "+ " + str(b) + "x "`
	
	`if c < 0:`
	`  eqn = eqn + "- " + str(-c)`
	`else:`
	`  eqn = eqn + "+ " + str(c)`
	
	`confirm = input("The quadratic equation is {}, correct? (Y/N):".format(eqn))`
		3m

TASK 3

10	**One mark each:** ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩	10m

```
age = 2021 - int(input("Enter your year of birth: "))
allergy = input("Do you have any drug allergies? (Y/N): ")
eligible_with_appt = []
eligible_without_appt = []
if age >= 60:
    if allergy == "N":
        eligible_without_appt = ["Pfizer-BioNTech"]
        eligible_without_appt += ["Moderna"]
    else:
        eligible_without_appt = ["Novavax"]
elif age >= 18:
    if allergy == "N":
        eligible_with_appt = ["Pfizer-BioNTech"]
        eligible_without_appt = ["Moderna"]
    else:
        eligible_with_appt = ["Novavax"]
elif age >= 12:
    if allergy == "N":
        eligible_with_appt = ["Pfizer-BioNTech"]
if len(eligible_without_appt) ==0 and len(eligible_with_appt) ==0:
    print("You are not eligible for any vaccination.")
else:
    print("You are eligible for:")
for vaccine in eligible_without_appt:
    print("- {} vaccine without appointment.".format(vaccine))
for vaccine in eligible_with_appt:
    print("- {} vaccine with an appointment.".format(vaccine))
```

TASK 4

<table>
<tr><td>11</td><td>One mark each:
- loop control (stop iteration only when input is empty)
- iteration to run through the input string
- checking for the four operators
- slicing each number and convert to float
- performing the correct operation on the two numbers

Sample code:
<pre>exp=input("Enter an expression: ")
while exp !="":
    for i in range(len(exp)):
        if exp[i]=="+":
            result=float(exp[:i])+float(exp[i+1:])
        elif exp[i]=="-":
            result=float(exp[:i])-float(exp[i+1:])
        elif exp[i]=="*":
            result=float(exp[:i])*float(exp[i+1:])
        elif exp[i]=="/":
            result=float(exp[:i])/float(exp[i+1:])
    print("="+str(format(result,'.10g')))
    exp=input("Enter an expression: ")</pre></td><td>5m</td></tr>
<tr><td>12</td><td>One mark each:
- input with appropriate prompt
- output with correct formatting applied

Sample Output:
<pre>Enter an expression: 0.1+0.2
=0.3
Enter an expression: 3-2
=1
Enter an expression: 0.5*6
=3
Enter an expression: 9/3.0
=3
Enter an expression:
>>></pre></td><td>2m</td></tr>
<tr><td>13</td><td>One mark each:
- checking for zero after division
- no calculation is performed if division by zero
- output an appropriate statement

Sample Code:
<pre>elif exp[i]=="/":
    if float(exp[i+1:])==0:
        print("This expression cannot be evaluated.")
    else:
        result=float(exp[:i])/float(exp[i+1:])</pre></td><td>3m</td></tr>
</table>

14 (a)	**One mark each:** - checking for case of no operators - output the original expression	2m
14 (b)	**One mark each:** - slicing to separate the operators from numbers in the string OR iteration to determine the indices of the operators - slice numbers from the string (and store in a list) and convert to float - iteration to check and perform each operation correctly, from left to right - logic for operating on multiplication & division before addition & subtraction - keeping track of the results of sub-calculations using a new list or update number list - check for division by zero and stop calculation* - output original number if input has no operator* - correct output format, depending on whether division by zero *only awarded if the calculation logic for the 4 operators are correct **Sample Code:**	8m

```
exp=input("Enter an expression: ")
while exp !="":
    exp_list=[]
    last=0
    for i in range(len(exp)):
        if exp[i] in "+-*/":
            exp_list=exp_list+[float(exp[last:i])]+[exp[i]]
            last=i+1
    exp_list=exp_list+[float(exp[last:])]
    if ("*" in exp_list) or ("/" in exp_list):
        for i in range(len(exp_list)):
            if exp_list[i] == "*" or exp_list[i] == "/":
                if exp_list[i] == "*":
                    exp_list[i+1]=exp_list[i-1]*exp_list[i+1]
                elif exp_list[i] == "/":
                    if exp_list[i+1]==0:
                        exp_list=[]
                        break
                    else:
                        exp_list[i+1]=exp_list[i-1]/exp_list[i+1]
                exp_list[i-1]=None
                exp_list[i]=None
    exp_list2=[]
    for item in exp_list:
        if item != None:
            exp_list2=exp_list2+[item]
    for i in range(len(exp_list2)):
        if exp_list2[i] == "+" or exp_list2[i] == "-":
            if exp_list2[i] == "+":
                exp_list2[i+1]=exp_list2[i-1]+exp_list2[i+1]
            elif exp_list2[i] == "-":
                exp_list2[i+1]=exp_list2[i-1]-exp_list2[i+1]
            exp_list2[i-1]=None
            exp_list2[i]=None
    if exp_list2==[]:
```

```
            print("This expression cannot be evaluated.")
        else:
            print("="+format(exp_list2[-1],'.10g'))
        exp=input("Enter an expression: ")
```