


```

        case add:r1.print(); cout << " + "; r2.print(); cout << " = "; (r1 +
r2).print(); cout << endl;

        case sub:r1.print(); cout << " - "; r2.print(); cout << " = "; (r1 -
r2).print(); cout << endl;

        case mult:r1.print(); cout << " * "; r2.print(); cout << " = "; (r1 *
r2).print(); cout << endl;

        case divi:r1.print(); cout << " / "; r2.print(); cout << " = "; (r1 /
r2).print(); cout << endl;

        case addPre:cout << "++"; cout << "("; r1.print(); cout << ") = ";
(++r1).print(); cout << endl;

        case addPost: cout << "("; r1.print(); cout << "++"; cout << " = ";
(r1++).print(); cout << endl;

        case subPre:cout << "--"; cout << "("; r1.print(); cout << ") = "; (--
r1).print(); cout << endl;

        case subPost:cout << "("; r1.print(); cout << "--"; cout << " = "; (r1--
).print(); cout << endl;

        case big:r1.print(); cout << " > "; r2.print(); cout << " ? "; (r1 > r2) ?
cout << "yes" : cout << "no"; cout << endl;

        case small:r1.print(); cout << " < "; r2.print(); cout << " ? "; (r1 < r2) ?
cout << "yes" : cout << "no"; cout << endl;

        case bigEq:r1.print(); cout << " >="; r2.print(); cout << " ? "; (r1 >= r2)
? cout << "yes" : cout << "no"; cout << endl;

        case smallEq:r1.print(); cout << " <="; r2.print(); cout << " ? "; (r1 <=
r2) ? cout << "yes" : cout << "no"; cout << endl;

        case eequal:r1.print(); cout << " == "; r2.print(); cout << " ? "; (r1 == r2)
? cout << "yes" : cout << "no"; cout << endl;

        case notEqual:r1.print(); cout << " != "; r2.print(); cout << " ? "; (r1 !=
r2) ? cout << "yes" : cout << "no"; cout << endl;

    }
    cout << "enter 1 to cont. 0 to exit." << endl;
    cin >> op;

}

return 0;}

```

דוגמא להרצת התכנית:

enter two rational numbers:

1/2 3/4

1/2 + 3/4 = 5/4

1/2 - 3/4 = -1/4

1/2 * 3/4 = 3/8

1/2 / 3/4 = 2/3

++(1/2) = 3/2

(3/2)++ = 3/2

--(5/2) = 3/2

(3/2)-- = 3/2

1/2 > 3/4 ? no

1/2 < 3/4 ? yes

1/2 >= 3/4 ? no

1/2 <= 3/4 ? yes

1/2 == 3/4 ? no

1/2 != 3/4 ? yes

enter 1 to cont. 0 to exit.

0

[YS1] עם הערות: לשנות את התכנית הראשית, כך שתקבל פלטים ותנסה כמה דברים, ולא רצף אחד כמו פה. (צריך לעשות switch) אפשר לתת תכנית ראשית

שאלה מס' 2:

1. הגדר/י מחלקה עבור ייצוג תאריך.
המחלקה תכלול את התכונות הבאות:

- יום (int)
- חודש (int)
- שנה (int)

ולכל הפחות, את המתודות הבאות:

- constructor – בנוי המקבל כפרמטרים ערכים עבור התכונות: יום, חודש, שנה, עם ערכי ברירת מחדל - 1/1/1920 (כלומר, במידה ולא התקבל ערך עבור אחד או יותר מהשדות, הפרמטר המתאים יקבל את ערך ברירת המחדל התואם לו). על הבנאי לאתחל את התכונות בערכים שהתקבלו, אך לפני עליו לבצע בדיקת תקינות הנתונים באופן הבא:
 - עבור התכונה יום: במידה והתקבל מספר שאינו בין 1-30, תודפס ההודעה: **"Error day"**, והתכונה יום תקבל את הערך 1.
 - עבור התכונה חודש: במידה והתקבל מספר שאינו בין 1-12, תודפס ההודעה: **"Error month"**, והתכונה חודש תקבל את הערך 1.
 - עבור התכונה שנה: במידה והתקבל מספר שאינו בין 1920-2099, תודפס ההודעה: **"Error year"**, והתכונה חודש תקבל את הערך 1920.
 - כמובן – ייתכן ותודפסנה כמה הודעות שגויה עבור תאריך אחד שהתקבל.

▪ copy constructor – בנוי העתקה.

▪ move constructor – בנוי הזה.

▪ מתודה setDate המעדכנת את התאריך. על המתודה לקבל כפרמטרים ערכים עבור כל התכונות. במידה והערכים תואמים את הטווחים התקינים (כאמור במתודה הבונה), על המתודה להציב את הערכים החדשים בתכונות. במידה ואחד הערכים אינו תואם לטווח המתאים לעיל, לא יתבצע כל שינוי **בכל התכונות**, ללא שום פלט.

▪ ++ operator **תחילי**. המתודה משנה את התאריך למחרתו (כלומר - לתאריך של יום למחרת), ומעדכנת את התכונות בהתאם. במידה והשנה לאחר השינוי חורגת מ-2099, יש להכניס לכל התכונות את הערכים 1/1/1920.

▪ ++ operator **סופי**. המתודה משנה את התאריך למחרתו (כלומר - לתאריך של יום למחרת), ומעדכנת את התכונות בהתאם. במידה והשנה לאחר השינוי חורגת מ-2099, יש להכניס לכל התכונות את הערכים 1/1/1920.

▪ += operator. המתודה משנה את התאריך, לתאריך שיחול כעבור מספר הימים שמימין לאופרטור, ומעדכנת את התכונות בהתאם. במידה והשנה לאחר השינוי חורגת מ-2099, יש להכניס לכל התכונות את הערכים 1/1/1920.

▪ > operator. המתודה בודקת קדימות בין שני תאריכים. במידה והתאריך שמשמאל לאופרטור מאוחר יותר מהתאריך שממין לאופרטור, המתודה תחזיר אמת, אחרת המתודה תחזיר שקר.

▪ < operator. המתודה בודקת קדימות בין שני תאריכים. במידה והתאריך שמשמאל לאופרטור מוקדם יותר מהתאריך שממין לאופרטור, המתודה תחזיר אמת, אחרת המתודה תחזיר שקר.

▪ operator==.

▪ מתודה אשר מדפיסה את התאריך בפורמט: dd/mm/yyyy (כאשר היום והחודש בני ספרה אחת או שתיים, והשנה בת ארבע ספרות).

עבור ערכים בוליאניים יש להדפיס true או false.

הערה: יש להניח כי כל החודשים הם בני 30 ימים.

כתוב/י תכנית ראשית אשר תקלוט מהמשתמש תאריך ("Enter a date") (שים לב לפורמט הקלט). על התכנית לאתחל אובייקט שהוא מופע של המחלקה Date בתאריך שהתקבל.

התכנית תדפיס את התאריך (באמצעות המתודה שהגדרת).

לאחר מכן, בלולאה – התכנית תדפיס: **"What do you want to do"**. תקלוט מהמשתמש קוד פעולה, תבצע את הפעולה המתאימה לקוד בהתאם לרשום להלן.

[YS2] עם הערות: להוסיף בנוי הזה. נבקש הדפסות מכל הבנאים.

שימי לב שזה מחייב אם יקבלו ויחזירו מהאופרטור עם & ובלי וכן הלאה.

[YS3] עם הערות: אני רוצה שיקלטו ויאתחלו, ולא שיבנו עצם ואחכ יעשו לו set (נוהג פסול שקיים) ולכן ההדפסה מהבנאי צריכה להיות נכונה – עם הערכים וכו'. הייתי מקפידה על const בכותרת של get.

שים לב! יש לבצע את פעולת האופרטור ואת ההדפסה של התאריך בפעולה אחת (כגון:
`(dateForExample++).print();`
 התכנית תסתיים כאשר ייקלט הקוד 1- (מינוס אחד).
 קודי הפעולה:

- 1 - עדכון תאריך. התכנית תדפיס `Enter a date`, ותקלוט תאריך חדש מהמשתמש. התכנית תפעיל את המתודה `setDate`. לאחר מכן התכנית תדפיס את התאריך המעודכן.
- 2 - התכנית תדפיס את פעולת האופרטור ++ תחילי.
- 3 - התכנית תדפיס את פעולת האופרטור ++ סופי.
- 4 - התכנית תדפיס `Enter # days`, ותקלוט מספר מהמשתמש. התכנית תדפיס את פעולת האופרטור += של התאריך עם המספר שנקלט.
- 5 - התכנית תדפיס `Enter a date`, ותקלוט תאריך מהמשתמש עבור עצם חדש. התכנית תדפיס את פעולת האופרטור > (העצם הישן משמאל לאופרטור).
- 6 - התכנית תדפיס `Enter a date`, ותקלוט תאריך חדש מהמשתמש. התכנית תדפיס את פעולת האופרטור < (התאריך הישן משמאל לאופרטור).
- 7 - התכנית תדפיס `Enter a date`, ותקלוט תאריך חדש מהמשתמש. התכנית תדפיס את פעולת האופרטור == (התאריך הישן משמאל לאופרטור).
- 1 - התכנית תפסיק את ביצוע הלולאה, ותסתיים.
 דוגמא להרצת התכנית: (אין להדפיס את ההערות אלא רק את הערכים)

```

Enter a date
-5/1/2012//קלט מהמשתמש/
Error day
1/1/2012//הדפסת הערך שהוצב בתוך התאריך/
What do you want to do
1
Enter a date
5/7/2010//קלט מהמשתמש/
5/7/2010//הדפסת התאריך אחרי השינוי/
What do you want to do
2
6/7/2010//הדפסת אופרטור ++ תחילי על התאריך/
What do you want to do
3
6/7/2010//הדפסת אופרטור ++ סופי על התאריך/
What do you want to do
4
Enter # days
7
14/7/2010//הדפסת הוספת 7 ימים לתאריך/
What do you want to do
5
Enter a date
14/7/2010
>: false
What do you want to do
7
Enter a date
14/7/2010
==: true
What do you want to do
-1
  
```

שאלה מס' 3:

השלם/י את הגדרת המחלקה `MyString` (שהוצגה בהרצאה) (ניתן להיעזר בדוגמאות המופיעות ה-github), כך שתאפשר להשתמש במחרוזות באופן טבעי. עליך להוסיף למחלקה לפחות את המתודות הבאות:

- אופרטורים: `<`, `>`, `=`, `<=`, `>=`, `!=` (ע"פ שוויון לקסיקוגרפי).
- אופרטור `[]` עבור הצבה ואיחזור. במקרה של חריגה יש להדפיס הודעת **ERROR**.
- מתודה בשם `insert` המקבלת מחרוזת `str` ומיקום `index`. המתודה אינה משנה את המחרוזת המקורית ומחזירה מחרוזת חדשה המהווה הכנסה של `str` למחרוזת המקורית החל ממיקום `index`. במקרה בו `index` אינו תקין כך שלא ניתן לבצע את ההכנסה הרצויה, תודפס ההודעה: **ERROR** ותוחזר מחרוזת ריקה. הצהרת המתודה הינה:

```
MyString insert( int index, const char* str )
```

כתוב/י תכנית ראשית אשר תקלוט שתי מחרוזות, `a`, `b` כל אחת בשורה נפרדת, וכן מספר `n`, על התוכנית לבצע את הפעולות הבאות:

- להדפיס הודעה המייצגת את היחס בין המחרוזות (לקסיקוגרפית): `a>b` או `a=b` או `a<b`.
 - ואת המחרוזת המתקבלת כאשר מפעילים את המתודה `insert`, כך שהמחרוזת `a`, משובצת במחרוזת `b`, החל מהמיקום `n`. במידה והקלט אינו תקין (כך שלא ניתן להפעיל את המתודה `insert`), תודפס ההודעה: **ERROR** ותוחזר מחרוזת ריקה.
 - בנוסף על התכנית לקלוט תו כלשהו וערך `index` ולשנות את התו במיקום ה-`index` במחרוזת החדשה להיות התו שנקלט (יש להשתמש באופרטור `[]`) ולהדפיס את המחרוזת המעודכנת.
- דוגמאות להרצת התכנית:

World Hello 5 a>b HelloWorld ! 5 Hello!orld	Hello Hello 2 a=b HeHellollo h 2 Hehellollo	Hello World 2 a<b WoHellorld ? 10 ERROR	Hello World 0 a<b HelloWorld - 6 HelloW-rlid	Hello World 8 a<b ERROR J 3 ERROR
--	--	--	---	--

בהצלחה רבה!

[YS4] עם הערות: קוראים לה `mystring` וזה בכוונה – כדי שיבחינו בינה לבין `string` של STL ושל `string.h`

[YS5] עם הערות: בואי נשנה את זה קצת. אפילו לשנות את הסדר בין הפעולות.