The Sixth Form at George Abbot

'Academic excellence within a vibrant community.'

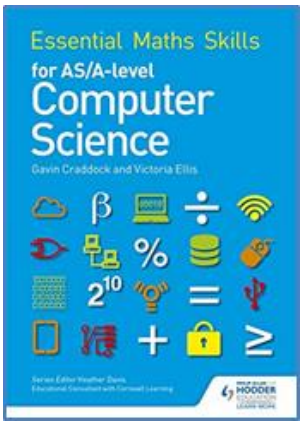
Subject: Computer Science
Head of Department: Mr Hunter jhunter@georgeabbot.surrey.sch.uk

Pre Sixth Form Tasks

Task	Detail	Demonstrated		
		Yes	Partially	No
Organisation	Prepare the following for storing lesson notes: • Large Ring-binder • Dividers • Plastic wallets • Multi-coloured pens • Highlighters • Ruler Complete the tasks below.			
Further Reading	**Books** *Trigger Happy: The inner life of videogames - Stephen Poole.* A witty, comprehensive and passionate discourse on the videogame explosion. Essential reading for anyone with an interest in this industry. *Accidental Empires – Robert X Cringely.* An insider's account of the origins and growth of the micro-computing industry from the earliest times to the present day; acerbic and funny in equal measure. **Magazines:** Wired New Scientist Websites: http://www.tnmoc.org/ http://pcpro.com http://www.theregister.co.uk/			
Additional task(s)	Complete the 'Computer Science – Skeleton Code Task' activity. Print out the questions below and answer them on the sheet. Bring the work to the first lesson.			

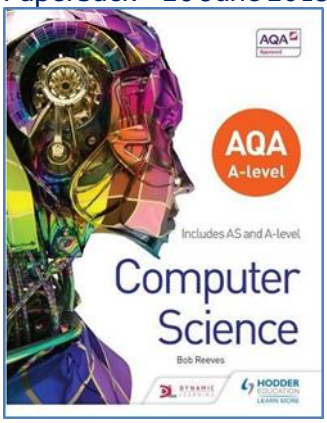
Suggested Reading List and Subject Resource

Essential Maths Skills for AS/A-Level Computer Science



Either of the following textbooks are beneficial to the course:

AQA A level Computer Science
Paperback – 26 June 2015



AQA AS and A Level Computer Science
Paperback – 24 Apr 2016

All tasks completed	**Yes**	**No**
Subject Teacher Signature		

Pre-Course Programming Task

INSTRUCTIONS:

Read the following preliminary material, which introduces the concept of the attached skeleton code.

Load the code into either Python/Thonny (Python), or other environment. Run the code, which will run a basic version of the program.

Answer the attached questions 1-4, which will test your understanding of your ability to interpret the respective skeleton code. Use the space provided to write your answers to each question.

THERE IS NO EXPECTATION TO WRITE ANY CODE IN THIS TASK.

PLANT GROWING SIMULATION – Preliminary Material

The **Skeleton Programs (Appendix A - Python)** accompanying this **Preliminary Material** are the programs for the simulation of plants growing.

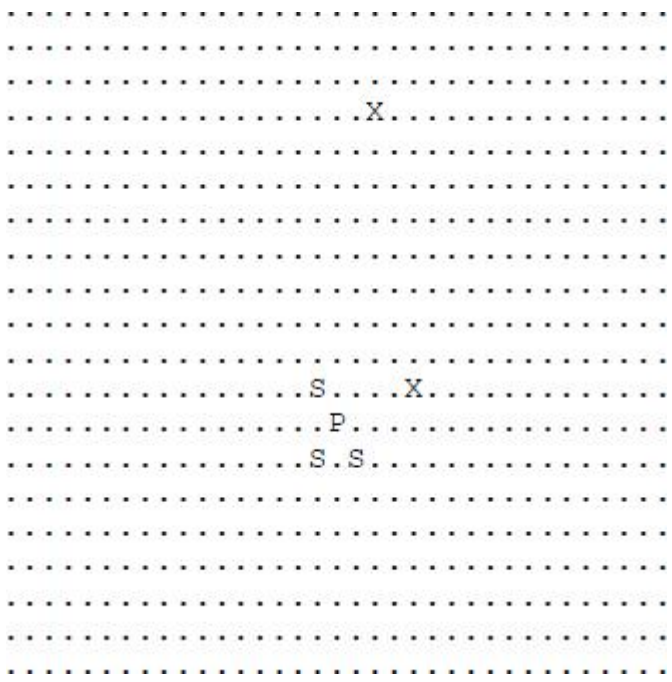
A plant scientist wants to use a computer to simulate how a specific plant will propagate over several years.

The field in which the plant is to grow and propagate is represented as a rectangular grid of cells. A cell can contain just soil, a plant, a seed or rock. It will always contain only one of these.

- If a cell contains just soil, then the cell is represented by '.'
- If a cell has a plant growing in it, the cell is represented by 'P'
- If a cell contains a seed, then the cell is represented by 'S'
- If a cell contains rock, then the cell is represented by 'X'

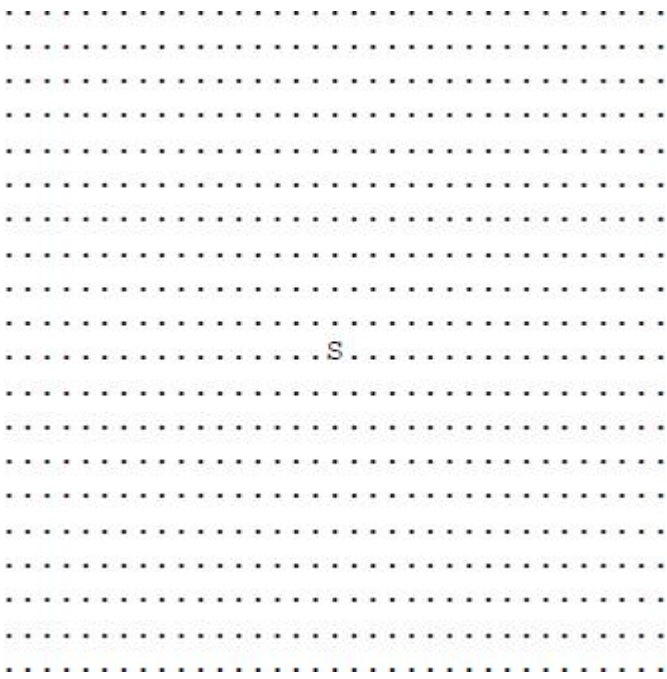
Figure 1 is an example of a field model.

Figure 1



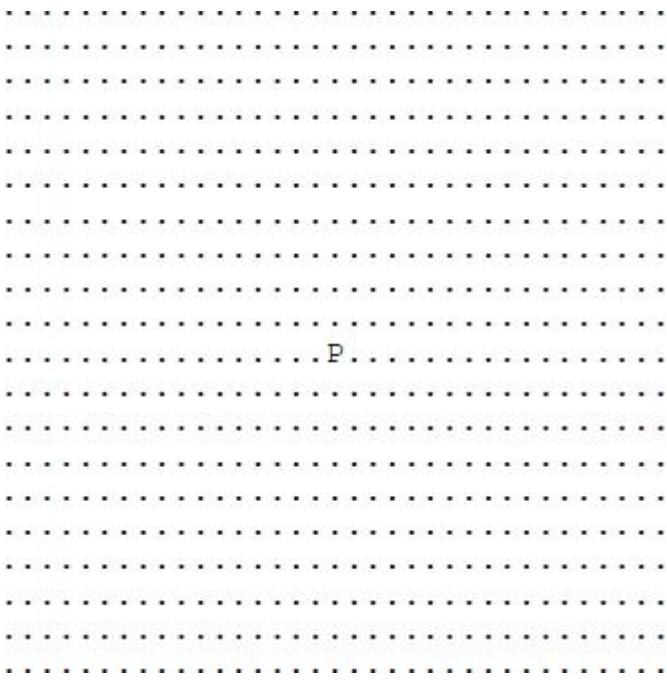
A new field starts with a seed in the middle of the field as shown in **Figure 2**.

Figure 2



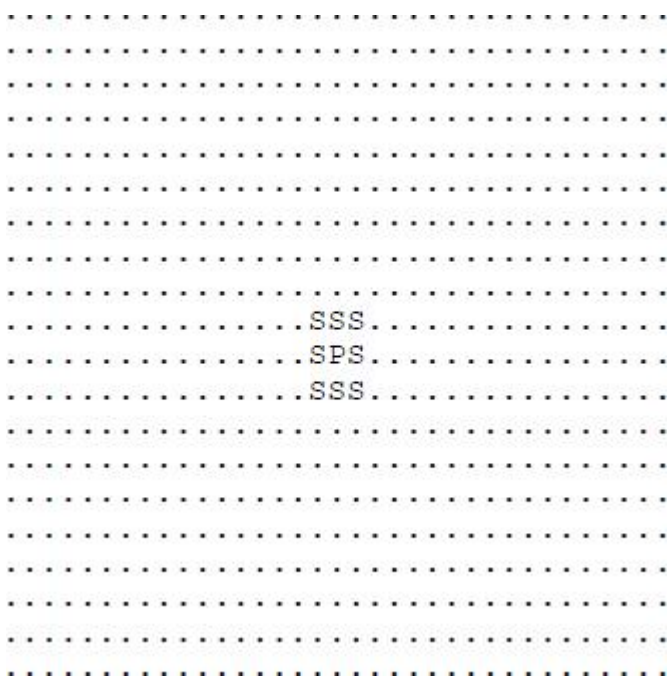
In the spring the seed germinates into a plant as shown in **Figure 3**.

Figure 3



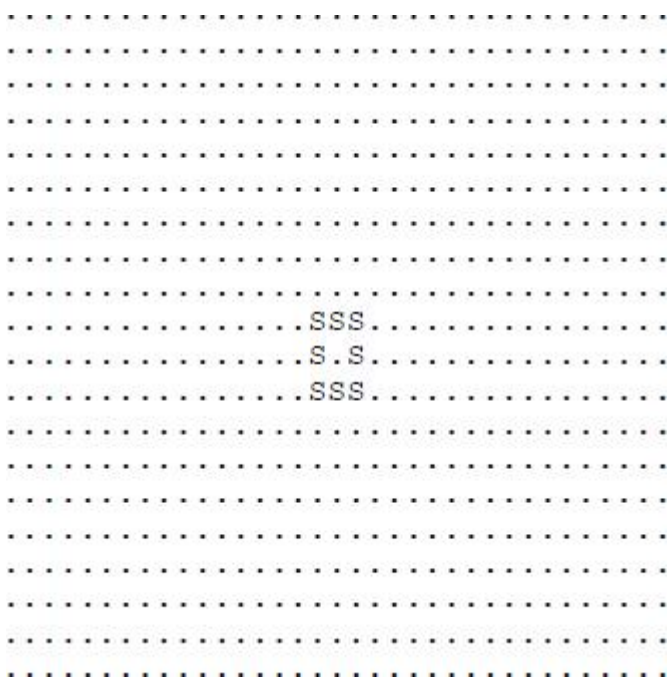
In the autumn the plant drops a seed in each cell immediately around the plant as shown in **Figure 4**.

Figure 4



In the winter the plant dies. This is represented by the cell content changing to a '.' as shown in **Figure 5**.

Figure 5

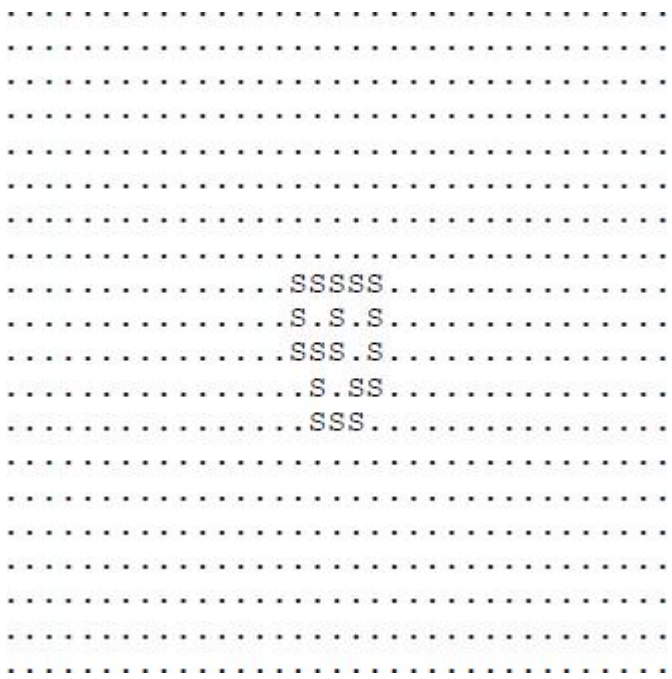


The seeds then lie dormant until spring when the cycle starts again and each seed germinates into a plant. In the spring a random frost may occur and kill off some of the plants. In the summer random rainfall patterns can result in a severe drought which also kills off some of the plants. In the autumn plants drop their seeds.

- If more than one seed lands in (drops into) a cell, only one seed survives.
- If there is a plant where a seed lands, the seed does not survive. The plant remains in the cell.
- If there is rock where a seed lands the seed does not survive. The rock remains in the cell.

At the end of year 2 the field contents may be as shown in **Figure 6**.

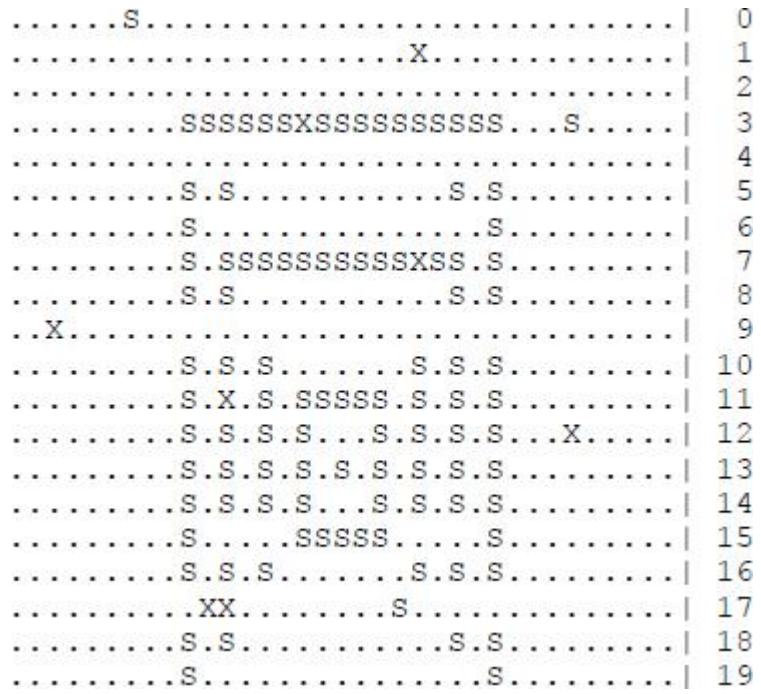
Figure 6



The **Skeleton Program** can use the **TestCase.txt Data File** to start the simulation with a different setup.

Figure 7 shows the contents of **TestCase.txt**.

Figure 7



The **Skeleton Program** allows the user to simulate plant growth and propagation for up to five years. There is also an option to step through the simulation a year at a time.

APPENDIX A

PYTHON SKELETON CODE

(Copy and paste into a Python environment, to run the basic version of the program, to help with the questions)

```
# Skeleton Program for the AQA A1 Summer 2017 examination
# this code should be used in conjunction with the Preliminary Material
# written by the AQA AS1 Programmer Team
# developed in a Python 3 environment

from random import *

SOIL = '.'
SEED = 'S'
PLANT = 'P'
ROCKS = 'X'

FIELDLENGTH = 20
FIELDWIDTH = 35

def GetHowLongToRun():
 print('Welcome to the Plant Growing Simulation')
 print()
 print('You can step through the simulation a year at a time')
 print('or run the simulation for 0 to 5 years')
 print('How many years do you want the simulation to run?')
 Years = int(input('Enter a number between 0 and 5, or -1 for stepping mode: '))
 return Years

def CreateNewField():
 Field = [[SOIL for Column in range(FIELDWIDTH)] for Row in range(FIELDLENGTH)]
 Row = FIELDLENGTH // 2
 Column = FIELDWIDTH // 2
 Field[Row][Column] = SEED
 return Field

def ReadFile():
 FileName = input('Enter file name: ')
 Field = [[SOIL for Column in range(FIELDWIDTH)] for Row in range(FIELDLENGTH)]
 try:
  FileHandle = open(FileName, 'r')
  for Row in range(FIELDLENGTH):
   FieldRow = FileHandle.readline()
   for Column in range(FIELDWIDTH):
    Field[Row][Column] = FieldRow[Column]
  FileHandle.close()
```

```
  except:
    Field = CreateNewField()
  return Field

def InitialiseField():
  Response = input('Do you want to load a file with seed positions? (Y/N): ')
  if Response == 'Y':
    Field = ReadFile()
  else:
    Field = CreateNewField()
  return Field

def Display(Field, Season, Year):
  print('Season: ', Season, '  Year number: ', Year)
  for Row in range(FIELDLENGTH):
    for Column in range(FIELDWIDTH):
      print(Field[Row][Column], end='')
    print('|{0:>3}'.format(Row))
  print()

def CountPlants(Field):
  NumberOfPlants = 0
  for Row in range(FIELDLENGTH):
    for Column in range(FIELDWIDTH):
      if Field[Row][Column] == PLANT:
        NumberOfPlants += 1
  if NumberOfPlants == 1:
    print('There is 1 plant growing')
  else:
    print('There are', NumberOfPlants, 'plants growing')

def SimulateSpring(Field):
  for Row in range(FIELDLENGTH):
    for Column in range(FIELDWIDTH):
      if Field[Row][Column] == SEED:
        Field[Row][Column] = PLANT
  CountPlants(Field)
  if randint(0, 1) == 1:
    Frost = True
  else:
    Frost = False
  if Frost:
    PlantCount = 0
    for Row in range(FIELDLENGTH):
      for Column in range(FIELDWIDTH):
        if Field[Row][Column] == PLANT:
          PlantCount += 1
          if PlantCount % 3 == 0:
```

```
            Field[Row][Column] = SOIL
    print('There has been a frost')
    CountPlants(Field)
  return Field

def SimulateSummer(Field):
  RainFall = randint(0, 2)
  if RainFall == 0:
    PlantCount = 0
    for Row in range(FIELDLENGTH):
      for Column in range(FIELDWIDTH):
        if Field[Row][Column] == PLANT:
          PlantCount += 1
          if PlantCount % 2 == 0:
            Field[Row][Column] = SOIL
    print('There has been a severe drought')
    CountPlants(Field)
  return Field

def SeedLands(Field, Row, Column):
  if Row >= 0 and Row < FIELDLENGTH and Column >= 0 and Column < FIELDWIDTH:
    if Field[Row][Column] == SOIL:
      Field[Row][Column] = SEED
  return Field

def SimulateAutumn(Field):
  for Row in range(FIELDLENGTH):
    for Column in range(FIELDWIDTH):
      if Field[Row][Column] == PLANT:
        Field = SeedLands(Field, Row - 1, Column - 1)
        Field = SeedLands(Field, Row - 1, Column)
        Field = SeedLands(Field, Row - 1, Column + 1)
        Field = SeedLands(Field, Row, Column - 1)
        Field = SeedLands(Field, Row, Column + 1)
        Field = SeedLands(Field, Row + 1, Column - 1)
        Field = SeedLands(Field, Row + 1, Column)
        Field = SeedLands(Field, Row + 1, Column + 1)
  return Field

def SimulateWinter(Field):
  for Row in range(FIELDLENGTH):
    for Column in range(FIELDWIDTH):
      if Field[Row][Column] == PLANT:
        Field[Row][Column] = SOIL
  return Field

def SimulateOneYear(Field, Year):
  Field = SimulateSpring(Field)
```

```
  Display(Field, 'spring', Year)
  Field = SimulateSummer(Field)
  Display(Field, 'summer', Year)
  Field = SimulateAutumn(Field)
  Display(Field, 'autumn', Year)
  Field = SimulateWinter(Field)
  Display(Field, 'winter', Year)

def Simulation():
  YearsToRun = GetHowLongToRun()
  if YearsToRun != 0:
    Field = InitialiseField()
    if YearsToRun >= 1:
      for Year in range(1, YearsToRun + 1):
        SimulateOneYear(Field, Year)
    else:
      Continuing = True
      Year = 0
      while Continuing:
        Year += 1
        SimulateOneYear(Field, Year)
        Response = input('Press Enter to run simulation for another Year, Input X to stop: ')
        if Response == 'x' or Response == 'X':
          Continuing = False
    print('End of Simulation')
  input()

if __name__ == "__main__":
  Simulation()
```

QUESTIONS

Q1.

State the name of an identifier for:

(a) a variable that is used to store a Boolean value.

(1)

_____

_____

(b) a user-defined subroutine that returns an integer value.

(1)

_____

_____

(c) an array or list variable.

(1)

_____

_____

(d) a local variable used to store a string value.

(1)

_____

_____

(Total 4 marks)

Q2.

(a) Explain the benefits of defining FIELDLENGTH and FIELDWIDTH as named constants instead of using the actual values in the code.

(2)

(b) Explain the purpose of the first selection structure in the subroutine SeedLands.

(1)

(c) The simulation is to be refined to include another level of drought: a minor drought.

There is a 1 in 3 chance of a severe drought, and a 1 in 3 chance of a minor drought. If there is a minor drought, then one in four plants will die.

Describe the changes that need to be made to the subroutine SimulateSummer in order to simulate a minor drought.

You are **not** expected to make any changes to the **Skeleton Program**.

(3)

(d) State the type of arithmetic operation carried out in the subroutine CreateNewField when the variables Row and Column are assigned new values outside the FOR loops.

State the values calculated for Row and Column using the values of FIELDLENGTH and FIELDWIDTH that are specified in the **Skeleton Program**.

(2)

_____

_____

_____

_____

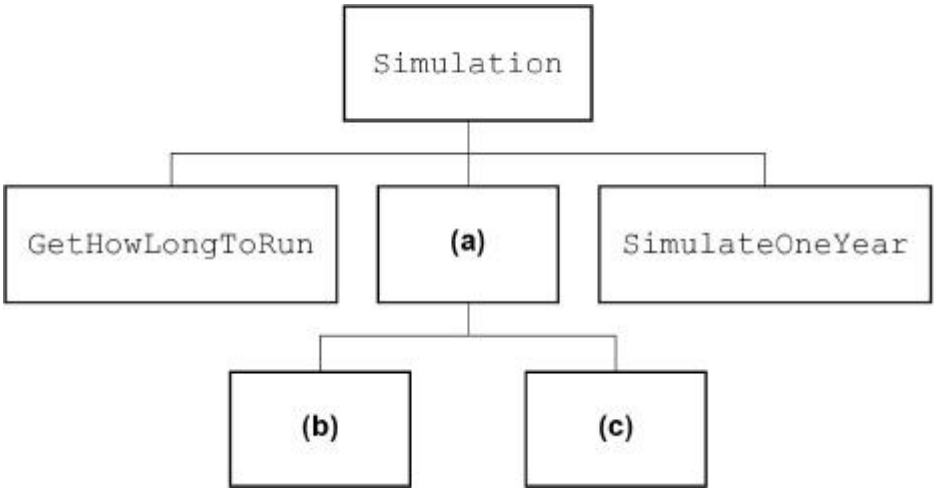
_____

(Total 8 marks)

Q3.

Figure 1 shows an incomplete hierarchy chart for part of the **Skeleton Program**.

Figure 1



With reference to the **Skeleton Program** and **Figure 1**, answer the questions below.

(a) What should be written in box **(a)** in **Figure 1**?

(1)

_____

_____

(b) What should be written in box **(b)** in **Figure 1**?

(1)

_____

_____

(c) What should be written in box **(c)** in **Figure 1**?

(1)

_____

_____

(d) Explain how data is shared between the separate subroutines.

(2)

_____

_____

_____

_____**(Total 5 marks)**

Q4.

This question refers to the subroutine ReadFile.

(a) Explain what will happen if the number of rows and columns supplied in the data file is greater than the number of rows and columns of the field modelled in the **Skeleton Program**.

(2)

(b) Assume it is a perfect growing year and there is no frost in spring and no drought in summer.

A **Data File** is read in which has an S (seed) in every possible location.

Explain what happens to the contents of Field in each season for the **first year only**.

(4)

(Total 6 marks)