

EECE 230 Introduction to Programming, Sections 1 and 2

Quiz I

March 10, 2014

- The duration of this exam is 1 hour and 55 minutes.
- It consists of 3 problems.
- The exam is open book. You can use also all the material on Moodle: lecture notes, programming assignments, and solutions, etc. You are **NOT** allowed to use the **web** (**imail** included), You are not allowed to use **USB's** or files previously stored in your **account**.
- If you violate the above rules or if you communicate with a person other than the exam proctors during the exam, you will immediately get zero and you will be referred to the appropriate disciplinary committee.
- Active cell phones and any other unauthorized electronic devices are absolutely not allowed in the exam rooms. They should be turned off and put away.
- Plan your time wisely. Do not spend too much time on any one problem. Read through all of them first and attack them in the order that allows you to make the most progress.
- Submit your solutions each part in a separate file as indicated in the booklet. Include your name and ID number in each file. Submit the files online in a single zip file called *abcMN.zip*, where *abcMN* is your AUB user i.d., e.g., *abc01.zip*.
- Good luck!

Problem 1 (25 points). Divisibility by 2 and/or 3

Write a program which prompts the user to enter an integer n , and checks if

- n is divisible by 2 and 3, or
- n is divisible by 2 or 3 but not both, or
- n is neither divisible by 2 nor by 3.

Examples:

```
Please enter an integer: 24
24 is divisible by 2 and 3
```

```
Please enter an integer: -15
-15 is divisible by 2 or 3 but not both
```

```
Please enter an integer: 25
25 is neither divisible by 2 nor by 3
```

```
Please enter an integer: 1
1 is neither divisible by 2 nor by 3
```

Submit your solution in a file called Prob1.cpp including your name and ID number.

Problem 2 (25 points). Check if sorted

Write a program which prompts the user to enter a positive integer n followed by a list of n integers. Your program is supposed to check whether or not the input list is sorted in nondecreasing order. Your program is supposed to display an error message if n is less than or equal to zero.

Examples:

```
Please enter the number of integers: 8
Please enter 8 integers: 3 -4 -3 5 2 0 0 -1
Not sorted
```

```
Please enter the number of integers: 8
Please enter 8 integers: -3 0 1 4 5 12 12 19
Sorted
```

```
Please enter the number of integers: -2
Number of integers less than or equal to zero!
```

Any correct solution of this problem is worth 20/25 points. To get full grade you are supposed to solve it without using arrays.

Submit your solution in a file called Prob2.cpp including your name and ID number.

Problem 3 (60 points+10 bonus points). Partitionable lists

We call a list of integers *partitionable at position i* if the first i integers in the list are strictly smaller than the remaining integers in the list. More precisely, we call a list of n integers $a_0, a_1, \dots, a_{n-1}$ *partitionable at position i* , where $0 < i < n$, if for each j such that $0 \leq j < i$, and each k such that $i \leq k < n$, we have $a_j < a_k$.

Examples:

- $\langle 7, 1, 9, 8, 20, 17 \rangle$ is partitionable at position 2
- $\langle 7, 1, 9, 8, 20, 17 \rangle$ is also partitionable at position 4
- $\langle 7, 1, 9, 8, 20, 17 \rangle$ is NOT partitionable at position 1 (since $7 \geq 1$).
- $\langle -2, 3, 5, 20 \rangle$ is partitionable at all positions (at position 1: $\langle -2, 3, 5, 20 \rangle$; at position 2: $\langle -2, 3, 5, 20 \rangle$; and at position 3: $\langle -2, 3, 5, 20 \rangle$). More generally, a length- n list sorted in increasing order is partitionable at all positions (positions $1, 2, \dots, n-1$).
- $\langle -2, 3, 5, 20, -3 \rangle$ is NOT partitionable at any position.
- $\langle 0, 2, 0, 1 \rangle$ is NOT partitionable at any position.

- a) **(25 points) Check if partitionable at the second position.** Write a program which prompts the user to enter a list of integers whose end is the indicated by the sentinel -999. Your program is supposed to check whether or not the list is partitionable at position 2. (i.e., check whether or not the first two integers in the list are strictly smaller than the remaining integers in the list).

Your program is supposed to give a YES/NO answer. If the number of the elements in the list is less than 3, your program is supposed to return an error message.

Examples:

- $\langle 7, 1, 9, 8, 20, 17 \rangle$ is partitionable at position 2
- $\langle 7, 10, 9, 8, 20, 17 \rangle$ is NOT partitionable at position 2 (since $10 \geq 9, 8$)
- $\langle 0, 2, 0, 1 \rangle$ is NOT partitionable at position 2 (since $0 \geq 0$, also $2 \geq 0, 1$)

Any correct solution is worth full grade.

Submit your solution in a file called Prob3a.cpp including your name and ID number.

- b) **(35 points + 10 bonus points) Check if partitionable.** Write a program which prompts the user to enter a list of integers whose end is the indicated by the sentinel -999. Your program is supposed to check whether or not the list is partitionable at some position (i.e., check whether or not the list is partitionable at some position i for some $0 < i < n$, where n is the list length).

Here again, your program is supposed to give a YES/NO answer. If the number of the elements in the list is less than 2, your program is supposed to return an error message.

Examples:

- $\langle 7, 1, 9, 8, 20, 17 \rangle$ is partitionable
- $\langle -2, 3, 5, 20 \rangle$ is partitionable
- $\langle -2, 3, 5, 20, -3 \rangle$ is NOT partitionable
- $\langle 0, 2, 0, 1 \rangle$ is NOT partitionable

Any correct solution is worth 25/35 points. Faster solutions are worth more points. If you solve it without nested loops, you will get 45/35.

Submit your solution in a file called Prob3b.cpp including your name and ID number.