



שאלה 2

בבית ההשקעות "כפית הכסף" הצליחו לחזות במדויק את מחיר מניית "דג הזהב" ב-ח הימים הקרובים. הניחו כי המחירים העתידיים שמורים במערך `quotes` כאשר הכניסה הראשונה במערך מכילה את המחיר הנוכחי (היום הראשון). נרצה לכתוב פונקציה המחשבת את הרווח המקסימלי שניתן להשיג ע"י קניית המניה ומכירתה. שימו לב שיש לחשב את הרווח באחוזים. נוסחה כללית

$$revenue = \frac{sell_price - buy_price}{buy_price} \text{ ע"י:}$$

דוגמה: עבור סדרת המחירים הבאה

100	110	80	90	120
-----	-----	----	----	-----

- קניית המניה ביום הראשון (מחיר 100) ומכירתה ביום האחרון (120) תניב רווח של

$$\frac{120 - 100}{100} = 0.2 = 20\%$$
- קניית המניה ביום השלישי (מחיר 80) ומכירתה ביום האחרון (120) תניב רווח של

$$\frac{120 - 80}{80} = 0.5 = 50\%$$
- קניית המניה ביום הראשון (מחיר 100) ומכירתה ביום השלישי (80) תניב הפסד של

$$\frac{80 - 100}{100} = -0.2 = -20\%$$
- שילוב פעולת קנייה ביום הראשון (מחיר 100) ומכירה ביום השני (110) ואז קנייה מחדש ביום השלישי (מחיר 80) ומכירתה ביום האחרון (120) תניב רווח כולל של

$$\left(1 + \frac{110 - 100}{100}\right) \cdot \left(1 + \frac{120 - 80}{80}\right) - 1 = 1.1 \cdot 1.5 - 1 = 0.65 = 65\%$$
 במילים אחרות אילו היינו משקיעים 100 דולר במניות "דג הזהב" ומוכרים אותן אחרי יום, ולאחר מכן משקיעים את תמורתן במניות ביום השלישי ומוכרים אותן ביום החמישי הסכום אותו היינו מקבלים הוא 165 דולר, ולכן הרווח הוא 65%.

ניתן להניח את ההנחות הבאות:

- המחיר אינו משתנה במהלך היום
- תמיד ישנן מניות למכירה וקונים במחירים הנתונים
- קניה ומכירה אינה עולה כסף (עמלות, מיסים וכיו"ב)
- לא ניתן לקנות מניות לפני היום הראשון, לא ניתן למכור מניות אחרי היום האחרון
- ניתן לקנות חלקי מניות ללא הגבלה
- לחברת "כפית הכסף" יש סכום התחלתי להשקעה X, בכל פעולת קנייה נעשה שימוש בכל הכסף הזמין ובכל פעולת מכירה נמכרות כל המניות שניקנו



סעיף א – 2 נקודות

בהינתן מערך כנ"ל ואורכו, כתבו פונקציה יעילה ככל האפשר המחזירה את הרווח במידה וקונים את המניה ביום הראשון ומוכרים אותה ביום האחרון. למשל, עבור המערך לעיל הפונקציה תחזיר 0.2 (20%).

```
double buy_first_sell_last(double quotes[], int n)
{
    return (quotes[n-1]-quotes[0])/quotes[0];
}
```



סעיף ב – 9 נקודות

בהינתן מערך כנ"ל ואורכו, כתבו פונקציה יעילה ככל האפשר המחזירה את הרווח המירבי אותו ניתן להשיג ללא הגבלה של פעולות קניה ומכירה (כלומר מותר לקנות ולמכור מספר פעמים). למשל, עבור המערך לעיל הפונקציה תחזיר 0.65 (65%).
רמז: בהנחה שאתם מחזיקים במנייה כלשהי וידוע לכם שמכירה עומד לרדת, ובהנחה שעלות פעולות מכירה וקנייה היא 0 (כפי שנתון), מהו הדבר הנכון לעשות? מהו הדבר הנכון לעשות במידה וידוע לכם שמכיר המנייה עומד לעלות?

```
double buy_sell_nolimit(double quotes[], int n)
{
    double sum = 1;
    double last buy = 0;
    int i = 0;

    for(i = 0; i < n-1; i++) {
        if((quotes[i+1] < quotes[i]) && (last buy > 0)) {
            sum *= quotes[i]/last buy;
            last buy = 0;
        }
        else if((quotes[i+1] > quotes[i]) && (last buy == 0)) {
            last but = quotes[i];
        }
    }

    if(last buy)
        sum *= quotes[n-1]/last buy;

    return (sum - 1);
}
```



סעיף ג – 14 נקודות

בהינתן מערך כנ"ל ואורכו, כתבו פונקציה יעילה ככל האפשר המחזירה את הרווח המירבי אותה ניתן להשיג ע"י פעולת קניה אחת בלבד ופעולת מכירת אחת בלבד. למשל, עבור המערך לעיל הפונקציה תחזיר 0.5 (50%).
רמז: מרגע שקניתם מנייה, מתי הכי כדאי למכור אותה?

```
double buy_sell_once(double quotes[], int n)
{
    double max_revenue = 0;
    double *buy, *sell;

    for(buy=sell=quotes; sell<(quotes+n); sell++) {
        {
            if(*buy > *sell)
                buy = sell;

            if((( *sell-*buy)/ *buy-1) > max_revenue)
                max_revenue = (( *sell-*buy)/ *buy-1);
        }
        return max_revenue;
    }
}
```