1. Write a pseudocode algorithm using a **for … next** loop to read five lowercase letters   
and output the largest and smallest. (a is less than b). [6]

a = ""

-- for i = 1 to 5 do

--     if i is lowercase then

--         a = a + i

--     if i > previous(i) then

--         a[previous(i)], a[i] = a[i], a[previous(i)]

--     next

-- print(a[0], a[length(a)])

2. Write a pseudocode algorithm that asks a user for a password. They are allowed three attempts to type the correct password, which is “Tues1212”.

If they type the correct password, output “Password accepted”, otherwise output   
“Password rejected”. [6]

end = false

-- while end == false do

--     password = input("enter the password")

--     if password == "Tues1212" then

--         print("password accepted")

--         break

--     else:

--         print("try again")

--         next

# 3. (a) Complete the trace table below with the values supplied. [4]

sunshine = 0

maxHours = 0

minHours = 100

totalSunshine = 0

repeat

sunshine = input(“Input hours of sunshine: ”)

if sunshine > maxHours then

maxHours = sunshine

endif

if sunshine < minHours then

minHours = sunshine

endif

totalSunshine = totalSunshine + sunshine

until sunshine = -1

print(“Max sunshine hours: ”, maxHours)

print(“Min sunshine hours: ”, minHours)

print(“Total sunshine hours: ”, totalSunshine)

Test Data: 2 7 3 8 -1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sunshine** | **maxHours** | **minHours** | **totalSunshine** | **Output** |
| 0 | 0 | 100 | 0 | Max sunshine hours: 0  Min sunshine hours: 0  Total sunshine hours: 0 |
| 2 | 2 | 2 | 2 | 2, 2, 2 |
| 7 | 7 | 2 | 9 | 7, 2, 9 |
| 3 | 7 | 2 | 12 | 7, 2, 12 |
| 8 | 8 | 2 | 20 | 8, 2, 20 |
| -1 | 8 | -1 | 19 | 8, -1, 19 |
|  |  |  |  |  |

# (b) What is the problem with the algorithm above? [2]

The until sunshine = -1 line comes too late since the min hours comparison has already been run, meaning we now have a negative value in our minHours variable, which is a problem because you cant have a negative number of hours.

# (c) This time the algorithm uses an entry condition WHILE loop.

# Complete the trace table to see the difference between the two. [2]

sunshine = 0

maxHours = 0

minHours = 100

totalSunshine = 0

sunshine = input(“Input hours of sunshine: ”)

while sunshine <> -1

if sunshine > maxHours then

maxHours =sunshine

endif

if sunshine < minHours then

minHours = sunshine

endif

totalSunshine =sunshine + totalSunshine

sunshine =INPUT

endwhile

print(“Max sunshine hours: ”, maxHours)

print(“Min sunshine hours: ”, minHours)

print(“Total sunshine hours: ”, totalSunshine)

Input data: 2 7 3 8 -1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sunshine** | **maxHours** | **minHours** | **totalSunshine** | **Output** |
| 0 | 0 | 100 | 0 | Max sunshine hours: 0  Min sunshine hours: 100  Total sunshine hours: 0 |
| 2 | 2 | 2 | 2 | 2, 2, 2 |
| 7 | 7 | 2 | 9 | 7, 2, 9 |
| 3 | 7 | 2 | 12 | 7, 2, 12 |
| 8 | 8 | 2 | 20 | 8, 2, 20 |
| -1 | 8 | 2 | 20 | 8, 2, 20 |
|  |  |  |  |  |

[Total 20 Marks]