IB Computer Science

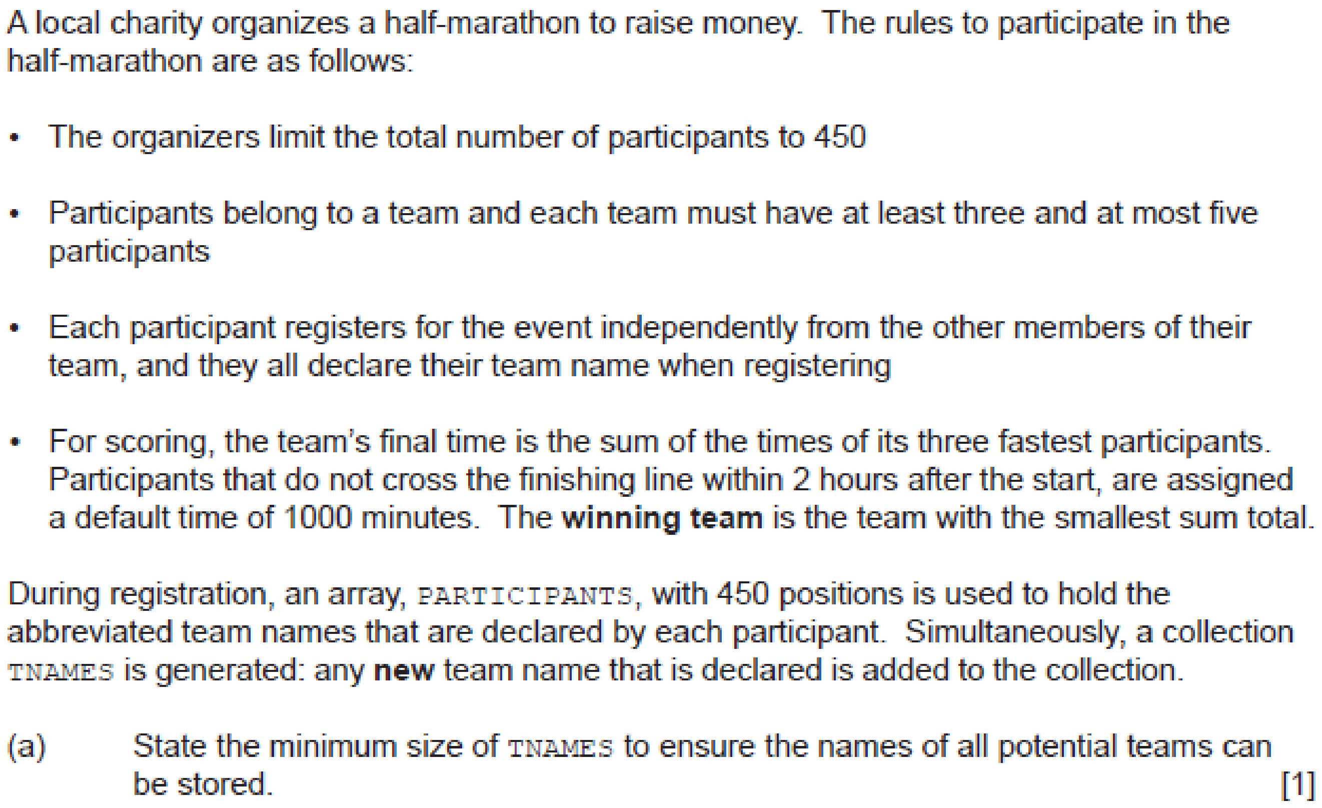
Student_______________

Collections

Date: _________________

Directions:

You many use the Collections handouts to answer each part of this question. This practice question was from the 2016 SL Paper 1. You may check your answers with a partner but each student should complete each part of the question.



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.

(a) State the minimum size of `TNAMES` to ensure the names of all potential teams can be stored. [1]

IB Computer Science

Student________________

Collections

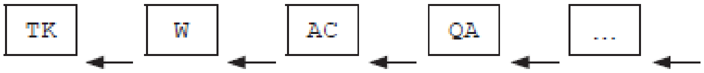
Date: __________________

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.

PARTICIPANTS

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	...
TK	W	AC	TK	W	TK	AC	W	TK	TK	AC	QA	AC	W	AC	...

TNAMES



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.

TEAM

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	...
3	4	6	5	7	8	10	13	9	0	12	73	14	15	2	...

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.

(b) Identify the position in PARTICIPANTS of the second participant that registered for team QA. [1]

IB Computer Science

Student_______________

Collections

Date: _________________

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

```
//Input PARTICIPANTS array, TNAMES collection
TEAM    // array with 450 positions, initialized to '999'
CURRENT // variable to store current name of team;
T, P    // variables to store the indexes of TEAM and PARTICIPANTS,
        // respectively;
MINP    // stores the first index P of members of the CURRENT team;

TNAMES.resetNext()
loop while TNAMES.hasNext()
       CURRENT = TNAME.getNext()
       T = 0; P = 0; MINP = 0     // variables' initialization
       //*
       //* Code to be completed in part (c)(i)
       //*
       //* Code to be completed in part (c)(ii)
       //*
end loop
output TEAM
```

(c) In order to complete this code, and return the correct TEAM array,

(i) construct pseudocode to find MINP, the first index in PARTICIPANTS of the CURRENT team, and use it to start the construction of TEAM [3]

(ii) 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.

(d) 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]