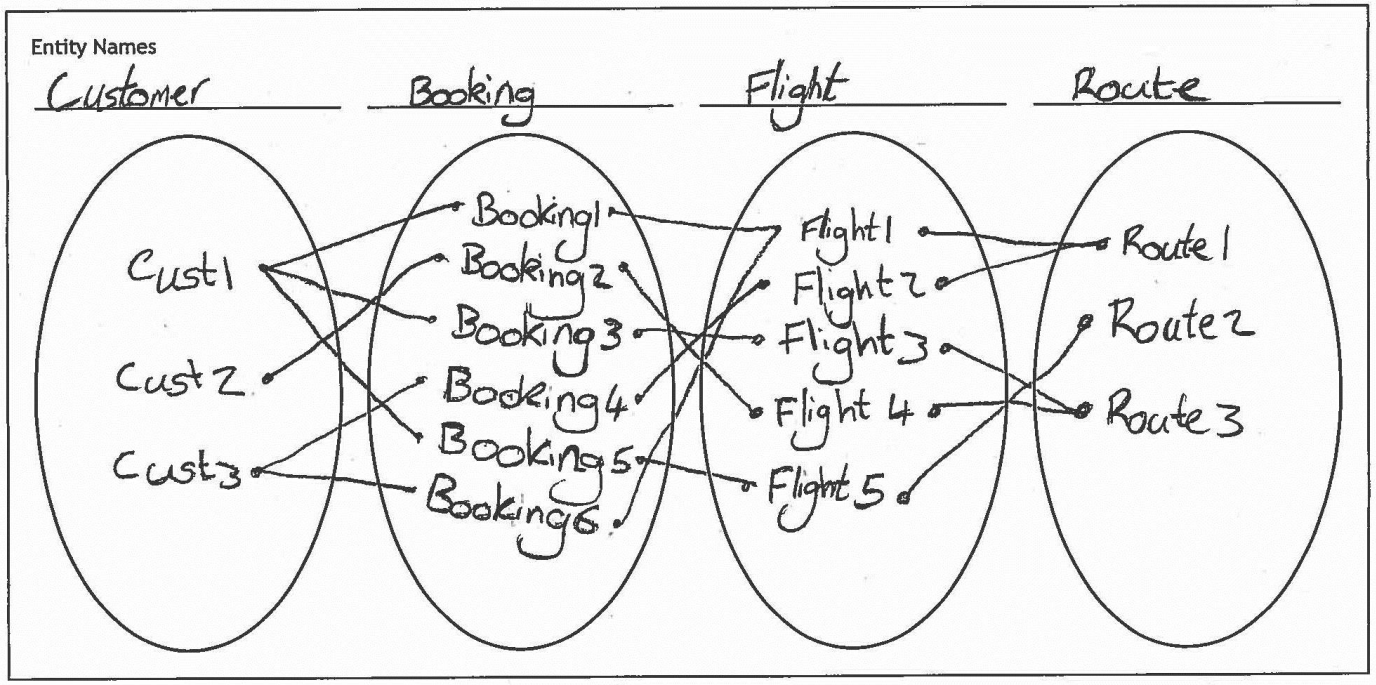
Candidate 2

Task 1 – Database design and development

Question 1(a)

Task 1a

Entity-occurrence diagram



- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Question 1(b)(i)

Task 1: database design and development (part B)

1B(i):

SQL:

John Smith Tax

```
SELECT forename, surname, (adultTicket * 5.50) + (childTicket * 2) + (concessionTicket * 1.50) AS [Tax (£)]
FROM Customer, Booking
WHERE Customer.customerID = Booking.customerID
AND Customer.customerID = 'GR01932'
AND flightID = 'QH182';
```

Output:

John Smith Tax

forename	surname	Tax (£)
John	Smith	52.5

Question 1(b)(ii)

1B(ii):

SQL(1):

John Smith Tax | MostChildren | Customers with Greatest Number of Children

```
SELECT MAX(childTicket) AS [MostChildren]
FROM Booking;
```

Output(1):

John Smith Tax | MostChildren | Customers with Greatest Number of Children | Booking | Customer

MostChildre
8

SQL(2):

John Smith Tax | MostChildren | Customers with Greatest Number of Children | Booking | Customer

```
SELECT forename, surname
FROM Customer, Booking, MostChildren
WHERE Customer.customerID = Booking.CustomerID
AND childTicket = mostChildren;
```

Output(2):

John Smith Tax | MostChildren | Customers with Greatest Number of Children | Booking | Customer

forename	surname
Tahir	Baqri
Kim	Pettigrew

Question 1(c)

1c The database has primary key fields but has no other validation. Evaluate two potential problems that may occur when adding new data to the Flight table.

Problem 1 (1 mark)

- Without a range check on the capacity field, the user could accidentally enter a value that doesn't make sense.

- This would negatively affect the accuracy of output if a customer is looking for a flight but the capacity in the systems is wrong...

Problem 2 (1 mark)

- Without a lookup check on routeID the user could ~~accept~~ accidentally input a value that doesn't exist in the route table.

- ~~This would affect the~~ This could mean that the database is not fit for purpose as flights may not display properly when a customer searches for them as the tables are not correctly linked.

Candidate name te number

Task 2 – Software design and development
Question 2(a)

2a Using the problem description, identify the functional requirements of the program. (3 marks)

Input(s)

- Text file containing the forename, surname and the total distance walked in miles of each of the walkers

Process(es)

- Find the furthest distance walked by any of the walkers
- Find the forename and surname of everyone who walked more than 70% of the furthest distance

Output(s)

- Display the furthest distance walked
- Write the forenames + surnames of the walkers who have walked more than 70% of the furthest distance

Candidate nd

Question 2(b)

2b The top-level design for the program is shown below.
Complete the design to show the missing data flow in and out of each module.

(2 marks)

Top-level design main program modules		
Read members' data from file into array of records	IN	
	OUT	members(forename,surname,distance)
Find the furthest distance walked	IN	members(forename,surname, distance)
	OUT	furthest
Display the furthest distance walked	IN	furthest
	OUT	
Write club prize winners to file	IN	members(forename,surname,distance) furthest
	OUT	

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_ mber_

Question 2(c)(i)

H:\Higher Computing Science\Programming\Walkers (Assignment)\Walkers (Assignment)\Form1.vb 1

```
1  Public Class Form1
2      'Candidate Name:                Candidate Number:
3      'Task 2: software design and development (part B)   2c(i)
4      Structure People
5          Dim forename, surname As String
6          Dim distance As Single
7      End Structure
8
9      Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
       btnStart.Click
10         Dim members(19) As People
11         Dim furthest As Single
12
13         readData(members) 'Read the data from member.txt into the program
14
15         furthest = findFurthest(members) 'Find the furthest distance walked by a member and store it into
       the furthest variable
16
17         displayFurthest(furthest) 'Display the furthest distance walked in the list box
18
19         writeToFile(members, furthest) 'Write the winners forename and surname to the file results.txt
20     End Sub
21
22     Sub readData(ByRef members() As People)
23         Dim items() As String
24         Dim filename As String
25         Dim textline As String
26         Dim counter As Integer
27
28         filename = "members.txt" 'Store the name of the file members.txt into the variable filename
29
30         FileOpen(1, filename, OpenMode.Input) 'Open the members.txt file in input mode
31         counter = 0 'Set the counter to 0
32
33         Do 'Go through the file and save the forename, surname and distance walked to the members array
34             textline = LineInput(1)
35             items = Split(textline, ",")
36
37             members(counter).forename = items(0)
38             members(counter).surname = items(1)
39             members(counter).distance = items(2)
40
41             counter = counter + 1 'Increase the counter by 1
42
43         Loop Until EOF(1) 'Keep looping until the end of the file
44
45         FileClose(1) 'Close the file
46     End Sub
47
48     Function findFurthest(ByVal members() As People)
49
50         Dim max As Single
51
52         max = members(0).distance 'Set the max distance to the first distance in the members array
53
54         For i = 1 To 19 'Start a fixed loop to go through the array
55             If members(i).distance > max Then 'Check if the distance is greatest than the furthest
56                 max = members(i).distance 'Save the new max distance
57             End If
58         Next
59         Return max 'Return the value of max
60
61     End Function
62
63     Sub displayFurthest(ByVal furthest As Single)
64         lstDisplay.Items.Add("The furthest distance walked was " & furthest & " miles.") 'Display the
       furthest distance walked
65     End Sub
66
67     Sub writeToFile(ByVal members() As People, ByVal furthest As Single)
68
69         System.IO.File.Delete("results.txt") 'Clear the results.txt file
70
71         Dim filename As String
```

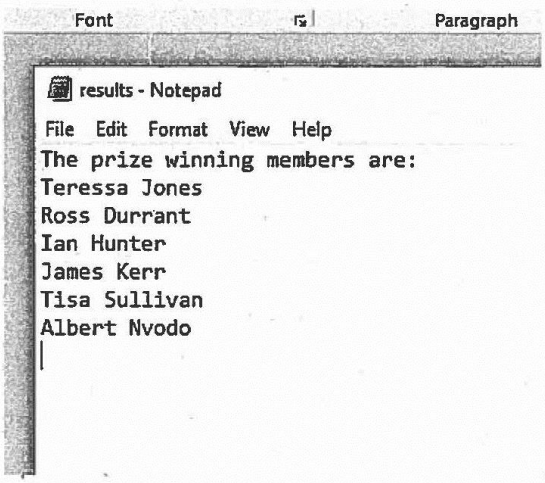
2c(i)

H:\Higher Computing Science\Programming\Walkers (Assignment)\Walkers (Assignment)\Form1.vb 2

```
72
73          filename = "results.txt" 'Save the filename of results.txt to the variable filename
74
75          FileOpen(1, filename, OpenMode.Output) 'Open the results.txt file in output mode
76
77          PrintLine(1, "The prize winning members are:") 'Write "The prize winning members are:" to the file
78
79          For i = 0 To 19
80              If members(i).distance > (0.7 * furthest) Then 'Check if the member has walked more than 70%
        of the furthest distance
81                  PrintLine(1, members(i).forename & " " & members(i).surname) 'Write the forename and
        surname to the results.txt file
82              End If
83          Next
84
85      End Sub
86  End Class
87
```

Task 2: Software Design and Development

2c(i): results.txt file:



Question 2(c)(ii)

H:\Higher Computing Science\Programming\Walkers (Assignment)\Walkers (Assignment)\Form1.vb 1

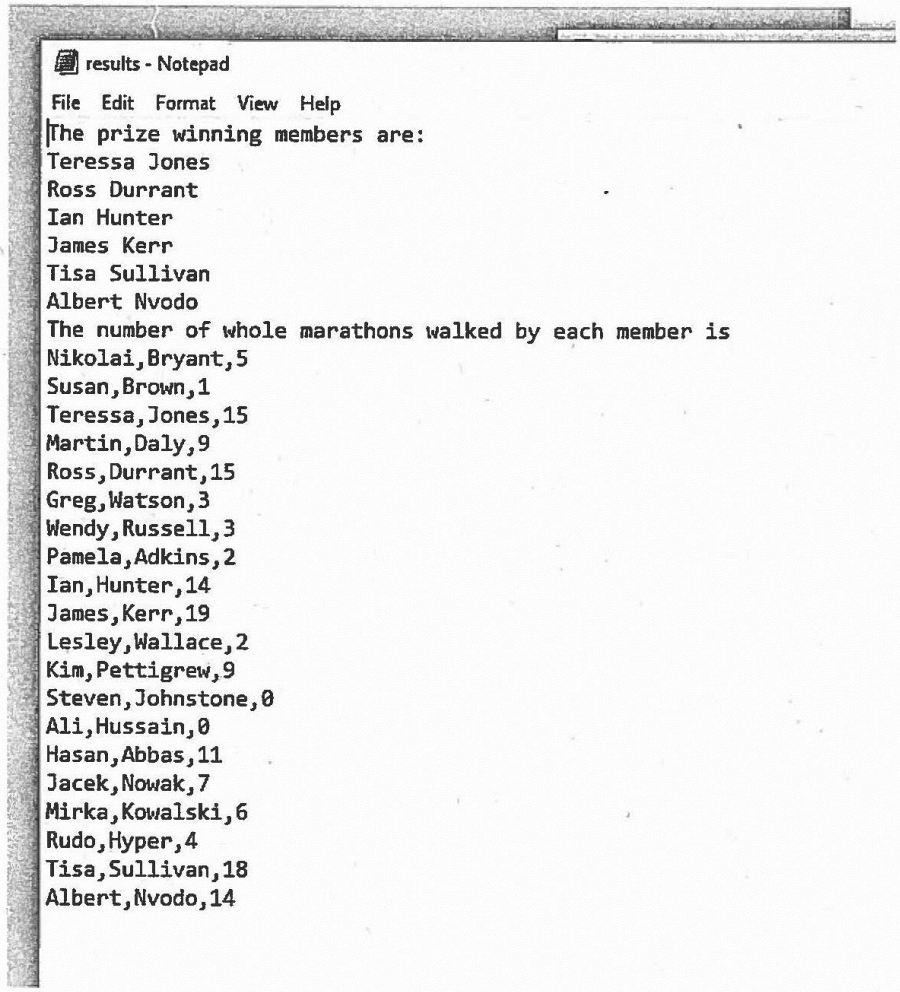
```
1  Public Class Form1
2      'Candidate Name:              Candidate Number:
3      'Task 2: software design and development (part B)  Task 2cc.ii)
4      Structure People
5          Dim forename, surname As String
6          Dim distance As Single
7      End Structure
8
9      Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
       btnStart.Click
10         Dim members(19) As People
11         Dim furthest As Single
12
13         readData(members) 'Read the data from member.txt into the program
14
15         furthest = findFurthest(members) 'Find the furthest distance walked by a member and store it into
       the furthest variable
16
17         displayFurthest(furthest) 'Display the furthest distance walked in the list box
18
19         writeToFile(members, furthest) 'Write the winners forename and surname to the file results.txt
20     End Sub
21
22     Sub readData(ByRef members() As People)
23         Dim items() As String
24         Dim filename As String
25         Dim textline As String
26         Dim counter As Integer
27
28         filename = "members.txt" 'Store the name of the file members.txt into the variable filename
29
30         FileOpen(1, filename, OpenMode.Input) 'Open the members.txt file in input mode
31         counter = 0 'Set the counter to 0
32
33         Do 'Go through the file and save the forename, surname and distance walked to the members array
34             textline = LineInput(1)
35             items = Split(textline, ",")
36
37             members(counter).forename = items(0)
38             members(counter).surname = items(1)
39             members(counter).distance = items(2)
40
41             counter = counter + 1 'Increase the counter by 1
42
43         Loop Until EOF(1) 'Keep looping until the end of the file
44
45         FileClose(1) 'Close the file
46     End Sub
47
48     Function findFurthest(ByVal members() As People)
49
50         Dim max As Single
51
52         max = members(0).distance 'Set the max distance to the first distance in the members array
53
54         For i = 1 To 19 'Start a fixed loop to go through the array
55             If members(i).distance > max Then 'Check if the distance is greatest than the furthest
56                 max = members(i).distance 'Save the new max distance
57             End If
58         Next
59         Return max 'Return the value of max
60
61     End Function
62
63     Sub displayFurthest(ByVal furthest As Single)
64         lstDisplay.Items.Add("The furthest distance walked was " & furthest & " miles.") 'Display the
       furthest distance walked
65     End Sub
66
67     Sub writeToFile(ByVal members() As People, ByVal furthest As Single)
68
69         System.IO.File.Delete("results.txt") 'Clear the results.txt file
70         Dim filename As String
71
```

H:\Higher Computing Science\Programming\Walkers (Assignment)\Walkers (Assignment)\Form1.vb 2

```
72          Dim marathons As Integer
73
74          filename = "results.txt" 'Save the filename of results.txt to the variable filename
75
76          FileOpen(1, filename, OpenMode.Output) 'Open the results.txt file in output mode
77
78          PrintLine(1, "The prize winning members are:") 'Write "The prize winning members are:" to the file
79
80          For i = 0 To 19
81              If members(i).distance > (0.7 * furthest) Then 'Check if the member has walked more than 70%
    of the furthest distance
82                  PrintLine(1, members(i).forename & " " & members(i).surname) 'Write the forename and
    surname to the results.txt file
83              End If
84          Next
85
86          PrintLine(1, "The number of whole marathons walked by each member is:")
87
88          For i = 0 To 19 'Loop through the array of members
89              marathons = Int(members(i).distance / 26.22) 'Calculate the number of marathons walked, and
    round the answer down to the nearest whole integer.
90              PrintLine(1, members(i).forename & "," & members(i).surname & "," & marathons) 'Write the
    results to the file "results.txt".
91          Next
92
93          FileClose(1) 'Close the file "results.txt"
94
95      End Sub
96  End Class
97
```

Task 2: Software Design and Development

2c(ii): results.txt file:



results - Notepad

File Edit Format View Help

```
The prize winning members are:
Teressa Jones
Ross Durrant
Ian Hunter
James Kerr
Tisa Sullivan
Albert Nvodo
The number of whole marathons walked by each member is
Nikolai,Bryant,5
Susan,Brown,1
Teressa,Jones,15
Martin,Daly,9
Ross,Durrant,15
Greg,Watson,3
Wendy,Russell,3
Pamela,Adkins,2
Ian,Hunter,14
James,Kerr,19
Lesley,Wallace,2
Kim,Pettigrew,9
Steven,Johnstone,0
Ali,Hussain,0
Hasan,Abbas,11
Jacek,Nowak,7
Mirka,Kowalski,6
Rudo,Hyper,4
Tisa,Sullivan,18
Albert,Nvodo,14
```

Question 2(d)

2d The function in step 2 is to be tested with the data shown below.

John,Davie,189.4
Susie,Small,14.6
Johnny,Atom,490.2
Wendy,Khan,512.5
Emir,Jones,170.3

Using the variable names and data structure names from your own code, create a trace table to find the furthest distance walked by the members in the test data.

(2 marks)

Counter	Maxposition	Member(counter).distance	Max
	0	189.4	189.4
1	0	14.6	189.4
2	2	490.2	490.2
3	3	512.5	512.5
4	3	170.3	512.5

Candidate na

Question 2(e)

2e With reference to your own program code, evaluate:

- the fitness for purpose of your program

(1 mark)

I believe this program carries out all tasks set in the problem description and accurately follows the design structures given.

- the maintainability of your program with reference to readability and modularity

(2 marks)

In terms of readability. I believe my program is quite readable. I have made use of indentation so that procedures, loops and conditions can be identified. I have made use of white space to leave gaps between elements so they can be more easily identified. I have used internal commentary to describe what each line of the program is doing and have used meaningful variable names to show what each element is representing so that it makes sense to other programmers.

In terms of modularity. I have used procedures (Private sub) and functions in my program, this allows changes to be made without affecting the rest of the program's code and it is easier to identify separate components. I have made use of global variables through these and local variables within these.

Overall, this program is very maintainable.

Candidate name

Task 3 – Web design and development
Question 3(a)

3a State two functional requirements for the website.

Functional requirement 1

(1 mark)

The website must be able to display images across web pages

Functional requirement 2

(1 mark)

The foundation contact page should include an online survey where users can complete a form to receive a set of free cards

Candidate nam

Question 3(b)(i)

3b(i)

```
<!DOCTYPE html>
<html>

<head>
<title>History</title>
<link rel="stylesheet" type="text/css" href="../CSS/styles.css">
</head>

<body>

<!-- Page Header -->
<header>
<img class="imageBannerLeft" src="../images/bannerCardsLeft.png">
<img class="imageBannerRight" src="../images/bannerCardsRight.png">
<h1>Playing Cards</h1>

</header>

<!-- Navigation Bar -->
<nav>
<ul>
<li><a href="home.html">Home</a></li>
<li><a href="history.html">History</a></li>
<li><a href="multi.html">Multi-player</a></li>
<li><a href="single.html">Single-player</a></li>
<li><a href="register.html">Free Cards</a></li>
</ul>
</nav>

<!-- The main content of the page -->
<main>
<section id="earlyhistory">
<h2>Early history</h2>
<p>The first playing cards are recorded as being invented in China around the
9th century AD by the Tang dynasty author Su E who writes about the card game
"leaf" in the text Collection of Miscellanea at Duyang.  The text describes
Princess Tongchang, daughter of Emperor Yizong of Tang, playing leaf in 868AD
with members of the family of the princess' husband.</p>
<p>The mass production of Cards became possible following the invention of
wooden printing block technology.  Early Chinese packs contained 30 cards with
no suits.</p>
<p>The first cards may have doubled as actual paper currency being both the
tools of gaming and the stakes being played for.  This is similar to modern
trading card games. Using paper money was inconvenient and risky so they were
substituted by play money known as "money cards".</p>
<p>The earliest dated instance of a game involving cards with suits and numerals
occurred on 17 July 1294.</p>
</section>

<section id="european">
<h2>European Adoption</h2>
<p>The first four-suited playing cards appeared in Europe in 1365.  They are
thought to originate from traditional latin decks whose suits included: cups,
coins, swords, and polo-sticks.  As Polo was not yet a European game, polo
sticks became batons (or cudgels).  Wide use of playing cards is recorded from
1377 onwards.</p>
<p>Professional card makers in Ulm, Nuremberg, and Augsburg created printed
decks. Playing cards even competed with devotional images as the most common
uses for woodcuts in this period. These 15th-century playing cards were probably
painted.</p>
<p>The Flemish Hunting Deck, held by the Metropolitan Museum of Art is the
oldest complete set of ordinary playing cards made in Europe.</p>
<p>Cards were adapted in Europe to contain members of the royal court and by the
15th Century French and English packs of 56 cards contain the King, Queen and
Knave cards.</p>
</section>

<section id="modern">
<img class="imageCardsRight" src="../images/germanPlayingCards.jpg" id="german">
<h2>Modern Cards</h2>
```

```
51    <p>Contemporary playing cards are grouped into three broad categories based on
      the suits they use: French, Latin, and Germanic. Latin suits are used in the
      closely related Spanish and Italian formats. The Swiss-German suits are distinct
      enough to merit their subcategory. Excluding Jokers and Tarot trumps, the French
      52-card deck preserves the number of cards in the original Mamluk deck, while
      Latin and Germanic decks average fewer.</p>
52    <p>Within suits, there are regional or national variations called "standard
      patterns" because they are in the public domain, allowing multiple card
      manufacturers to copy them. Pattern differences are most easily found in the
      face cards but the number of cards per deck, the use of numeric indices, or even
      minor shape and arrangement differences of the pips can be used to distinguish
      them. Some patterns have been around for hundreds of years. Jokers are not part
      of any pattern as they are a relatively recent invention and lack any
      standardized appearance so each publisher usually puts their own trademarked
      illustration into their decks. </p>
53    </section>
54
55
56
57    </main>
58
59    <!-- Page Footer -->
60    <footer>
61    <p> &copy; Card Foundation <br> 2017 <br> Please provide feedback to:
      jlongridge@jlinternet.co.uk </p>
62    </footer>
63
64    </body>
65    </html>
```

```
/* The Universal Selector has been used here to cancel the browser default
margins and */
/* padding for every page element.  This ensures that only assigned margin and
padding values are seen when viewing pages. */
* {margin:0;padding:0}

/* Margins - main page areas */
header, main, footer  {margin-top:5px}

/* Padding */
header, footer, section {padding:10px}

/* Positioning (Float, Display and Clear) */
header, nav, main, footer {display:block;clear:both}
.hidden{display:none}

/* Background Colours */
body, div, main {background-color:LightBlue}
nav {background-color:LightSteelBlue}
header, footer {background-color:DarkBlue}
section {background-color:White}



/* CSS specific to page areas */

/* Body */
body{margin:auto;width:1000px}

/* History */

#earlyhistory {float: left}
#european {float: right}
#modern {clear: both}

#earlyhistory {margin-right: 10px}
#european, #earlyhistory {margin-bottom: 10px}

#earlyhistory, #european {width: 475px}

#german {display: inline; float: right;}

/* Header */
header {height:60px}
h1{font-family:Verdanda;color:White;font-size:30pt;text-align:center;display:block
}
.imageBannerRight {width:60px;height:60px;float:right}
.imageBannerLeft {width:60px;height:60px;float:left}



/* Nav */
nav  {margin-top:0px}
nav {height:34px}

nav ul {list-style-type:none}
nav ul li {float:left;width:200px;text-align:center}
nav ul li a {display:block;padding:8px}
nav ul li a:hover {background-color:DarkBlue;color:White}



/* Main */
main {display:block}
main section {margin-bottom:5px}
p ul {margin-top:15px;margin-bottom:15px;font-size:12pt;text-align:left}
h2, p{font-family:Helvetica;color:Black}
h2 {margin-bottom:10px;font-size:14pt;text-align:left}
p{margin-bottom:10px;font-size:10pt;text-align:left}
input, select {margin-left:50px}
```

```
69
70  /*Home Page*/
71  .imageLargeLeft
    {width:300px;height:225px;float:left;margin-right:10px;margin-bottom:10px}
72
73  /* History Page  */
74  .imageCardsRight {width:120px;height:90px}
75
76  /* Patience Games Page  */
77  .imageMediumRight
    {width:120px;height:90px;float:right;margin-left:10px;margin-bottom:10px}
78  .patienceButtons {width:250px;height:170px;margin-left:30px;margin-right:30px}
79  .patiencePhotos
    {width:180px;height:135px;float:right;margin-left:10px;margin-bottom:20px}
80
81  /*Single Page*/
82  .imageLargeRight
    {width:300px;height:225px;float:right;margin-left:10px;margin-bottom:10px}
83
84  /*Multi Page*/
85  .multiPhotos {width:330px;height:200px;margin-bottom:5px}
86  .firstTwoColumns {float:left;width:330px;margin-right:5px}
87  .thirdColumn {float:left;width:329px}
88
89
90
91
92  /* Footer */
93  footer {height:50px}
94  footer p
    {font-family:Verdanda;text-align:left;color:White;font-size:10pt;margin-top:0px}
95
96
97
```

04/03/2019 History

Playing Cards

Home | History | Multi-player | Single-player | Free Cards

Early history

The first playing cards are recorded as being invented in China around the 9th century AD by the Tang dynasty author Su E who writes about the card game "leaf" in the text Collection of Miscellanea at Duyang. The text describes Princess Tongchang, daughter of Emperor Yizong of Tang, playing leaf in 868AD with members of the family of the princess' husband.

The mass production of Cards became possible following the invention of wooden printing block technology. Early Chinese packs contained 30 cards with no suits.

The first cards may have doubled as actual paper currency being both the tools of gaming and the stakes being played for. This is similar to modern trading card games. Using paper money was inconvenient and risky so they were substituted by play money known as "money cards".

The earliest dated instance of a game involving cards with suits and numerals occurred on 17 July 1294.

European Adoption

The first four-suited playing cards appeared in Europe in 1365. They are thought to originate from traditional latin decks whose suits included: cups, coins, swords, and polo-sticks. As Polo was not yet a European game, polo sticks became batons (or cudgels). Wide use of playing cards is recorded from 1377 onwards.

Professional card makers in Ulm, Nuremberg, and Augsburg created printed decks. Playing cards even competed with devotional images as the most common uses for woodcuts in this period. These 15th-century playing cards were probably painted.

The Flemish Hunting Deck, held by the Metropolitan Museum of Art is the oldest complete set of ordinary playing cards made in Europe.

Cards were adapted in Europe to contain members of the royal court and by the 15th Century French and English packs of 56 cards contain the King, Queen and Knave cards.

Modern Cards

Contemporary playing cards are grouped into three broad categories based on the suits they use: French, Latin, and Germanic. Latin suits are used in the closely related Spanish and Italian formats. The Swiss-German suits are distinct enough to merit their subcategory. Excluding Jokers and Tarot trumps, the French 52-card deck preserves the number of cards in the original Mamluk deck, while Latin and Germanic decks average fewer.

Within suits, there are regional or national variations called "standard patterns" because they are in the public domain, allowing multiple card manufacturers to copy them. Pattern differences are most easily found in the face cards but the number of cards per deck, the use of numeric indices, or even minor shape and arrangement differences of the pips can be used to distinguish them. Some patterns have been around for hundreds of years. Jokers are not part of any pattern as they are a relatively recent invention and lack any standardized appearance so each publisher usually puts their own trademarked illustration into their decks.

3b(i)

Question 3(b)(ii)

```
<!DOCTYPE html>
<html>

<head>
<title>Patience</title>
<link rel="stylesheet" type="text/css" href="../CSS/styles.css">

</head>

<script>

function displayKlondike() {
document.getElementById("Klondike").style.display="block";
document.getElementById("Freecell").style.display="none";
document.getElementById("Pyramid").style.display="none";

}

function displayFreecell() {
document.getElementById("Klondike").style.display="none";
document.getElementById("Freecell").style.display="block";
document.getElementById("Pyramid").style.display="none";

}

function displayPyramid() {
document.getElementById("Klondike").style.display="none";
document.getElementById("Freecell").style.display="none";
document.getElementById("Pyramid").style.display="block";

}

</script>

<body>

<!-- Page Header -->
<header>
<img class="imageBannerLeft" src="../images/bannerCardsLeft.png">
<img class="imageBannerRight" src="../images/bannerCardsRight.png">
<h1>Playing Cards</h1>

</header>

<!-- Navigation Bar -->
<nav>
<ul>
<li><a href="home.html">Home</a></li>
<li><a href="history.html">History</a></li>
<li><a href="multi.html">Multi-player</a></li>
<li><a href="single.html">Single-player</a></li>
<li><a href="register.html">Free Cards</a></li>
</ul>
</nav>

<!-- The main content of the page -->
<main>
<section>
<h2>Popular Patience Games</h2>
<p>Click on the names below to read more about each game of patience.</p>
<p>
<img class="patienceButtons" src="../images/klondikeButton.png"
onclick="displayKlondike()">
<img class="patienceButtons" src="../images/freecellButton.png"
onclick="displayFreecell()">
<img class="patienceButtons" src="../images/pyramidButton.png"
onclick="displayPyramid()">
</p>
</section>

```

```
68   <section id="Klondike" style="display:block;">
69   <img class="patiencePhotos" src="../images/klondikePhoto.jpeg">
70   <h2>Klondike</h2>
71   <p>Klondike is the best known of the patience games.  It was invented in the
     late 19th century and was named after the place Klondike, the gold rush area in
     Canada. Some think it may have been invented by the gold prospectors of the
     time. </p>
72   <p>Klondike is played by laying 7 piles of cards face down with each pile having
     one more card than the pile to its left.  The top card of each pile is turned
     face up.  The remaining cards are placed face down in a separate playing pile.</p>
73   <p>The player must build new 4 piles (one for each suit) starting with the ace,
     then 2, then 3 and so on.  The piles are created as the next cards required to
     build each pile are discovered.</p>
74   </section>
75
76   <section id="Freecell" style="display:none;">
77   <img class="patiencePhotos" src="../images/freecellPhoto.jpeg">
78   <h2>Freecell</h2>
79   <p>FreeCell is played using the standard 52-card deck. Compared to most patience
     games it is very easy to complete the game. All cards are dealt face-up at the
     beginning of the game.</p>
80   <p>FreeCell has been included as part of Windows since 1995 making it very
     popular.  Many other platforms have attempted to replicate the MicroSoft version
     as players don't tend to like variations from that version.</p>
81   <p>The game begins with four open cells and four foundations.  A standard 52
     card deck of cards is used.</p>
82   <p>Cards are dealt face-up into eight columns positioned so that every card can
     be seen.  Four columns have seven cards and four have six cards.</p>
83   <p>The top card of any colum can be covered with a card one lower in value and
     the opposite colour. Foundations are built up by suit.</p>
84   <p>The four cells can be used to temporarly store cards while moving other cards
     around.</p>
85   <p>The game is complete when all four foundation piles are complete from Ace to
     King in their correct suits. </p>
86   </section>
87
88
89   <section id="Pyramid" style="display:none;">
90   <img class="patiencePhotos" src="../images/pyramidPhoto.jpeg">
91   <h2>Pyramid</h2>
92   <p>The object is to remove all the cards from the pyramid to the foundation. </p>
93   <p>The game begins by laying out a pyramid of cards.  One card at the top
     covered by two cards on the next row down, covered by three cards on the next
     row and so on up to the last row of seven cards (28 cards in total).  The
     remaining cards are placed face down in a pile.</p>
94
95   <p>When using a a standard deck, Kings score 13, Queens 12 and Jacks 11.</p>
96
97   <p>Play begins by turning over the top card in the face down pile.  Players try
     to match pairs of cards to a score of 13 (or the King, which itself is worth
     13).  When a pair is identified the cards are removed from the game.  If no pair
     is possible the turned card is placed face up in a discarded cards pile next to
     the face down pile. A pair may be made from any visible card in the pyramid, the
     turned card or the last discarded card.</p>
98
99   <p>When the pile is complete the discarded cards are turned over and becomes the
     new pile.</p>
100
101  <p>The game is scored by counting the number of cards left in the pyramid at the
     end of the game. A perfect score is 0 as the whole pyramid has been removed.</p>
102  </section>
103
104
105
106  </main>
107
108  <!-- Page Footer -->
109  <footer>
110  <p> &copy; Card Foundation <br> 2017 <br> Please provide feedback to:
     jlongridge@jlinternet.co.uk </p>
```

```
111  </footer>
112
113  </body>
114  </html>
```

04/03/2019 Patience

Playing Cards

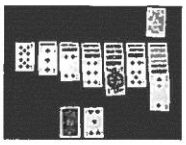
Home History Multi-player Single-player Free Cards

Popular Patience Games

Click on the names below to read more about each game of patience.

KLONDIKE FREECELL PYRAMID

Klondike



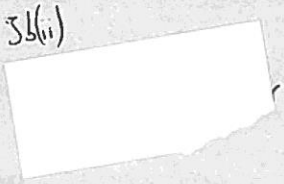
Klondike is the best known of the patience games. It was invented in the late 19th century and was named after the place Klondike, the gold rush area in Canada. Some think it may have been invented by the gold prospectors of the time.

Klondike is played by laying 7 piles of cards face down with each pile having one more card than the pile to its left. The top card of each pile is turned face up. The remaining cards are placed face down in a separate playing pile.

The player must build new 4 piles (one for each suit) starting with the ace, then 2, then 3 and so on. The piles are created as the next cards required to build each pile are discovered.

[illegible] Card Foundation
2017
Please provide feedback to: jlongridge@jlinternet.co.uk

3b(ii)



file://lt-hol-ad-1/080501153$/Documents/Computing%20Science/S5/Assignment/Content/Website%20Design/HTML/patience.html 1/1

04/03/2019 Patience

Playing Cards

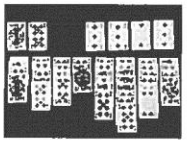
Home History Multi-player Single-player Free Cards

Popular Patience Games

Click on the names below to read more about each game of patience.

KLONDIKE FREECELL PYRAMID

Freecell



FreeCell is played using the standard 52-card deck. Compared to most patience games it is very easy to complete the game. All cards are dealt face-up at the beginning of the game.

FreeCell has been included as part of Windows since 1995 making it very popular. Many other platforms have attempted to replicate the MicroSoft version as players don't tend to like variations from that version.

The game begins with four open cells and four foundations. A standard 52 card deck of cards is used.

Cards are dealt face-up into eight columns positioned so that every card can be seen. Four columns have seven cards and four have six cards.

The top card of any colum can be covered with a card one lower in value and the opposite colour. Foundations are built up by suit.

The four cells can be used to temporarily store cards while moving other cards around.

The game is complete when all four foundation piles are complete from Ace to King in their correct suits.

[redacted] Card Foundation
2017
Please provide feedback to [redacted]@btinternet.co.uk

3b(ii)

[redacted]

file://lt-hol-ad-1/080501153$/Documents/Computing%20Science/S5/Assignment/Content/Website%20Design/HTML/patience.html 1/1

04/03/2019 Patience

Playing Cards

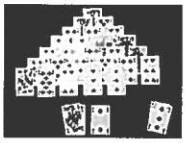
Home History Multi-player Single-player Free Cards

Popular Patience Games

Click on the names below to read more about each game of patience.

KLONDIKE FREECELL PYRAMID

Pyramid



The object is to remove all the cards from the pyramid to the foundation.

The game begins by laying out a pyramid of cards. One card at the top covered by two cards on the next row down, covered by three cards on the next row and so on up to the last row of seven cards (28 cards in total). The remaining cards are placed face down in a pile.

When using a a standard deck, Kings score 13, Queens 12 and Jacks 11.

Play begins by turning over the top card in the face down pile. Players try to match pairs of cards to a score of 13 (or the King, which itself is worth 13). When a pair is identified the cards are removed from the game. If no pair is possible the turned card is placed face up in a discarded cards pile next to the face down pile. A pair may be made from any visible card in the pyramid, the turned card or the last discarded card.

When the pile is complete the discarded cards are turned over and becomes the new pile.

The game is scored by counting the number of cards left in the pyramid at the end of the game. A perfect score is 0 as the whole pyramid has been removed.

© Card Foundation
2017
Please provide feedback to: jlongridge@jlinternet.co.uk

3b (ii)

file://lt-hol-ad-1/080501153$/Documents/Computing%20Science/S5/Assignment/Content/Website%20Design/HTML/patience.html 1/1

Question 3(c)

3c The 'Free Cards' page contains a form that users can fill in if they wish to register for a free pack of playing cards.

Open the 'register.html' page in a suitable editor and examine the form code carefully.

Describe how all the form's inputs could be comprehensively tested.

(3 marks)

- The data entered is stored on a database correctly
- The website displays a message and doesn't send the form if a require field isn't filled in.
- Make sure the fields only allow a certain data type, e.g.: number of years should be a real number
- Make sure the max field length is able to support longer names
- Ensure that the "Form completed" message appears on screen after the form has been submitted
- Make sure that the input validation works correctly

Candidate name_____ Candidate number_____