

קורס תכנות - תרגיל מס' 11

מועד הגשה: התרגול בשבוע הבא

הנחיות כלליות: קראו בעיון את השאלות והקפידו שהתכניות שלכם יפעלו בהתאם לנדרש.. הקפידו על כללי ההגשה המפורסמים באתר.

שימו לב: את התרגיל יש לפתור לבד!

1. חשבון בנק

```
typedef struct accountType_t{
    char type;
    double maxWithdrawPerDay;
    double minBalance;
}AccountType;

typedef struct bankAccount_t{
    char owner[50];
    char identifier[10];
    double balance;
    double withdrawPerDay;
    AccountType* typePtr;
} BankAccount;
```

נתון מבנה המייצג חשבון בנק. המבנה מכיל שדות המציינים את שם בעל החשבון, מספר חשבון, יתרה וסוג חשבון. נתון שהשם של בעל החשבון הוא באורך 49 תווים לכל היותר, מספר החשבון הוא בן תשע ספרות (יכול להתחיל ב-0) ויתרה היא מספר ממשי (תתכן יתרה שלילית).

ישנם שני סוגי חשבון אפשריים:

חשבון רגיל (S) וחשבון עסקי (B). חשבון רגיל יכול למשוך עד 500 ₪ ליום וחשבון עסקי יכול למשוך עד 1000 ₪ ליום. כמו כן, חשבון רגיל יכול להיות במינוס של עד 500 ₪ וחשבון עסקי יכול להיות במינוס של עד 5000 ₪.

כתבו 4 פונקציות:

- `int isPositiveBalance(BankAccount *bnkac)`
פונקציה זו מחזירה `true` אם היתרה בחשבון של לקוח נתון, היא חיובית (כולל 0).
אחרת תחזיר `false`.
- `int applyTransaction(BankAccount * bnkac, double amount)`
פונקציה זו מוסיפה ליתרה את `amount` (או מחסירה, במידה ו-`amount` שלילי).
הפונקציה תבצע את הפעולה רק אם היא חוקית על פי סוג החשבון. (כלומר משיכה לא חורגת מ-`maxWithdrawPerDay` ויתרה לא חורגת מ-`minBalance`). הפונקציה תחזיר `true` אם הפעולה בוצעה ו-`false` אחרת.
- `AccountType * getAccountType(AccountType types[], int numTypes, char type)`

פונקציה זו מוסיפה מחזירה מצביע ל-accountType המתאים, ו-NULL אם ה-type לא נמצא. אם קיימים כמה accountType הפונקציה תחזיר מצביע לראשון מביניהם..

- void printAccount (BankAccount *bnkac)

פונקציה זו מדפיסה את פרטי החשבון (בעל החשבון, מספר חשבון, יתרה וסוג חשבון).

כתבו תכנית בהתאם להנחיות הבאות:

- בנו (hard-coded) מערך של סוגי החשבונות האפשריים ומערך נוסף שמכיל 5 חשבונות (שני חשבונות עסקיים ושלושה חשבונות פרטיים), היתרה בכולם 0.
- אפשרו למשתמש לבצע פעולות על החשבונות בלולאה (בכל איטרציה, המשתמש יקיש את האינדקס של החשבון במערך ואת הסכום שאותו הוא רוצה להפקיד/למשוך).
- הניחו שכל הפעולות מתבצעות באותו יום.
- התוכנית תדפיס הודעה מתאימה, אם לא ניתן לבצע את הפעולה
- סיימו את התכנית כאשר האינדקס שהוקש חורג מגבולות המערך
- בסיום התכנית הדפיסו את פרטי החשבון שבהם היתרה היא שלילית

דוגמת הרצה : (מחשב, משתמש)

```
account: 1111 owener: David balance: 0 type: S
account: 2222 owener: Rotem balance: 0 type: S
account: 3333 owener: Tomer balance: 0 type: S
account: 4444 owener: Ronit balance: 0 type: B
account: 5555 owener: Guy balance: 0 type: B
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
0 -300
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
0 -300
```

```
The transaction is invalid!
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
1 -600
```

```
The transaction is invalid!
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
1 -200
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
1 -200
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
1 -200
```

```
The transaction is invalid!
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
3 -800
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
4 1000
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
1 2000
```

```
Enter a transaction (in fmt: accountIdx amount)
```

```
6 0
```

The negative balances are:

identifier: 1111 owener: David balance: -300 type: S

identifier: 4444 owener: Ronit balance: -800 type: B