

במדינה אנונימית התלוננו על זיוף תוצאות הבחירות, ולכן הוחלט למחשב את מערכת ניהול הבחירות.

הבחירות במדינה מתנהלות כך (קיימת דוגמא מפורטת בסיום התרגיל):

- המדינה מחולקת למחוזות, כאשר לכל מחוז משויכת כמות של נציגים בהתאם לגודלו (לדוגמא: למחוז NY יש 29 נציגים).
- לכל מפלגה המתמודדת בבחירות יש מועמד יחיד לראשות המדינה, ורשימת נציגים נפרדת לכל מחוז במדינה. כל אזרח מצביע במחוז בו הוא מתגורר לאחת המפלגות המתמודדות בבחירות.
- זהות הנציגים בכל מחוז נקבעת לפי רשימת כל מפלגה במחוז, בהתאם ליחס הקולות שקיבלה (לדוגמא: עבור מפלגה R עם חצי מהקולות במחוז KS עם 6 נציגים, יבחרו 3 הנציגים הראשונים ברשימת המפלגה R למחוז KS, ושאר הנציגים במחוז יהיו לפי שאר המפלגות).
- בסיום הבחירות, כל מחוז משויך למפלגה שקיבלה את רוב ההצבעות במחוז. ראש המדינה הנבחר הוא זה שזכה במחוזות עם סכום הנציגים הגדול ביותר.

לכל סבב בחירות יש לשמור את הנתונים הבאים:

- תאריך בו מתקיים סבב הבחירות (יום, חודש, שנה).
- פנקס הבוחרים: רשימת האזרחים הרשאים לבחור. לכל אזרח יש לשמור את שמו, מספר תעודת הזהות שלו, שנת הלידה שלו והמחוז אליו הוא משויך.
- רשימת המפלגות המועמדות: לכל מפלגה יש לשמור את שמה, את מועמד המפלגה לראשות המדינה, ואת רשימת הנציגים מטעם המפלגה לכל מחוז, ממוינת לפי סדר. כל נציג הוא אזרח.
- מחוז: רשימת המחוזות במדינה. לכל מחוז יש שם, מספר סידורי, רשימת אזרחים הרשאים להצביע במחוז, אחוז המצביעים מכלל האזרחים, ותוצאות הבחירות במחוז: איזה נציגים נבחרו מכל מפלגה.

בתרגיל זה יש לכתוב תפריט אינטראקטיבי החוזר על עצמו, ומאפשר את הפעולות הבאות:

1. הוספת מחוז (קלט: שם המחוז, מספר הנציגים במחוז).
2. הוספת אזרח (קלט: שם, מספר זהות, שנת לידה, מספר מחוז).
3. הוספת מפלגה (קלט: שם מפלגה, מספר זהות מועמד לראשות המדינה).
4. הוספת אזרח כנציג מפלגה (קלט: מספר זהות הנציג, מספר מפלגה, מספר מחוז).
5. הצגת כל המחוזות (פלט: מספר מחוז, שם מחוז, מספר הנציגים).
6. הצגת כל האזרחים (פלט: שם, מספר זהות, שנת לידה, מחוז).
7. הצגת כל המפלגות (פלט: מספר ושם מפלגה, מועמד לראשות המדינה, נציגים לכל מחוז).
8. הצבעה: יש לקלוט מספר זהות ומספר מפלגה, ולהוסיף הצבעה של האזרח למפלגה במחוז הרלוונטי. יש להימנע מהצבעות כפולות ולהודיע על כך למשתמש.
9. הצגת תוצאות הבחירות: לכל מחוז יש להציג את שם המחוז, מספר הנציגים שהמחוז מעניק ומועמד המפלגה אליה המחוז משויך. כמו כן יש להציג לכל מחוז את רשימת הנציגים שנבחרה מכל מפלגה, מספר ואחוז הקולות שקיבלה כל מפלגה, ואחוז ההצבעה במחוז. לבסוף יש להציג עבור כל מפלגה את המועמד שלה לראשות המדינה, סכום הנציגים שזכה בהם וסכום הקולות שקיבלה המפלגה בכל המחוזות. הצגת המועמדים תתבצע בסדר יורד של מספר נציגים (כלומר – המועמד הראשון הוא זה שנבחר לראשות המדינה).
10. יציאה מהתוכנית.

הנחיות הגשה

1. העבודה על התרגילים היא בזוגות **בלבד**. ניתן להגיש עם שותף מקבוצה אחרת.
2. את התרגילים יש להגיש במאמא, על-ידי אחד השותפים **בלבד**.
3. יש להגיש קובץ zip עם קבצי הקוד שלכם (cpp, h), ובנוסף קובץ **readme** עם פרטי המגישים (שם + ת"ז). אין להגיש קבצים נוספים.
4. יש להשתמש אך ורק בנושאים שנלמדו בכיתה (הרצאות 1 עד 5).

הנחיות כלליות

1. המספר הסידורי של כל מפלגה ומחוז יינתן באופן אוטומטי. במקרים של תיקו יש להכריע לטובת המפלגה עם המספר הסידורי הקטן יותר.
 2. יש לממש או לבטל בנאי העתקה ואופרטור השמה לכל מחלקה בהתאם ללוגיקה שהחלטתם.
 3. יש להוסיף אופרטור הדפסה עבור אזרח, מחוז, מפלגה ופנקס הבוחרים.
 4. שדות שהם מחרוזת יוקצו דינמית ויחזיקו גודל מדויק (**לא** להשתמש ב-strdup אלא לממש לבד). בקליטת מחרוזות ניתן להניח גודל מקסימלי (אך עדיין לשמור גודל מדויק במחלקה).
 5. הגדלה של מערכים תהיה פי 2 כל פעם. לכן לכל מערך דינמי יש להחזיק גודל לוגי וגודל פיזי.
 6. אין לבצע קלט במחלקות הפנימיות. ה-main הוא זה שקולט את המידע, מדפיס למשתמש, שולח ערכים למחלקות ושיטות, וכו'.
 7. עליכם להחליט כיצד התוכנית תיראה – חלוקה למחלקות, תפריט, תת-תפריטים, וכו'. הגיוני כי כל תרגיל יהיה שונה. אתם נבדקים על איכות ועיצוב הקוד, ולא על פרשנות מינורית כזו או אחרת של איך בדיוק הדברים צריכים להיראות.
 8. בדיקות תקינות:
- א. אין להמציא הגבלות על הקלט מעבר למה שנתון. יש להקפיד שמתודות ה-setter יחזירו bool כתשתית לבדיקת תקינות עתידית.
- ב. ניתן להניח שהקלט המתקבל הוא מהטיפוס המבוקש (כלומר, אם הקלט הצפוי הוא מספר אז ניתן להניח שאכן הוקלד מספר). אם יש הגבלות לטווח ערכים אפשרי, לא ניתן להניח שהערך שנקלט הוא בטווח ויש לוודא את זה ולהודיע על שגיאה בהתאם.
- ג. אין צורך לטפל בקלט שגוי בבנאי, אלא ברמת ה-main כמפורט בסעיף ב'.
- ד. אין צורך לבדוק הצלחה של הקצאות דינמיות וניתן להניח שכולן מצליחות.

דוגמא לבחירות

- במדינה יש 3 מחוזות ו-2 מפלגות: מחוז A עם 15 נציגים, מחוז B עם 10 נציגים, ומחוז C עם 7 נציגים. בכל מחוז 100 אזרחים. מפלגת **דמבו** עם הנציגה **הילי** לראשות המדינה, ומפלגת **איור** עם הנציג **דוני**.
- מחוז A: הצביעו 60 ל**דמבו** ו-40 ל**איור**. לכן יהיו 9 נציגים של מפלגת **דמבו** (9 הראשונים ברשימת הנציגים של **דמבו** למחוז A), ו-6 נציגים של מפלגת **איור**. המחוז מעניק את כל 15 הנציגים **להילי**.
- מחוז B: הצביעו 25 ל**דמבו** ו-25 ל**איור**. לכל מפלגה 5 נציגים. מדובר בתיקו, לכן המועמד הזוכה נבחר לפי המספר הסידורי של המפלגה, במקרה זה המחוז יעניק את 10 הנציגים **לדוני**.
- מחוז C: הצביעו 40 ל**דמבו** ו-50 ל**איור**. לכן יהיו 3 נציגים של **דמבו** ו-4 נציגים של **איור**. שימו לב לחלוקת הנציגים כמספר שלם לכל מפלגה, באופן יחסי. המחוז מעניק את 7 הנציגים **לדוני**.
- הזוכה בבחירות הוא **דוני** עם 17 נציגים לעומת 15 (שימו לב: אפילו שלא קיבל את מירב הקולות).

בעקבות הצלחת מערכת הבחירות הממוחשבת, הוחלט להרחיב את המערכת כך שתתאים למקרים נוספים ומדינות נוספות. בהתאם לכך יש לעדכן את הפונקציונליות של כל סבב בחירות.

הורשה ופולימורפיזם

בחלק זה נגדיר ישויות חדשות עבור המערכת, ואת השינויים הנדרשים בתפריט עבור ישויות אלו.

יש לעצב ולממש את הישויות עם מינימום שכפול קוד ובאופן מכון עצמים שיתמוך בשינויים ותוספות בעתיד (לדוגמא: את המחוזות יש לעצב כך שיהיה ניתן בקלות יחסית להוסיף מחוז חדש עם אופן אחר לחלוקת הנציגים). עליכם להשתמש בהורשה, הכלה, פולימורפיזם, ולהחליט היכן לממש מחלקות בסיס, מחלקות יורשות, מחלקות אבסטרקטיות, ולעדכן את המערכת הקיימת בהתאם. כרגיל, יש לעצב את הישויות באופן גנרי כקופסה שחורה ולא להסתמך על השימוש הספציפי בהן בתפריט התרגיל.

ראשית, נגדיר 2 סוגים של מחוזות אשר ניתן להוסיף לכל סבב בחירות:

1. **מחוז אחיד** – כפי שהוגדר בתרגיל 1, מחוז אשר מעניק את כל הנציגים שלו לזכות המועמד לראשות המדינה של המפלגה שזכתה במירב הקולות במחוז.
2. **מחוז מחולק** – מחוז אשר מעניק את הנציגים שלו למועמדים לראשות המדינה באופן יחסי, בהתאם למס' הנציגים בהם זכתה המפלגה במחוז (לדוגמא, אם במחוז מחולק זכתה מפלגת דמבו ב-2 נציגים ומפלגת איור ב-3 נציגים, לזכות המועמדת הילי יספרו 2 נציגים ולזכות המועמד דוני יספרו 3 נציגים עבור מחוז זה).

בהוספת מחוז (1) יש להוסיף קלט מהמשתמש עבור סוג המחוז (יכול להיות בוליאני).

בהצגת כל המחוזות (5) יש להציג גם את סוג המחוז.

בהצגת תוצאות הבחירות (9), עבור מחוז מחולק, במקום להציג את "המועמד לראשות המדינה של המפלגה אליה המחוז משויך" יש להציג את כל המועמדים להם המחוז מעניק נציגים כלשהם (יותר מ-0) ואת מספר הנציגים שהמחוז מעניק לכל מועמד, בסדר יורד.

שנית, נגדיר 2 סוגים של סבבי בחירות אשר המערכת תדע לנהל:

1. **סבב בחירות רגיל** – כפי שהוגדר בתרגיל 1.
2. **סבב בחירות פשוט** – ללא מחוזות. בסבב בחירות פשוט לכל מפלגה יש רשימת נציגים אחת ואזרחים אינם משויכים למחוז. מבחינת מימוש, ניתן להתייחס לסבב בחירות פשוט כסבב בחירות עם מחוז מחולק יחיד, אשר נוצר מאחורי הקלעים באופן אוטומטי ונסתר מהמשתמש.

עבור סבב בחירות פשוט:

- בהוספת מחוז (1) יש להציג הודעת שגיאה למשתמש ולהמשיך כרגיל. בהוספת אזרח ונציג (2,4) יש להתעלם מהקלט בתפריט למספר המחוז עבור האזרח ו/או נציג שהתווסף.
- בהצגת המחוזות (5) ניתן לבחירתכם: להציג הודעת שגיאה, להציג מחוז יחיד, או להציג פלט ריק. בהצגת כל האזרחים (6) והצגת כל המפלגות (7) ניתן לא להציג את המחוז או להציג מחוז יחיד (זהה) עבור כל האזרחים והמפלגות. לכל מפלגה נציג רשימת נציגים אחת.
- בהצגת תוצאות הבחירות (9) יש להדפיס את אחוז ההצבעה הכולל, ועבור כל מפלגה (בסדר יורד של מספר הקולות) יש להציג את מספר הנציגים שבו זכתה המפלגה, מספר ואחוז הקולות שקיבלה, רשימת הנציגים שנבחרו עבורה, ומועמד המפלגה לראשות המדינה.

סריאליזציה

בחלק זה נממש אפשרות לשמור את סבב הבחירות לקובץ ולטעון סבב בחירות קיים מקובץ. עליכם להוסיף סריאליזציה עבור **סבב בחירות** כך שיהיה ניתן לשמור את סבב הבחירות וכל הנתונים הרלוונטיים לקובץ ולטעון אותם מהקובץ, כפי שנלמד.

הנתונים יישמרו באופן בינארי ויש לדאוג לשמירה נכונה של כל הערכים, מצביעים, הקצאות דינמיות וכו', ולטעינה נכונה של הנתונים. בנוסף, יש לוודא שפעולות השמירה והטעינה עובדות נכון עם פולימורפיזם. ניתן להוסיף מחלקות חדשות לשם כך.

שימו לב: בכל קלט של שם קובץ, ניתן להניח ששם הקובץ אינו מכיל רווחים. המשתמש חופשי להכניס כל שם קובץ (ללא רווחים) ואין צורך לבדוק את סיומת הקובץ, ובפרט אין להניח סיומת כלשהי (כמו bin או dat), אלא לטעון עם שם הקובץ כפי שהתקבל.

שינויים בתפריט

בכניסה הראשונית לתוכנית, נציג למשתמש (לפני התפריט הראשי) את האפשרויות הבאות:

1. יצירת סבב בחירות חדש (קלט: תאריך, סוג סבב).
אין צורך לוודא את התאריך, ניתן להניח שהוכנסו 3 מספרים תקינים.
 2. טעינת סבב בחירות קיים (קלט: שם קובץ).
בכישלון, נדפיס הודעת שגיאה ונצא.
 3. יציאה.
- לאחר בחירת אפשרות 1 או 2, נעבור מיד לתפריט הראשי לניהול סבב הבחירות שנבחר. לא ניתן לחזור לתפריט הראשוני (אלא ע"י הפעלה מחדש של התוכנית).
- בהצגת תוצאות הבחירות (9), נדפיס ראשית (בכותרת) את התאריך של סבב הבחירות. בנוסף, נוסיף 2 אפשרויות חדשות לתפריט הראשי:
11. שמירת סבב בחירות (קלט: שם קובץ).
לאחר השמירה נחזור לתפריט הראשי ונמשיך כרגיל.
 12. טעינת סבב בחירות קיים (קלט: שם קובץ).
לאחר הטעינה נחזור לתפריט הראשי ונמשיך עם סבב הבחירות שנטען.

תיקונים

עליכם לבצע תיקונים בעיצוב ומימוש הקוד שלכם בהתאם למשוב של תרגיל 1, כאשר יתקבל המשוב ובהתאם להערות והפירוט שיתקבל. קריטריונים אלו יבדקו שוב בתרגיל 2.

שימו לב, המשוב לתרגיל 1 יתקבל כשבוע לפני מועד ההגשה של תרגיל 2 (אך נשתדל להקדים יותר). לכן, עליכם לדאוג לעצב ולממש את המחלקות עם ממשק פומבי מוגדר וכקופסה שחורה ככל האפשר, כך שתוכלו לבצע את התיקונים בקלות ועם מינימום שינויים.

הנחיות

עבור פתרון התרגיל יש להשתמש אך ורק בנושאים שנלמדו בכיתה (הרצאות 1 עד 8). מעבר לכך, הנחיות ההגשה וההנחיות הכלליות זהות לתרגיל 1.

בעדכון האחרון למערכת הבחירות הממוחשבת נוסף לכל **סבב בחירות** בדיקת שגיאות ושימוש בחריגות, יחד עם עדכון הקוד לשימוש בתבניות ומבני הנתונים והאלגוריתמים הסטנדרטיים של C++.

תבניות

בחלק זה נעדכן את הקוד כך שיעשה שימוש בתבניות.

עליכם לזהות אילו מחלקות מייצגות או מכילות מבני נתונים או אלגוריתמים כלליים, ולעדכן אותן לשימוש בתבניות. ניתן לייצר מחלקות תבנית חדשות ולהשתמש בהן. שימו לב: מחלקות אלו אינן צריכות להיות ספציפיות למערכת הבחירות ועיצוב המחלקות הכולל ייבדק. לדוגמא, בהינתן מערך דינמי יש ליצור מחלקת תבנית של מערך דינמי כללי, ולא ספציפי לסבב בחירות.

בחלק זה עליכם לממש לפחות אלגוריתם תבנית אחד. בנוסף, עליכם להשלים את `DynamicArray` מהרצאה 11 כך שתתמוך באיטרטורים ואלגוריתמים של STL באופן מלא ולהשתמש בה, או לממש מבנה נתונים אחר משלכם (עם שימוש בתבנית וכולל איטרטור והתאמה ל-STL).

כלומר, יש לממש: `iterator, const_iterator, begin/end, cbegin/cend, rbegin/rend, insert, erase x2`.

חריגות

בחלק זה נעדכן את המערכת לזיהוי וטיפול בשגיאות.

עליכם להמיר את כל בדיקות השגיאות לשימוש בחריגות, כך שכל שגיאה אפשרית תיבדק במקום הנכון ותיתפס (והטיפול בה יתבצע) במקום הראוי. בכל הקוד כולו אין לעשות שימוש בערכי החזר, אלא בחריגות בלבד. כמו-כן יש להוסיף בדיקות תקינות לכל הבנאים וזריקת חריגות בהתאם, היכן שיש צורך ומתאפשרת שגיאה בארגומנטים המתקבלים בבנאי.

יש לדאוג לשחרר זיכרון באופן נכון בכל מקרה של חריגה, ולבדוק ולטפל נכון גם בשגיאות של הקצאת זיכרון (וכמובן גם גישה לקבצים וכו'). כל חריגה צריכה להיתפס באיזשהו שלב – כך שלא תתאפשר חריגה שתקריס את התוכנית.

הוסיפו למערכת את הבדיקות הבאות:

- כל הקצאת זיכרון מצליחה.
- תעודת הזהות של אזרח מורכבת מ-9 ספרות.
- שנת הלידה של אזרח תקינה כך שגילו מעל 18 (בהתאם למועד של סבב הבחירות הנוכחי). יש לבצע השוואה לפי שנת הלידה לשנה של סבב הבחירות, אין צורך להתייחס ליום וחודש.
- בדיקות תקינות נתונות של לוגיקת המערכת (לדוגמא, כל אזרח מצביע פעם אחת בלבד).
- יש לוודא שקלט של מספרים לאורך התוכנית (לדוגמא, תאריך סבב הבחירות) הוא תקין. עבור התאריך ניתן לקבל כל שנה שהיא מספר חיובי, וניתן להניח שבחודש פברואר תמיד יש 28 יום.
- בכל בנאי (בו יש מגבלה על ערכי הארגומנטים) יש לוודא שניתן ערך תקין לכל ארגומנט.
- בחישוב תוצאות הבחירות נזרק חריגה בכל מקרה בו לא ניתן לחשב את התוצאות.
- בדיקות נוספות ושימוש בחריגות בהתאם לעיצוב המחלקות שלכם וללוגיקה הנתונה.

STL

בחלק זה נעדכן את הקוד כך שיעשה שימוש במחרוזות ובמבני הנתונים והאלגוריתמים הסטנדרטיים. יש להמיר בקוד כל שימוש במחרוזות (`char*`) ומבני נתונים (פרט לאחד שמימשתם בסעיף של תבניות) לטיפוסים המתאימים מתוך STL, היכן שאפשר. כמו-כן יש לעדכן כל אלגוריתם כללי (פרט לאחד שמימשתם בסעיף של תבניות) לאלגוריתם המתאים מתוך STL, היכן שאפשר. שימו לב לכל קוד שהופך למיותר בעקבות שינויים אלו ועליכם להוריד.

הנחיות

עליכם לבצע תיקונים בעיצוב ומימוש הקוד כמו בהנחיות התרגיל הקודם, בהתאם למשוב של תרגיל 2. קריטריונים אלו יבדקו שוב בתרגיל 3.

שימו לב לעיצוב נכון של המחלקות. כרגיל, עליכם לעצב את המחלקות באופן מלא וכקופסה שחורה – לכל מחלקה תחום אחריות מוגדר משלה, ממשק פומבי ברור, ועיצוב המחלקות צריך לאפשר שימוש בהן באופן מלא גם במערכות אחרות, ללא תלות חזקה בתפריט או במערכת הספציפית (אלא רק לפי הגדרת הישות שניתנה).

שימו לב למועד ההגשה של התרגיל (אשר מכיל כבר דחיה של שבוע) ותחילת תקופת המבחנים. עליכם לנהל את הזמן נכון כדי להגיש את התרגיל לפני מועד ההגשה. לא יתקבלו דחיות להגשה!

עבור פתרון התרגיל ניתן להשתמש בכל הנושאים שנלמדו בכיתה. יש לבדוק שגיאות של הקצאת זיכרון ולהימנע משימוש בערכי החזר עבור שגיאות, כפי שפורט בסעיף של חריגות. עבור מימוש מבנה הנתונים יש להשתמש בחריגות ולבדוק שגיאות רק היכן שנדרש כפי שנלמד (לדוגמא, אופרטור `[]` לא בודק את תקינות האינדקס אך פעולת `at` כן בודקת).

מעבר לכך, הנחיות ההגשה וההנחיות הכלליות זהות לתרגילים הקודמים.

בהצלחה!