2020 Computing Sec 4 Prelim Paper 1 Marking Guide

Qn	Answer
1	One mark for working top formula, one mark for rest
	=VLOOKUP(\$B2,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B3,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B4,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B5,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B6,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B7,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B8,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B9,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B10,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B11,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B12,\$A\$18:\$D\$23,4,TRUE)
	=VLOOKUP(\$B13,\$A\$18:\$D\$23,4,TRUE)
2	One mark for working top formula, one mark for rest
	=MIN(D2:F2)
	=MIN(D3:F3)
	=MIN(D4:F4)
	=MIN(D5:F5)
	=MIN(D6:F6)
	=MIN(D7:F7)
	=MIN(D8:F8)
	=MIN(D9:F9)
	=MIN(D10:F10)
	=MIN(D11:F11)
	=MIN(D12:F12)
	=MIN(D13:F13)
3	One mark for working top formula, one mark for rest
	=PMT(G2/12,12*C2,B2)
	=PMT(G3/12,12*C3,B3)
	=PMT(G4/12,12*C4,B4)
	=PMT(G5/12,12*C5,B5)
	=PMT(G6/12,12*C6,B6)
	=PMT(G7/12,12*C7,B7)
	=PMT(G8/12,12*C8,B8)
	=PMT(G9/12,12*C9,B9)
	=PMT(G10/12,12*C10,B10)
	=PMT(G11/12,12*C11,B11)
	=PMT(G12/12,12*C12,B12)
	=PMT(G13/12,12*C13,B13)
4	One mark for working top formula, one mark for rest
	=H2*C2*12
	=H3*C3*12
	=H4*C4*12
	=H5*C5*12
	=H6*C6*12

	=H7*C7*12
	=H8*C8*12
	=H9*C9*12
	=H10*C10*12
	=H11*C11*12
	=H12*C12*12
	-1112 C12 12 -112*C12*12
5	One mark for working ten formula, one mark for reat
5	One mark for working top formula, one mark for rest
	=12+B2
	=13+B3
	=14+B4
	=I5+B5
	=16+B6
	=I7+B7
	=I8+B8
	=I9+B9
	=I10+B10
	=l11+B11
	=I12+B12
	=I13+B13
6a	ans = random.randint(1,100)
6b	count = 1
	count += 1
	print(count)
6c	<pre>x = input("Enter a number: ")</pre>
	<pre>while not x.isdigit():</pre>
	<pre>x = input("Enter only numbers: ")</pre>
	x=int(x)
7	while x != ans and count < 8: #1 mark correct loop
	<pre>11 x == ans: #1 mark correct message</pre>
	print("Correct!")
	erse:
0	Single errors are underlined, errors requiring two corrections are in hold
0	Single errors are underlined, errors requiring two corrections are in bold
	try1 = int(input <mark>(E</mark> nter distance for Attempt 1: "))
	<pre>try1 = int(input("Enter distance for Attempt 2: "))</pre>
	if trv1 < trv2:
	dis = try1
	elif:
	ais <u>==</u> tryz
	if dis <u>>=</u> 245:
	print("5 Points")
	print("4 Points")
	elif dis >= 226:
	print("3 Points <mark>"</mark> elif dis >= 216:
	print("2 Points")
	elif dis >= 206:

	print("1 Points")
	else print ("6 Points")
	trv1 = int(input("Enter distance for Attempt 1: "))
	try2 = int(input("Enter distance for Attempt 2: "))
	if try1 > try2:
	else:
	dis = trv2
	if dis > 245:
	elif dis >= 236:
	print("3 Points")
	else:
	print("0 Points")
9	List set up for World.
_	Assign 'A' to first space (index 0)
	Use of loop to print world
	with proper control to print four correct spaces in each of four lines
	Note: Also accept use of string as data structure for world.
	world = ["."]*16
	world[U] = 'A'
	<pre>for i in range(len(world)//4):</pre>
	line = ""
	for j in range(4):
	<pre>line += world[1^4 +]] print(line)</pre>
	print()
10	Input from user
	Variable set up for current
	Use of loop to iterate user input string with proper loop control
	Correct update of all spaces
	Correct printout
	world = ["."]*16
	<pre>for i in range(len(world)//4):</pre>
	line = ""
	<pre>ior j in range(4): line t= world[i*4 + j]</pre>
	print (line)
	print()
	current = 0
	print()
	for i in sequence:
	if i == 'w':
	elif i == 's':
	current += 4
	elif i == 'a':
	current -= 1
	$\begin{array}{c} \text{current} += 1 \end{array}$
	world[current] = 'A'
	<pre>for i in range(len(world)//4):</pre>

```
line = ""
          for j in range(4):
              line += world[i*4 + j]
          print(line)
      print()
11
     Α...
     Α...
     AAAA
      ...A
      output matches stored program
12
     Input from user for Player B
      Variable set up for current b
      Correct use of second loop to iterate Player B's sequence
      Check for spaces visited by both players
      world = ["."]*16
      world[0], world[15] = 'A', 'B'
      for i in range(len(world)//4):
          line = ""
          for j in range(4):
              line += world[i*4 + j]
          print(line)
      sequence_a = input("Enter sequence for Player A: ")
      sequence_b = input("Enter sequence for Player B: ")
      current a, current b = 0, 15
      for i in sequence a:
          if i == 'w':
              current_a -= 4
          elif i == 's':
              current_a += 4
          elif i == 'a':
              current a -= 1
          elif i == 'd':
              current a += 1
          if world[current a] == 'B' or world[current a] == 'X':
              world[current a] = 'X'
          else:
              world[current a] = 'A'
      for i in sequence b:
          if i == 'w':
              current b -= 4
          elif i == 's':
              current b += 4
          elif i == 'a':
              current b -= 1
          elif i == 'd':
              current b += 1
          if world[current b] == 'A' or world[current b] == 'X':
              world[current b] = 'X'
          else:
              world[current b] = 'B'
      for i in range(len(world)//4):
          line = ""
          for j in range(4):
              line += world[i*4 + j]
          print(line)
```

```
13
     Input from user for size of world
     Adaption of all other parts of program
     side = int(input("Enter length of grid"))
     world = ["."]*side**2
     world[0],world[side**2 - 1] = 'A','B'
     for i in range(len(world)//side):
         line = ""
         for j in range(side):
              line += world[i*side + j]
         print(line)
     sequence a = input("Enter sequence for Player A: ")
     sequence_b = input("Enter sequence for Player B: ")
     current a, current b = 0, side**2 - 1
     for i in sequence a:
         if i == 'w':
             current a -= side
         elif i == 's':
             current a += side
         elif i == 'a':
             current a -= 1
         elif i == 'd':
             current a += 1
         if world[current a] == 'B' or world[current a] == 'X':
              world[current a] = 'X'
         else:
             world[current a] = 'A'
     for i in sequence b:
         if i == 'w':
             current b -= side
         elif i == 's':
             current b += side
         elif i == 'a':
             current b -= 1
         elif i == 'd':
             current b += 1
         if world[current b] == 'A' or world[current b] == 'X':
             world[current b] = 'X'
         else:
              world[current b] = 'B'
     for i in range(len(world)//side):
         line = ""
          for j in range(side):
              line += world[i*side + j]
         print(line)
```