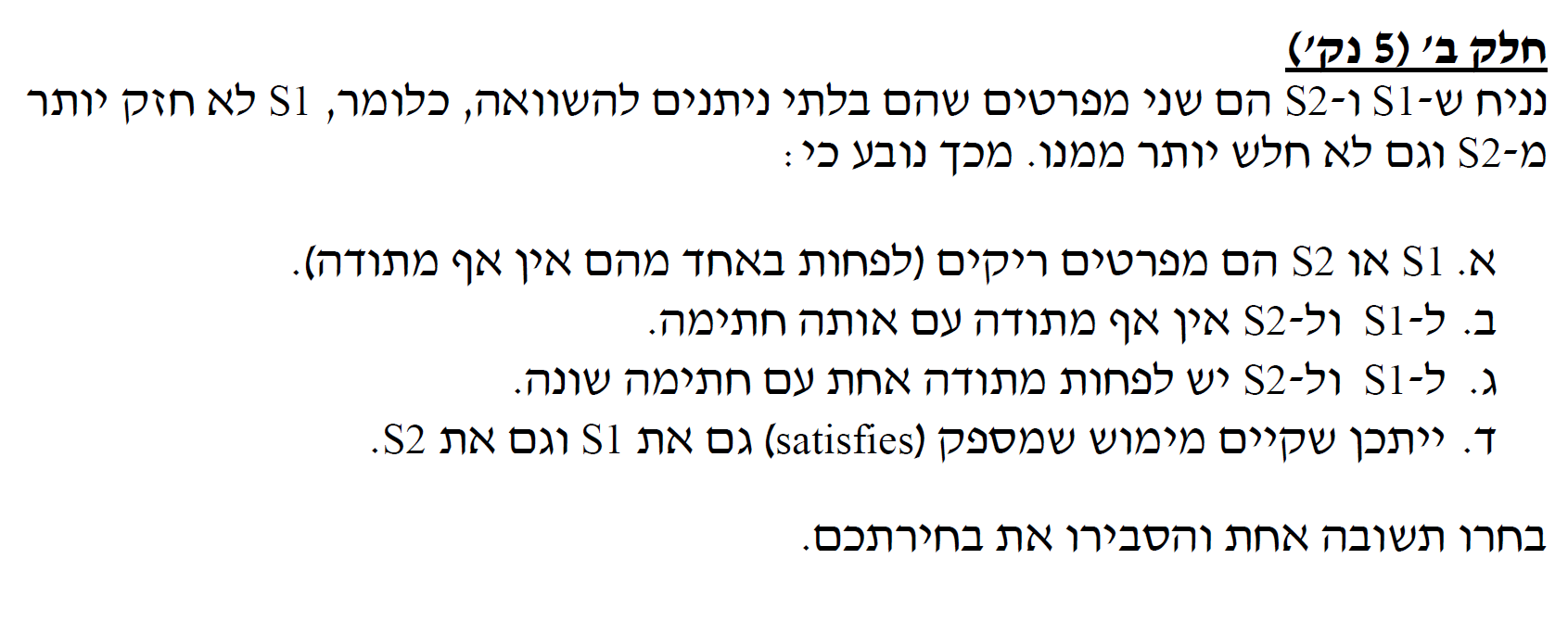


לא ניתן להגדיר יחס חוזק בין שתי המפרטים כיוון שהם בלתי ניתנים להחלפה. 1S לא יכול להחליף את 2S כיוון שהוא לא מספק את אותה רמת פירוט בפלט (אם 2S התחייב ספק מכונית אדומה לדוג' 1S יספק מכונית אך לא יתחייב על הצבע). S2 לא יכול להחליף את 1S כיוון שהוא דורש יותר בקלט, (לדוג' אם 2S חייב מערך ממוין על מנת לספק תוצאה ל1S ואלידית, ל1S ניתן להעביר מערך לא ממוין).

\



ד. ייתכן וקיים מימוש שמספק את שניהם.

כיוון שהתשובה היא ייתכן וקיים ולא בהכרח קיימת נראה ע"י דוג:

**S1**:

@requires: a sorted array, all numbers in the array must be positive

@returns: the largest number in the array in absolute value

Int maxValue(int[] array);

**S2**:

@requires: a sorted array, all numbers in the array must be negative

@returns: the largest number in the array in absolute value

Int maxValue(int[] array);

**S3**:

@requires: nothing

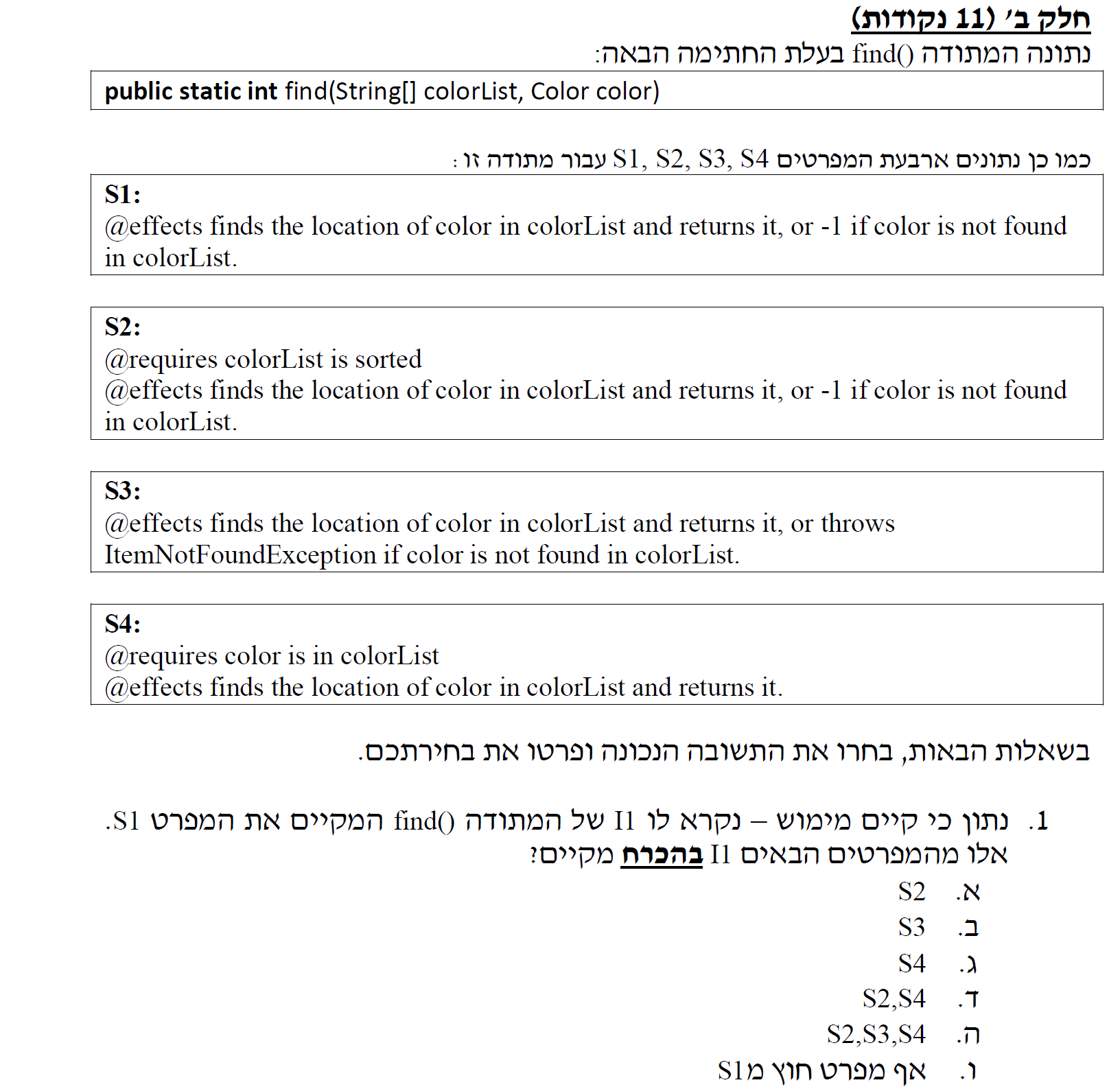
@returns: the largest number in the array in absolute value

Int maxValue(int[] array);

מפרטים 1 ו2 בלתי ניתנים להחלפה כיוון שכך אחד מהם דורש שכל המספרים במערך יהיו חיוביים/ שליליים בהתאמה.

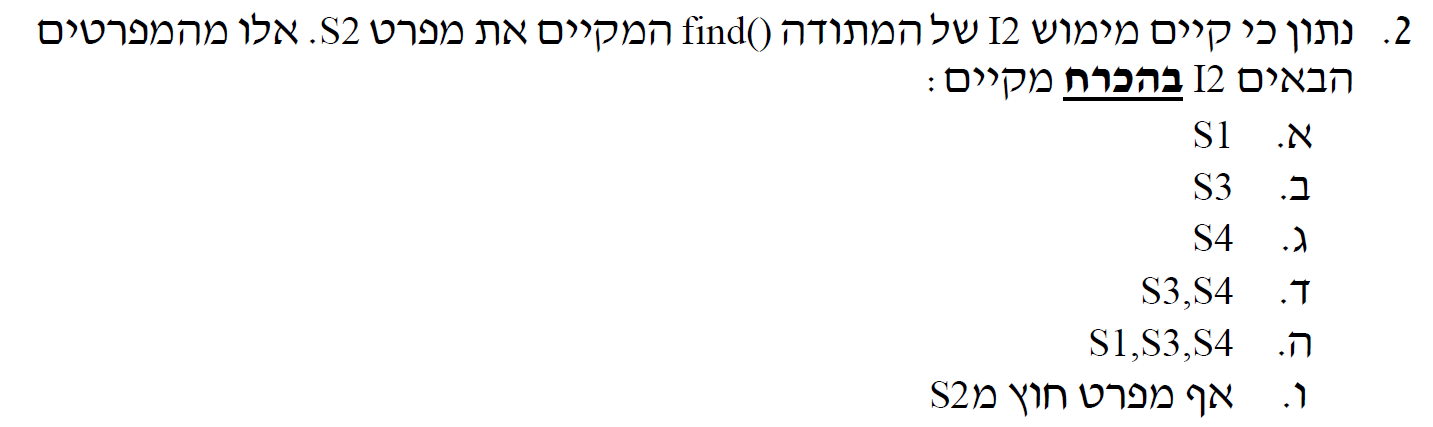
אם זאת, מפרט 2 לא דורש כלום ולכן מקיים את הדרישות של שתי המפרטים באופן ריק, ומחזיר את אותה תוצאה ולכל מקיים גם את התחייבות הקלט.

לכן מפרט 3 מספק את שני המפרטים ע"פ הגדרת סיפוק מפרטים.



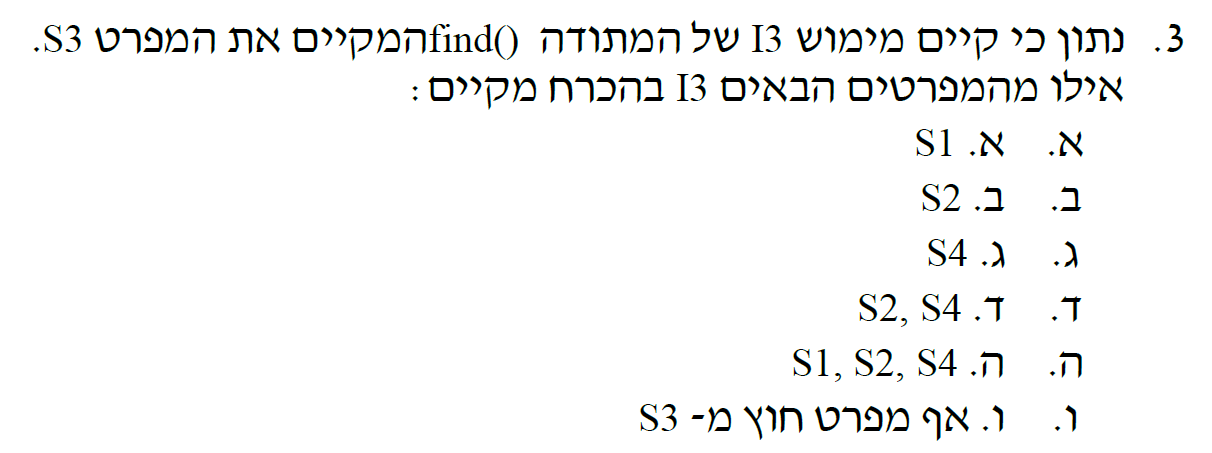
**המימוש הנ"ל מקיים גם את המפרטים 2 ו4. הסבר:**

* 2: מפרט 2 דורש מהמערך להיות ממוין, מפרט אחד לא דורש כלום (ולכן בהכרח יכול לקבל מערך ממוין). שני המפרטים מחזירים בדיוק אותו דבר ולכן מפרט 1 חזק מ2 וכל מימוש שמקיים אותו יכול להחליף את 2.
* 3: שני המפרטים אומנם לא דורשים כלום אולם אם הצבע לא נמצא במערך מפרט אחד יחזיר 1- בעוד שמפרט 3 יזרוק חריגה. לכן קיימת התנהגות מובטחת שונה בין שני המפרטים והם לא ניתנים להחלפה.
* 4: בדומה למפרט 2, מפרט 1 מקיים את דרישות הקלט של מפרט 4, כמו כן מפרט 4 מתחייב להחזיר את מיקום הצבע במערך, אם הצבע לא קיים ההתנהגות לא מוגדרת. מפרט 1 מקיים דרישה וזו ומפרט מגדיר את ההתנהגות כאשר הצבע לא קיים ולכן ניתן להחליף את מפרט4 במימוש של מפרט 1.



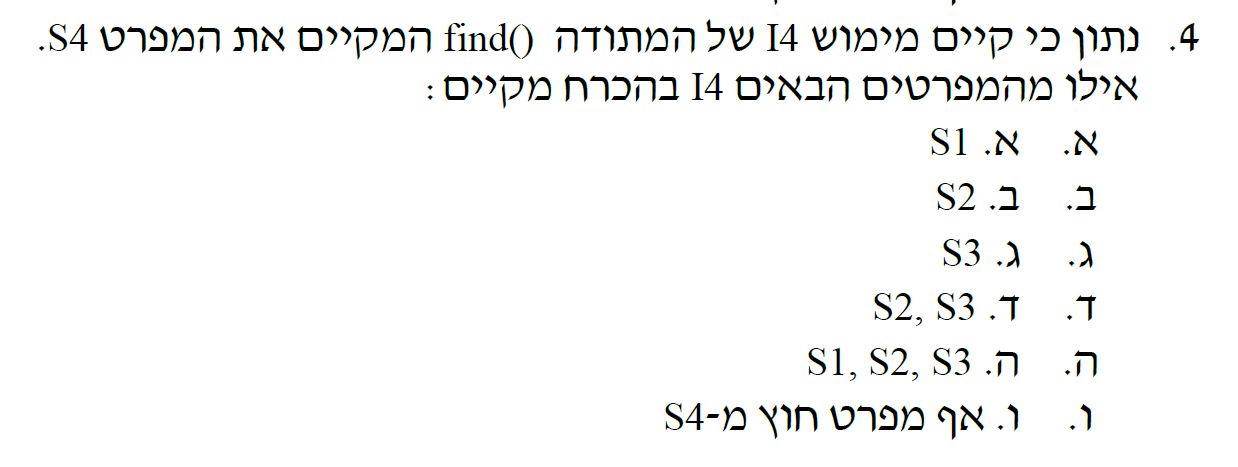
**המימוש הנ"ל לא מקיים אף מפרט פרט ל-2. הסבר:**

* 1: מפרט 1 לא דורש כלום בעוד שמפרט 2 דורש שהמערך יהיה ממוין, ולכן מימוש של 2 לא יכול להחליף את 1.
* 2: מפרט 3 זורק חריגה במידה והצבע לא נמצא במערך, מפרט 2 מחזיר 1- במקרה זה ולכן הם לא ברי החלפה.
* 4: מפרט 4 דורש שהצבע יהיה במערך בעוד שמפרט 2 דורש שהמערך יהיה ממוין. למפרט 4 ניתן להעביר מערך לא ממוין ולכן מימוש של 2 לא בהכרח יוכל להחליף אותו.



**המימוש הנ"ל מקיים רק את מפרט 4. הסבר:**

* 1: מפרט 3 זורק חריגה במידה והצבע לא נמצא במערך, מפרט 1 מחזיר 1- במקרה זה ולכן הם לא ברי החלפה.
* 3: מפרט 3 זורק חריגה במידה והצבע לא נמצא במערך, מפרט 2 מחזיר 1- במקרה זה ולכן הם לא ברי החלפה2
* 4: דורש שהצבע יהיה במערך ויחזיר את מיקומו, מפרט 3 לא דורש כלום ולכן יעבוד בפרט אם הצבע במערך. מפרט 3 יזרוק חריגה במקרה והצבע לא במערך אך התנהגות זו לא מוגדרת ע"י מפרט 4 ולכן מקרה ספציפי בהכרח יספק דרישה להתנהגות לא מוגדרת (שהיא דרישה ריקה). על כן מימוש ל3 מספק את מפרט 4.



המימוש הנ"ל לא מקיים אף מפרט פרט למפרט 4. הסבר:

* 1: מפרט 1 מחזיר 1- כאשר הצבע לא נמצא במערך, במפרט 4 לא מוגדרת התנהגות למקרה זה ולכן הם לא ברי החלפה.
* 2: מפרט 2 מחזיר 1- כאשר הצבע לא נמצא במערך, במפרט 4 לא מוגדרת התנהגות למקרה זה ולכן הם לא ברי החלפה.
* 3: מפרט 3 זורק חריגה כאשר הצבע לא נמצא במערך, במפרט 4 לא מוגדרת התנהגות למקרה זה ולכן הם לא ברי החלפה.