NANYANG TECHNOLOGICAL UNIVERSITY

Suggested Solutions

MH1401/CY1401 - Algorithms and Computing I

NOTE:

1. The following paper has been converted from MATLAB to Python.

Contents

QUESTION 1. **(28 marks)**

(a)
```
x=-1
while(x<=0):
    x=int(input("Input a positive integer"))
```

(b) 0

(c) 18

(d)
```
if x < -3 or x >= 3:
    y=f(x,1)
else:
    if x < 0:
        y=f(x,3)
    else:
        if x == 0:
            y=f(x,4)
        else:
            y=f(x,2)
```

QUESTION 2. **(24 marks)**

(i)
```
def income_tax(income):
    if income <= 20000:
        return 0
    elif income <= 40000:
        return 0.05 * income
    elif income <= 100000:
        return 0.1 * income
    elif income <= 200000:
        return 0.15 * income
    else
        return 0.2 * income
```

(ii)
```
def income_tax_sg(income):
    if income <= 20000:
        return 0
    elif income <= 40000:
        return 0.05 * (income-20000)
    elif income <= 100000:
        return 0.05 * 20000 + 0.1 * (income - 60000)
    elif income <= 200000:
        return 0.05 * 20000 + 0.1 * 60000 + 0.15 * \
```

```
            (income - 100000)
        else
            return 0.05 * 20000 + 0.1 * 60000 + 0.15 * 100000 + \
            0.2 * (income - 200000)

    # '\' is a newline character that continues the previous line
```

QUESTION 3. (10 marks)

(i)
```
def newton_sqrt(X,n):
    if X < 0 or n <= 0:
        return -1;
    if n == 1:
        R1 = 10;
        return R1;
    else:
        out = newton_sqrt(X,n-1)
        Rx = 0.5 * (out + X) / out
        return Rx;

# Note that if they did not ask for recursion,
# we can use the following for loop instead:

# Non-recursion method:
def newton_sqrt(X,n):
    R1 = 10
    for i in range(1,n+1):
        if i-1 == 0:
            Rx = R1
        else:
            Rx = 0.5 * (R1+x)/R1
            R1 = Rx
    return(Rx)
```

(ii)
```
def newton_sqrt_approx(X,a):
    import math
    i = 0;
    dist=a+1;
    while (i <= 0 or dist > a):
        i+=1;
        dist = abs(math.sqrt(X)-(newton_sqrt(X,i)))
    print("Requires %d steps, distance = %.6f" % (i,dist))
```

QUESTION 4. (10 marks)

(i)
```
def check_move(x,y):
    if x[-1] > y[-1] or x == []:
        return -1
    else:
        return 0
```

(ii)
```
def check_victory(x):
    for i in range(4):
        if x[i] <= x[i+1]:
            return -1
    return 0
```

Suggested Solutions (Brandon)

Suggested Solutions (Camille)