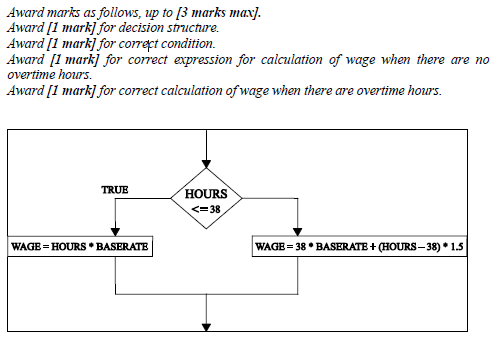
**SL Unit 4** **– Problem Solving**  
Quiz 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 1** | | | |
| Objectives: | 4.2.6 | Exam Reference: | Nov-14 7 |

When the wages for company employees are calculated, all hours above 38 are paid at   
the overtime rate of 1.5 times the base rate.

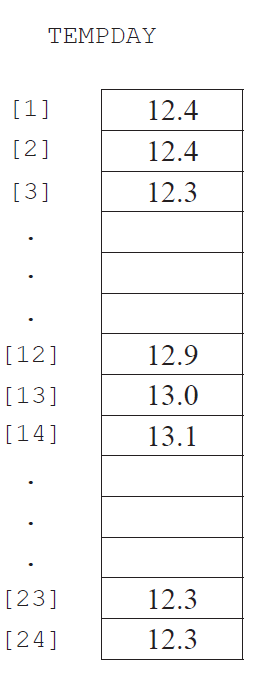
Construct a flowchart that represents this algorithm. [3]



C

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 2** | | | |
| Objectives: | 4.2.6 | Exam Reference: | Nov-14 10.a.b.c |

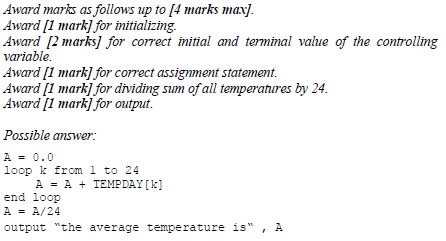
1. The temperature of a lake for one day is recorded every hour and data is stored in a one-dimensional array named TEMPDAY.



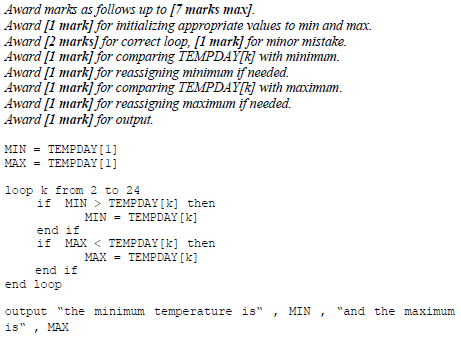
1. State the temperature of the lake at noon. [1]

12.9

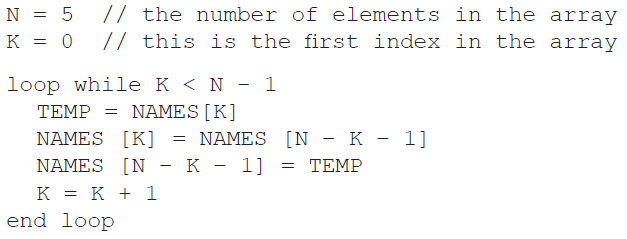
1. Construct an algorithm that will calculate and output the average temperature. [4 ]

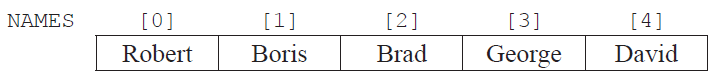


Construct an algorithm to find and output the minimum and maximum   
temperatures for the day. [7]

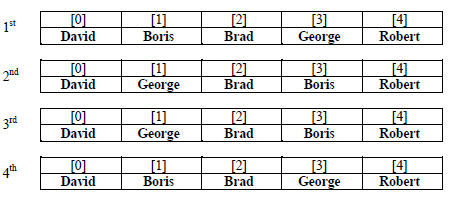


|  |  |  |  |
| --- | --- | --- | --- |
| **Question 3** | | | |
| Objectives: | 4.2.5 | Exam Reference: | Nov-14 8 |



Consider the following array  
  
  
  
  
and the following algorithm, which is constructed to reverse the contents of the array NAMES.

Trace the algorithm, showing the contents of the array after each execution of the loop. [2]



1. Identify the type of error that occurs. [1]

Logic error;

1. *Outline why the error occurs and how it could be corrected.* [2]

*Award* ***[1 mark]*** *for stating a possible cause of error.*

Loop executes too many times;

Terminating value for controlling variable was not correctly set;

*Award* ***[1 mark]*** *for stating a possible solution.*

Condition should be changed to k = n div 2;

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 4** | | | |
| Objectives: | 4.2.1, 4.2.3, 4.2.6 | Exam Reference: | May-16 13 |

A local charity organizes a half-marathon to raise money. The rules to participate in the  
half-marathon are as follows:

* The organizers limit the total number of participants to 450
* Participants belong to a team and each team must have at least three and at most five
* participants
* Each participant registers for the event independently from the other members of their
* team, and they all declare their team name when registering
* For scoring, the team’s final time is the sum of the times of its three fastest participants.  
  Participants that do not cross the finishing line within 2 hours after the start, are assigned  
  a default time of 1000 minutes. The **winning team** is the team with the smallest sum total.

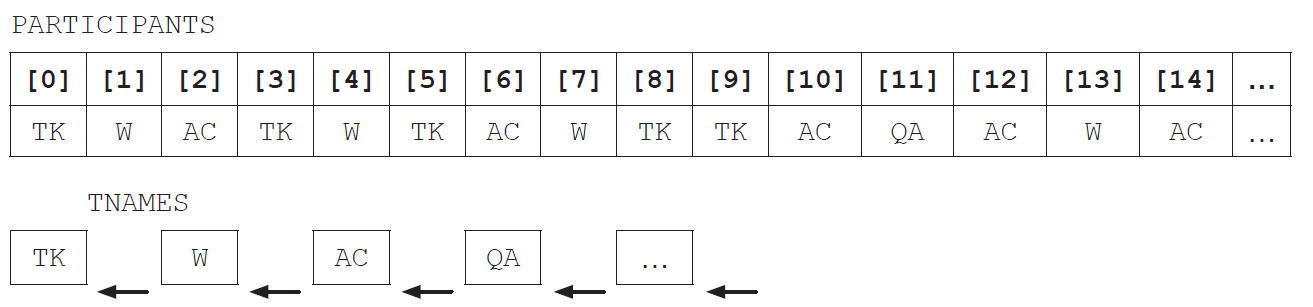
During registration, an array, PARTICIPANTS, with 450 positions is used to hold the  
 abbreviated team names that are declared by each participant. Simultaneously, a collection  
 TNAMES is generated: any new team name that is declared is added to the collection.

1. State the minimum size of TNAMES to ensure the names of all potential teams can

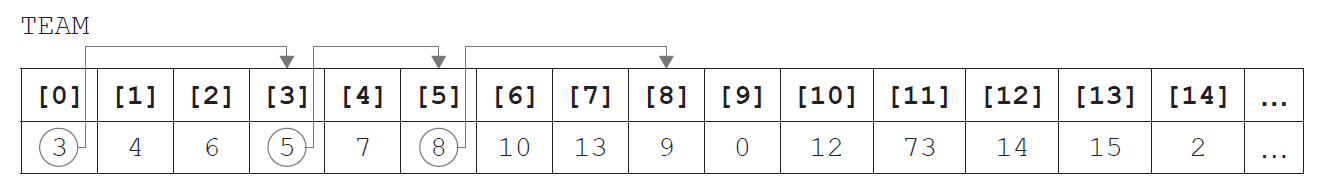
be stored. [1]

150 (= 450/3);

Part of the array PARTICIPANTS is shown below, where, for example, the first participant  
declared that they are part of team TK. The initial part of the collection TNAMES is also shown,  
with arrows indicating the direction of growth.



Both PARTICIPANTS and TNAMES are used to construct the array, TEAM, that groups all  
participants who belong to the same team. Part of the array TEAM is shown below.



In TEAM, each element is related to one other index in the array, shown by the arrows on the  
above diagram. This relation will eventually form a closed path (for this example 0, 3, 5, 8, 9  
and back to 0). The relation reflects the information in PARTICIPANTS, by grouping people  
who declared the same team name during registration.

Hence, participants 0, 3, 5, 8 and 9 are on the same team and, from PARTICIPANTS, that team  
is TK.

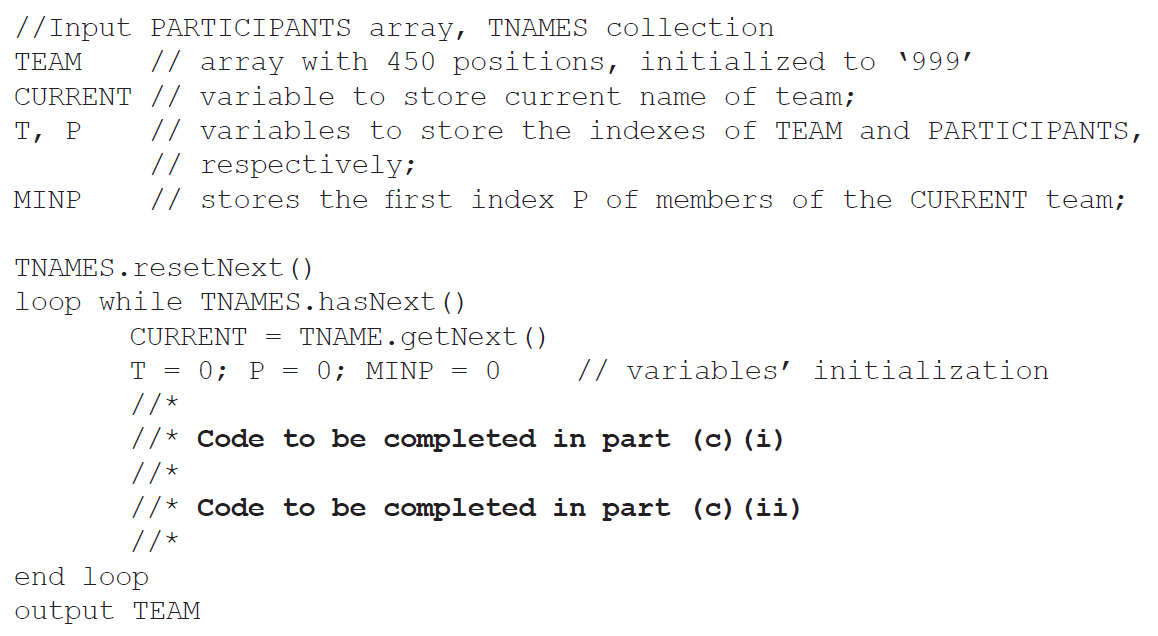
1. Identify the position in PARTICIPANTS of the second participant that registered for

team QA. [1]

73;

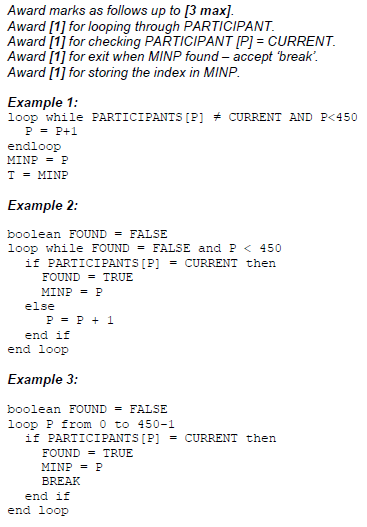
***Note:*** *Accept PARTICIPANTS[73]. Do not accept TEAM[11] because the question asks specifically for the array PARTICIPANTS.*

Part of the algorithm that generates the TEAM array is shown below, in pseudocode.

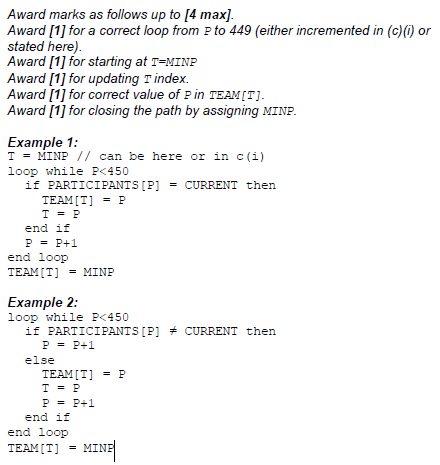


1. In order to complete this code, and return the correct TEAM array,
2. construct pseudocode to find MINP, the first index in PARTICIPANTS of the

CURRENT team, and use it to start the construction of TEAM [3]



1. construct pseudocode to find the other participants belonging to the CURRENT  
   team, implementing the idea of the closed paths in the TEAM array. [4]



As part of the program to determine the winning team, an array, TIMING, is maintained in  
parallel to PARTICIPANTS. For example, TIMING[5] and PARTICIPANTS[5] relate to the  
same participant.

TIMING is initialized to zero before the race starts, and updated with the finishing times for  
each participant. The algorithm sum3best is able to output the sum of the three fastest times  
from any group of times that are passed to the algorithm.

1. Describe the steps of an algorithm that will find the **winning team**, as defined by the

marathon rules on page 6. Clearly mention the use of existing or of new data structures. [6]

