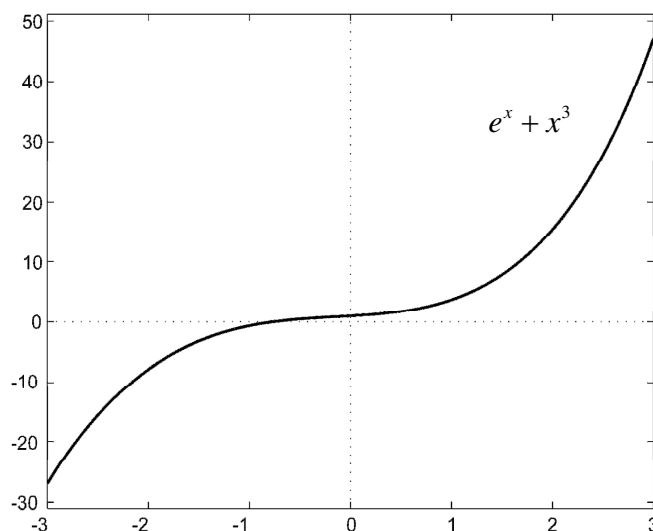




שאלה 2 (20 נקודות)

בשאלה זו נכתוב תוכנית המוצאת את נקודת האפס של הפונקציה $f(x) = e^x + x^3$ (ישנה נקודה יחידה כזו).
להלן גרף המתאר פונקציה זו באופן סכימטי:



סעיף א (10 נקודות)

בסעיף זה נכתוב פונקציה המחשבת את הפונקציה e^x . לשם כך, ידוע שלכל x מתקיים:

$$e^x = 1 + \frac{x}{1} + \frac{x^2}{1 \cdot 2} + \frac{x^3}{1 \cdot 2 \cdot 3} + \frac{x^4}{1 \cdot 2 \cdot 3 \cdot 4} + \dots$$

כאשר מובטח שהאיברים בזנב הטור הולכים וקטנים לאפס. השלימו את הפונקציה בעמוד הבא, המקבלת מספר x ומחשבת קירוב ל- e^x ; על הפונקציה לעצור כאשר התיקון שמתווסף לקירוב הנוכחי קטן בערכו המוחלט מ- ϵ (מוגדר כ-`#define`).

שימו לב: ניתן להעזר בפונקציה `fabs()` שמחשבת ערך מוחלט של מספר ממשי, וחתימתה היא

```
double fabs(double x);
```

אין להעזר בכל פונקציה אחרת בקוד שלכם (ובפרט לא בפונקציה `pow()`).



```
#define EPSILON 1e-10

double myexp(double x)
{
    double result = 1;
    double n = 1;
    double delta = 1;

    do {
        delta *= x/n;
        results += delta;
        n++;
    } while (fabs(delta) > EPSILON);

    return result;
}
```

לנוחותכם, הנוסחה מובאת כאן בשנית:

$$e^x = 1 + \frac{x}{1} + \frac{x^2}{1 \cdot 2} + \frac{x^3}{1 \cdot 2 \cdot 3} + \frac{x^4}{1 \cdot 2 \cdot 3 \cdot 4} + \dots$$

סעיף ב (10 נקודות)

נתונה הפונקציה הבאה המחשבת את $f(x) = e^x + x^3$ בנקודה x, ועושה שימוש בפונקציה מהסעיף הקודם:

```
double f(double x) {
    return myexp(x) + x*x*x;
}
```



כתבו פונקציה שמוצאת את נקודת האפס של $f(x)$ (הנקודה בה ערך הפונקציה הוא 0). פונקציה זו תקבל שתי נקודות קצה x_{\min} ו- x_{\max} , כאשר $x_{\min} < x_{\max}$, ומתקיים ש- $f(x_{\min}) < 0$ ו- $f(x_{\max}) > 0$; על הפונקציה להחזיר את נקודת האפס של f הנמצאת בין שתי נקודות הקצה (מובטח שיש כזו). הפונקציה נדרשת לעבוד בדיוק DELTA (קבוע נוסף המוגדר כ-#define), כשהכוונה היא שהנקודה x המוחזרת צריכה להיות לכל היותר במרחק DELTA מנקודת האפס האמיתית x_0 . (שימו לב שהתנאי על הנקודה המוחזרת x איננו תלוי בערך הפונקציה עצמו בנקודה זו).

בסעיף זה מותר להשתמש בפונקציה $f()$ הנתונה גם אם לא פתרתם את הסעיף הקודם.

```
#define DELTA 1e-8

double findzero(double xmin, double xmax) {
    double xmid, val;

    while (xmax-xmin > DELTA)
    {
        xmid = (xmax+xmin)/2;
        val = f(xmid);
        if (val==0)
            return xmid;
        else if (val>0)
            xmax = xmid;
        else
            xmin = xmid;
    }

    return (xmax+xmin)/2;
}
```