

Algorithms and Python practical exercises Feb 2017

TASK 1

1. Write the algorithm the question asks you to.
2. Write this program in Python and check it works

eTeifi Books sell ebooks via the Internet. A customer is awarded a bonus voucher which they are able to use to save money off a future purchase if they have bought:

- more than ten ebooks
- or
- more than five ebooks with a total value of more than £50.00

Write an algorithm with inputs `NumbereBooksBought` and `TotalValueeBooksBought` and outputs either "Award bonus" or "Do not award bonus".

For example, with inputs

6 60

The output would be "**Award bonus**".

PROCEDURE VOUCHER

Award = FALSE

Input `NumbereBooksBrought`

Input `TotalValueeBooksBorught`

IF (`NumbereBooksBrought`) > 10 OR (`TotalValueeBooksBorught` > 50.00 AND `NumbereBooksBrought` > 5) THEN

 Award = TRUE

END IF

IF award == TRUE THEN

 print("Award bonus")

ELSE:

 print("Do not award bonus")

END IF

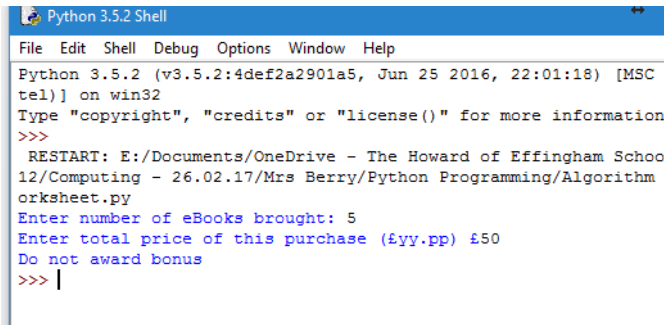
END PROCEDURE

```
def proc_award():
    award = False
    eBooks = int(input("Enter number of eBooks brought: "))
    price = float(input("Enter total price of this purchase (£yy.pp) £"))

    if (eBooks > 5 and price > 50.00) or (eBooks > 10):
        award = True

    if award == True:
        print("Award bonus")
    else:
        print("Do not award bonus")

proc_award() #runs award program
```



>>> RESTART: E:/Documents/OneDrive - The Howard of Effingham School/Computing - 26.02.17/Mrs Berry/Python Programming/Algorithm orksheet.py

Enter number of eBooks brought: 67

Enter total price of this purchase (£yy.pp) £1008

Award bonus

>>> |

TASK 2

1. Write the algorithm the question asks you to.
2. Write this program in Python and check it works

A diving competition calculates the final mark for each dive based on the marks of six judges. Each judge awards a mark individually (up to a maximum mark of 6.0).

The highest mark and the lowest mark are recorded but **not used** to calculate the final mark. The final mark is calculated by adding the four remaining marks together and dividing by four.

Using a basic text editor, write an algorithm, which inputs six judges' marks and outputs the lowest mark, the highest mark and the final mark. Save your completed algorithm as DivingAlgorithm.txt

For example, with inputs:

5.9
6.0
5.9
5.7
5.6
5.7

The output would be:

Highest:
6.0

Lowest:
5.6

Final Mark:
5.8

PROGRAM DivingCompetition

Judges = [] * 6

For i = 0 to length(Judges) - 1:

REPEAT

Judges[i] = int(input("Enter a score between 0 and 6"))

Until (Judges[i] <= 6) and (Judges[i] >= 0)

NEXT i

complete = FALSE

WHILE complete == FALSE:

complete = TRUE

For i = 0 to length(Judges) - 2:

IF Judges[i+1] < Judges[i] THEN

temp = Judges[i]

Judges[i] = Judges[i+1]

Judges[i+1] = temp

complete = FALSE

END IF

NEXT i

END WHILE

Print("Highest: " + str(Judges[length(Judges)-1]))

Judges.remove[length(Judges)-1]

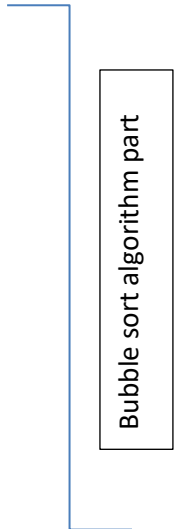
Print("Lowest: " + str(Judges[0]))

Judges.remove[0]

mark = (sum(Judges[])) / 4

print("Final mark: " + str(mark))

END PROGRAM



```

Enter a score between 0 and 6: 4
Enter a score between 0 and 6: 5
Enter a score between 0 and 6: 6
Enter a score between 0 and 6: 5
Enter a score between 0 and 6: 4
Enter a score between 0 and 6: 1
Highest: 6
Lowest: 1
Final mark: 4.5
...
    
```

```

#Diving Competition
def proc_diving():
    Judges = [0] * 6

    for i in range(0, len(Judges)):

        Judges[i] = -1
        while (Judges[i] > 6) or (Judges[i] < 0):
            Judges[i] = int(input("Enter a score between 0 and 6: "))
            #invalid range
            #error checking

        ###
        complete = False
        while complete == False:
            complete = True
            #Bubble sort

            for i in range (0, len(Judges) -1):
                if Judges[i+1] < Judges[i]:
                    temp = Judges[i]
                    Judges[i] = Judges[i+1]
                    Judges[i+1] = temp
                    complete = False

            ###

        print("Highest: " + str(Judges[len(Judges)-1]))

        Judges.remove(Judges[len(Judges)-1])
        print("Lowest: " + str(Judges[0]))
        Judges.remove(Judges[0])
        #removes last item (highest item) in list
        #removes 1st item (lowest item) in list

        mark = (sum(Judges)) / 4
        print("Final mark: " + str(mark))
        #Average of 4 remaining items
        #Final mark given
    
```

TASK 3

1. Write the algorithm the question asks you to.
2. Write this program in Python and check it works.

The manager of a mobile phone shop has many staff members. Each staff member's monthly sales of mobile phone contracts are recorded over a period of 12 months. If a staff member sells more than 4 mobile phone contracts in one month they are awarded a bonus.

Using a basic text editor, write an algorithm, which:

- inputs the number of staff members;
- inputs the number of monthly sales of mobile phone contracts for each staff member;
- outputs whether a staff member has a bonus for that month;
- calculates the total annual sales of each staff member;
- outputs the total annual sales for each staff member.

A partial example of the **input** and **output** of the algorithm is shown below.
(... indicates lines of input and output that are not shown)

```

Please enter the number of Staff members: 3
Enter month 1 figures for Staff member 1: 2
Enter month 2 figures for Staff member 1: 5
Bonus awarded.
Enter month 3 figures for Staff member 1: 3
...
...
Enter month 12 figures for Staff member 1: 2
Enter month 1 figures for Staff member 2: 3
...
...
Enter month 12 figures for Staff member 3: 3
Totals:
Staff member 1: 38
Staff member 2: 29
Staff member 3: 42
    
```

```
number_of_staff = int(input("Enter quantity of staff"))
```

```
monthly_sale = [[0 for x = 1 to 12] for y = 1 to number_of_staff ] #sets 2D list with staff number of lists containing 12 items
```

```
For i = 0 to number_of_staff -1:
```

```
    total = 0
```

```
    print("STAFF NUMBER: " + str(i) )
```

```
    For n = 0 to 11:
```

```
        monthly_sale [i][n] = Int(input("Enter net number of contracts sold during month " + n+1, ":"))
```

```
        IF monthly_sale [i][n] > 4 THEN
```

```
            print("Award bonus this month!")
```

```
        END IF
```

```
        total = total + monthly_sale [i][n]
```

```
    NEXT n
```

```
print("Total annual sales: ", total)
```

```
NEXT i
```

```
END
```

```
#Sales program
def proc_sales():
    number_of_staff = int(input("Enter quantity of staff? "))
    print("\n")

    monthly_sale = [[0 for x in range(12)] for y in range(number_of_staff)] #2d list containing staff number of lists each with 12 items/months filled with 0's

    for i in range (0,number_of_staff):
        print("STAFF MEMBER: " + str(i) + "\n")
        total = 0 #Annual total 0 to begin with

        for n in range (0,12):
            monthly_sale[i][n] = int(input("Enter net number of contracts sold during month " + str(n+1) + ": ")) #12 months per staff member #Sales each month
            if monthly_sale [i][n] > 4: #Award for sale >4 in 1 month
                print("Award bonus this month! \n")

            total = total + monthly_sale [i][n]
        print("Annual sales made this year: ", total, "\n") #Total no. of sales in 12 months

    proc_sales()

#ALTERNATIVE METHOD 1-D array
def proc_sales2():
    number_of_staff = int(input("Enter quantity of staff"))
    print("\n")
    monthly_sale = [0] * 12 #1D list of length 12 items

    for i in range (0,number_of_staff):
        print("STAFF MEMBER: " + str(i) + "\n")
        total = 0 #Annual total 0 to begin with

        for n in range (0,12):
            monthly_sale[i] = int(input("Enter net number of contracts sold during month ")) #12 months per staff member #Sales each month

            if monthly_sale[i] > 4: #Award for sale >4 in 1 month
                print("Award bonus this month!")

            total = total + monthly_sale [i]
        print("Annual sales made this year: ", total, "\n") #Total no. of sales in 12 months

    proc_sales2()
```

```
Enter quantity of staff? 2
|

Enter net number of contracts sold during month 0: 4
Enter net number of contracts sold during month 1: 4
Enter net number of contracts sold during month 2: 4
Enter net number of contracts sold during month 3: 4
Enter net number of contracts sold during month 4: 4
Enter net number of contracts sold during month 5: 4
Enter net number of contracts sold during month 6: 4
Enter net number of contracts sold during month 7: 4
Enter net number of contracts sold during month 8: 4
Enter net number of contracts sold during month 9: 4
Enter net number of contracts sold during month 10: 4
Enter net number of contracts sold during month 11: 4
Annual sales made this year: 48

Enter net number of contracts sold during month 0: 7
Award bonus this month!

Enter net number of contracts sold during month 1: 7
Award bonus this month!

Enter net number of contracts sold during month 2: 3
Enter net number of contracts sold during month 3: 3
Enter net number of contracts sold during month 4: 1
Enter net number of contracts sold during month 5: 1
Enter net number of contracts sold during month 6: 0
Enter net number of contracts sold during month 7: 0
Enter net number of contracts sold during month 8: 0
Enter net number of contracts sold during month 9: 0
Enter net number of contracts sold during month 10: 0
Enter net number of contracts sold during month 11: 4
Annual sales made this year: 26
```

TASK 4

Answer this 2 mark question

Below is a segment of an algorithm that determines if a number is present in a sorted array with 8 elements, and if present, at what position the number is located in the array.

DIV performs a division calculation, where the remainder is ignored.

e.g. $13 \text{ DIV } 4 = 3$ (remainder 1 is ignored)

```
Start = 0
End = 7
Found = True
Position = -1

input SearchValue
repeat
  set Middle = (Start - End) DIV 2

  if SearchValue = ThisArray(Middle) then
    set Found = True
    set Position = Middle
  endif

  if SearchValue > ThisArray(Middle) then
    set Start = Middle + 1
  endif

  if SearchValue < ThisArray(Middle) then
    set End = Middle - 1
  endif

until (Found = True) or (End < Start)
```

- (a) There are **two** errors in the algorithm. On the algorithm, identify and correct both errors. [2]

- =
1. Found should be set to = FALSE at the start of the program or the loop will not work correctly
 2. It should be Set MIDDLE = (END – START) DIV 2 so we are working with positive values (the target value will be positive and compared to positive middle).