

A



**GCSE**

**COMPUTER SCIENCE**

**Paper 1 Computational thinking and programming  
skills – C#**

**8525/1A**

**Diagram Booklet**

**[Turn over]**

**FIGURE 1**

```
country ← 'United States of America'  
state ← 'California'  
city ← 'San Francisco'  
landmark ← 'Alcatraz Island'
```

**FIGURE 2**

```
1   again ← True  
2   WHILE again = True  
3       a ← USERINPUT  
4       IF a > 0 THEN  
5           counter ← 0  
6           WHILE a > 0  
7               a ← a DIV 3  
8               counter ← counter + 1  
9           ENDWHILE  
10      ELSE  
11          again ← False  
12      ENDIF  
13      OUTPUT a  
14  ENDWHILE
```

**BLANK PAGE**

**[Turn over]**

**FIGURE 3**

```
public static int calculate(int width, int length,
int height) {
    if (height == -1)
    {
        return width * length;
    } else
    {
        return width * length * height;
    }
}

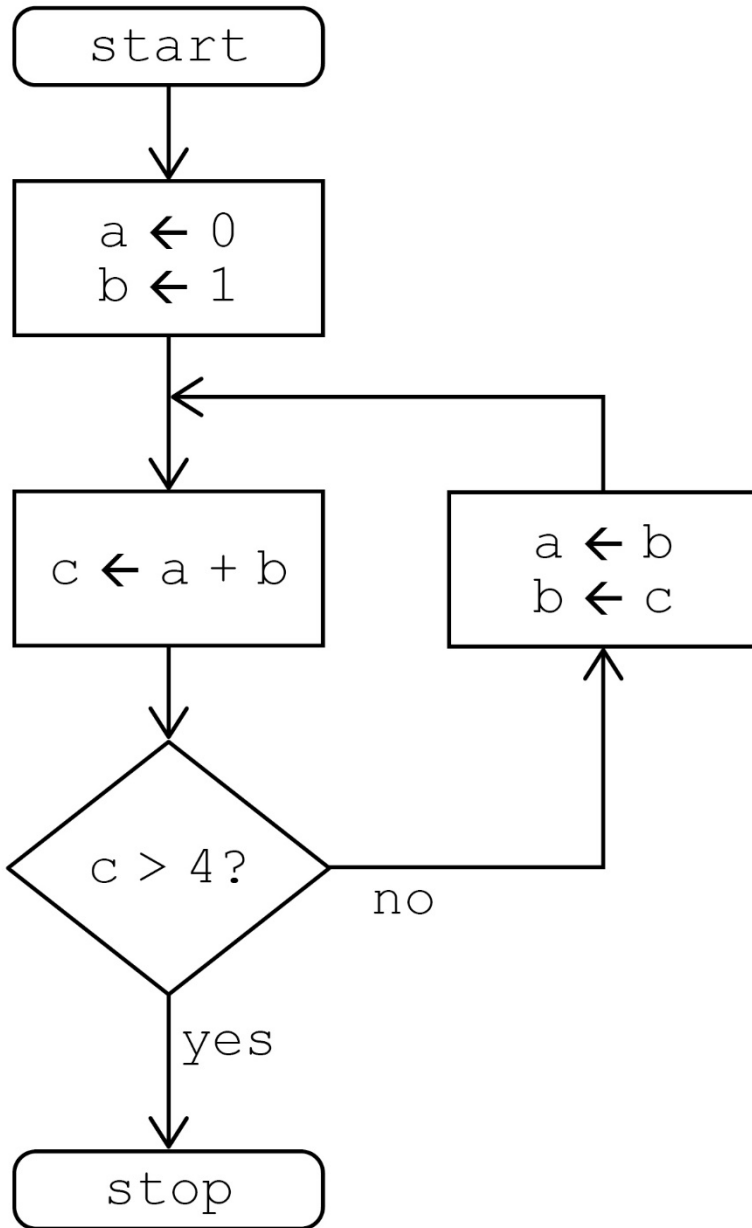
public static void Main() {
    int numOne, numTwo, numThree, answer;
    Console.Write("Enter width: ");
    numOne = Convert.ToInt32(Console.ReadLine());
    Console.Write("Enter length: ");
    numTwo = Convert.ToInt32(Console.ReadLine());
    Console.Write("Enter height, -1 to ignore: ");
    numThree = Convert.ToInt32(Console.ReadLine());
```

```
answer = calculate(numOne, numTwo, numThree);

if (numThree == -1)
{
    Console.WriteLine($"Area {answer}");
} else
{
    Console.WriteLine($"Volume {answer}");
}
}
```

**[Turn over]**

**BLANK PAGE**

**FIGURE 4****[Turn over]**

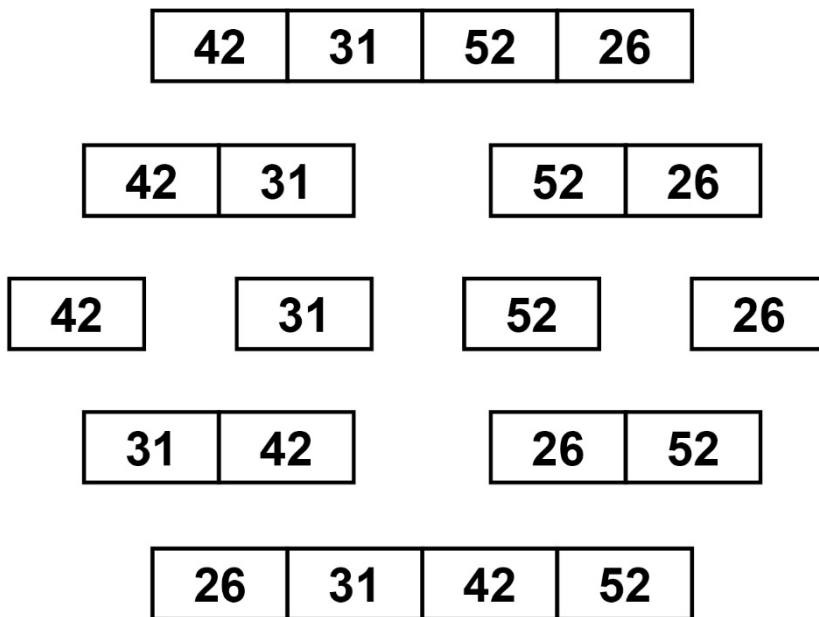
**FIGURE 5**

```
login ← False
REPEAT
    username ← ''
    WHILE username = ''
        OUTPUT 'Enter username: '
        username ← L1
    ENDWHILE
    password ← ''
    WHILE password = ''
        OUTPUT 'Enter password: '
        password ← USERINPUT
    ENDWHILE
    storedPassword ← getPassword(L2)
    IF storedPassword = L3 THEN
        OUTPUT 'L4'
    ELSE
        IF password = storedPassword THEN
            login ← True
        ELSE
            OUTPUT 'Try again.'
        ENDIF
    ENDIF
UNTIL login = True
OUTPUT 'You are now logged in.'
```



**FIGURE 6**

-1	OUTPUT	0
username	True	SUBROUTINE
1	User not found	' '
USERINPUT	password	Wrong password

**FIGURE 7**

**[Turn over]**

## FIGURE 8

```
RECORD Film
  title : String
  certificate : String
  year : Integer
  beingShown : Boolean
ENDRECORD

hulk ← Film('Hulk', '12A', 2005, False)
ironMan ← Film('Iron Man', '12A', 2008, False)
antMan ← Film('Ant-Man', '12A', 2015, False)
filmCollection ← [antMan, hulk, ironMan]
year ← 0
position ← 0
FOR i ← 0 TO L1
```

```
IF filmCollection[i].year > year THEN
    year ← filmCollection[i].year
    position ← i
ENDIF
ENDFOR

OUTPUT filmCollection[position].title, ' is the
newest film'
```

**[Turn over]**

**FIGURE 9**

```
1  names ← ['Natalie', 'Alex', 'Roshana']
2  scores ← [78, 81, 72, 27, 51, 54, 52, 55, 59]
3  count ← 0
4  FOR i ← 0 TO 2
5      person ← names[i]
6      OUTPUT 'Student: ', person
7      FOR j ← 0 TO 1
8          OUTPUT j + 1
9          result ← scores[i * 3 + j]
10         OUTPUT result
11         count ← count + 1
12     ENDFOR
13 ENDFOR
```

**BLANK PAGE**

**[Turn over]**

**FIGURE 10**

```
1  validChoice ← False
2  REPEAT
3    difference ← -1
4    OUTPUT 'Enter a start year '
5    startYear ← USERINPUT
6    OUTPUT 'Enter an end year '
7    endYear ← USERINPUT
8    IF startYear ≥ endYear THEN
9      OUTPUT 'Start year must be before end year'
10   ELSE
11     IF startYear < 2000 THEN
12       OUTPUT 'Start year must be before 2000'
```

```
13     ELSE
14         validChoice ← True
15     ENDIF
16 ENDIF
17 UNTIL validChoice = True
18     difference ← endYear - startYear
19 OUTPUT difference
```

**[Turn over]**

**FIGURE 11**

```
string[] animals = {"cat", "dog", "hippo", "llama",  
"ox", "rat", "tiger", "wolf"};  
Console.WriteLine("What animal would you like to find? ");  
string animalToFind = Console.ReadLine();  
bool validAnimal = false;  
int start = 0;  
int finish = animals.Length - 1;  
while (validAnimal == false && start <= finish) {  
    int mid = (start + finish) / 2;  
    if (animals[mid] == animalToFind) {  
        validAnimal = true; }  
    else if (animalToFind.CompareTo(animals[mid]) == 1)  
    {  
        start = mid + 1;  
    } else {
```



```
        finish = mid - 1;
    }
}
Console.WriteLine(validAnimal);
```

**[Turn over]**

**FIGURE 13**

```
1  SUBROUTINE diffCurrencies(currencies)
2      currencies ← ['baht', 'dollar', 'euro',
3                  'koruna', 'lira', 'rand',
4                  'rupee', 'yen']
5
6      RETURN currencies[x]
7  ENDSUBROUTINE
8
9  FOR i ← 8 TO 0 STEP 1
10     OUTPUT(diffCurrencies(i))
11 ENDFOR
```

**FIGURE 14**

	<b>A</b>	<b>B</b>	<b>C</b>
<b>1</b>			
<b>2</b>			
<b>3</b>			X

**[Turn over]**

**FIGURE 15**

```
bool check = false;
while (check == false) {
    string square = "";
    while (square.Length != 2) {
        Console.Write("Enter grid reference (eg C2): ");
        square = Console.ReadLine();
        square = square.ToUpper();
    }
}
```

**BLANK PAGE**

**[Turn over]**

**FIGURE 16**

```
SUBROUTINE showResults(method, numberOfGenres)
  results ← [['Pop', 'Post-Punk', 'Techno', 'Metal',
             'Dance'], ['7', '19', '14', '1', '9']]
  pos ← 0
  high ← -1
  IF method = 'HIGHEST' THEN
    FOR i ← 0 TO numberOfGenres - 1
      Votes ← STRING_TO_INT(results[L1][i])
      IF votes > high THEN
        high ← votes
        pos ← L2
      ENDIF
    ENDFOR
  ELSE
```

```
    OUTPUT 'not yet working'
ENDIF
IF high ≠ -1 THEN
    OUTPUT results[0][pos], ' with ', results[1][pos]
ENDIF
ENDSUBROUTINE
```

```
OUTPUT 'Show the genre with the HIGHEST or LOWEST number
of votes? '
method ← USERINPUT
showResults( L3, 5)
```

**[Turn over]**

**FIGURE 17**

Roll 1: 1

Roll 2: 4

Current score: 5

Would you like to roll again? yes

Roll 1: 1

Roll 2: 6

Current score: 12

Would you like to roll again? yes

Roll 1: 1

Roll 2: 2

Current score: 15

Would you like to roll again? yes

Roll 1: 6

Roll 2: 1

Current score: 22

You lost!

**END OF DIAGRAM BOOKLET**



**BLANK PAGE**

**BLANK PAGE****Copyright information**

For confidentiality purposes, all acknowledgements of third-party copyright material are published in a separate booklet. This booklet is published after each live examination series and is available for free download from [www.aqa.org.uk](http://www.aqa.org.uk).

Permission to reproduce all copyright material has been applied for. In some cases, efforts to contact copyright-holders may have been unsuccessful and AQA will be happy to rectify any omissions of acknowledgements. If you have any queries please contact the Copyright Team.

Copyright © 2023 AQA and its licensors. All rights reserved.

**WP/M/CD/Jun23/8525/1A/E3**