

סדנא ב- C++ – 150018

תרגיל בית מספר 11

קבצים בינאריים ואריתמטיקה ביטים

שים/י לב:

- א. הקפד/י על קריאות התכנית ועל עימוד (Indentation).
- ב. הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
- ג. בכל אחת מהשאלות יש להגדיר פונקציות במידת הצורך עבור קריאות התכנית.
- ד. יש להגיש את התרגיל על פי ההנחיות להגשת תרגילים (המופיע באתר הקורס) וביניהם:
השתמש/י בשמות משמעותיים עבור המשתנים.
יש לתעד את התכנית גם עבור פונקציות אותם הנך מגדיר/ה וכן על תנאים ולולאות וקטעי קוד מורכבים, ובנוסף, **דוגמת הרצה לכל תכנית בסוף הקובץ!**
הגשה בזוגות.

הערה חשובה: לכל תרגיל בית מוגדר שבוע אחד בלבד להגשה (ולא יאוחר ממועד הבחינה).
תיבות ההגשה הפתוחות לא מהוות היתר להגשה באיחור.

שאלה מס' 1:

בתרגיל זה עלייך לבצע גישה אקראית לקובץ בינארי (Random Access Binary Files) שעל חלק מהנתונים בו תתבצע אריתמטיקה של ביטים ונשתמש כמובן בזריקת חריגות עבור שגיאות בקלט.

נתונים על חריגות

הפעם במקום לזרוק מחרוזת עבור שגיאה, תשתמשו במחלקה הקיימת exception ותשתמשו ב-message כדי להדפיס את סוג השגיאה.

נתונים על הקובץ:

התכנית מאתחלת קובץ בינארי עבור 100 רשומות זהות בגודלן, ולכל אחת מהן מפתח ראשי ייחודי הממוספר מ-1 עד 100, בהתאם למיקומה בקובץ (המספור מתחיל מ-1, כלומר – לרשומה השנייה יהיה את המפתח 2 ולרשומה האחרונה את המפתח 100).
בתחילה הרשומות ריקות, ובמהלך התכנית מתבצעת גישה ישירה לרשומה, לפי דרישת המערכת (מאחר והרשומות זהות בגודלן, ניתן לדעת את מיקומה של כל רשומה בקובץ, בהינתן המפתח הראשי שלה). לאור זאת – ניתן להניח שבמידה והמפתח הראשי של רשומה הוא 0, הרשומה ריקה. ולעומת זאת אם המפתח הראשי שונה מ-0, הרשומה קיימת במערכת. (במילים אחרות – נגיד שמשפחה קיימת במערכת במידה שהיא קיימת בקובץ – כלומר: המספר המזהה במקום המתאים בקובץ שווה למספר המשפחה ולא 0). **נתונים על החוגים:**

בשכונה ישנה אפשרות להירשם לשמונת החוגים הבאים: שחיה, התעמלות קרקע, מחול, אומנות, הגנה עצמית, מוסיקה, דרמה וכדור סל.

כדי לשמור נתונים על החוגים, נחזיק בית (byte) אחד ונשתמש באריתמטיקה של ביטים.

עבור כל חוג ישנו ביט מסויים המייצג את החוג, ולכן אם הביט הזה "דלוק" (1) - זה מייצג שהמשפחה משתתפת בחוג, ואם הביט "מכובה" (0) - זה מסמן שהמשפחה לא משתתפת בחוג הזה.

כלומר :

החוג	שחיה	התעמלות קרקע	מחול	אומנות	הגנה עצמית	מוסיקה	דרמה	כדור סל
מספר ביט	0	1	2	3	4	5	6	7

לדוגמה:

למספר 1 התצוגה בבינארי הוא 00000001 (ערך 1) המסמן השתתפות בחוג שחיה
למספר 2 התצוגה בבינארי הוא 00000010 המסמן השתתפות בחוג התעמלות קרקע
למספר 68 התצוגה בבינארי הוא 01000100 המסמן השתתפות גם בחוג מחול וגם בחוג דרמה

לצורך ביצוע החישובים בקלות, נגדיר את ה-enum הבא:

```
enum ACTIVITY {
    NONE,
    SWIMMING,           // שחיה
    GYMNASTICS,         // התעמלות קרקע
    DANCE = 4,          // מחול
    ART = 8,             // אומנות
    SELF_DEFENSE = 16,  // הגנה עצמית
    MUSIC = 32,          // מוסיקה
    DRAMA = 64,          // דרמה
    BASKETBALL = 128;   // כדור סל
};
```

כתוב/י תכנית אשר תנהל את מערכת הרישום של ילדים בחוגים במקום מגוריהם.

1. הגדר/י מחלקה Family המתאימה לשמירת נתונים של כל משפחה. הנתונים הם:

- מס' משפחה (מס' בטווח 1-100. מס' המהווה מפתח ייחודי)
- שם משפחה (עד 20 תווים)
- מס' נפשות במשפחה
- מס' טלפון

ה. תכונה התשמור רשימה של כל החוגים שהמשפחה משתתפת בה.
נשתמש באריתמטיקה של ביטים כדי לשמור נתונים על כמה חוגים שונים בתוך **משתנה אחד מסוג short**. פרטים בהמשך.

2. כתוב/י פונקציה בשם setFile המקבלת כפרמטר הפניה לקובץ בינארי ומאתחלת אותו לשמירת נתונים של 100 משפחות. לשם כך, יש להציב בקובץ 100 פעם, רשומה ריקה של משפחה, ובה הנתונים הבאים:

- מספר משפחה – 0
- שם משפחה – מחרוזת ריקה באורך 20
- מס' נפשות במשפחה- 0
- מס' טלפון- 0

- ה. חוגים - ערך של NONE (כי אין רשום לאף חוג).
3. כתוב/י את הפונקציות הבאות, לצורך מערכת הרישום:
- א. פונקציה להוספת משפחה בשם add. הפונקציה מקבלת כפרמטר הפניה לקובץ בינארי. הפונקציה קולטת את הנתונים עבור משפחה מתוך הקלט הסטנדרטי (מקלדת) ללא רישום לחוגים. כלומר, יש לאתחל ב"NONE", את התכונה של הרישום לחוגים עבור משפחה חדשה. במידה שמספר המשפחה שהתקבל עדיין לא מופיע בקובץ הנתונים - יתווספו פרטי המשפחה לקובץ. במידה והמשפחה כבר מופיעה במערכת הפונקציה תזרוק את ההודעה:
- "ERROR: Family is already in the file"
- ועבור מספר משפחה שאינו בטווח התקין (1-100):
- "ERROR: Invalid family number"
- ב. פונקציה למחיקת משפחה בשם del. הפונקציה מקבלת כפרמטר הפניה לקובץ בינארי ופרמטר שני המהווה מספר משפחה. הפונקציה מוחקת מהקובץ את המשפחה שמספרה התקבל כפרמטר. (שים לב – משמעות המחיקה היא הצבת משפחה "ריקה" במקום, כך שהמפתח הראשי שלה הוא 0. אין צורך לדאוג לשדות האחרים). במידה והתקבל מספר משפחה שאינו בטווח התקין (1-100) הפונקציה תזרוק את הודעת החריגה:
- "ERROR: Invalid family number".
- . במידה והמשפחה לא מופיעה במערכת (בקובץ) הפונקציה תזרוק את ההודעה:
- "ERROR: Family is not in the file"
- ג. פונקציית עזר בשם count. הפונקציה מקבלת כפרמטר הפניה לקובץ בינארי ופרמטר שני המהווה חוג מתוך רשימה של חוגים (פרמטר של הפונקציה הוא מסוג int ופרמטר הנשלח לפונקציה יהיה ערך מתוך ACTIVITIES. על הפונקציה לעבור על כל המשפחות בקובץ, למנות ולהחזיר את מספר המשפחות הרשומות לחוג שהתקבל כפרמטר. (נשתמש בפונקציה הזו בסעיף הבא).
- יש לכתוב את המימוש של הפונקציה הזו אך ורק על ידי אריתמטיקה של ביטים.
- רמז: אפשר לדעת האם משפחה רשומה לחוג מסוים על ידי הפעלת האופרטור & עם שם החוג – אם התוצאה אינה 0 אז המשפחה רשומה לחוג.
- כלומר אם מדובר על משפחה המשתתפת בחוגים של מחול ודרמה אז שמור בתכונה הערך 68 שתצוגה בבינארי הוא 01000100
- אם בודקים האם המשפחה משתתפת בחוג של SWIMMING שערכו הוא 1 (תצוגה בבינארי 00000001) ונעשה & ביניהם אז נקבל ערך של 0 – המייצג שהמשפחה לא משתתפת בחוג שחיה
- אבל עבור אותה משפחה, אם בודקים האם משתתפים בחוג של DANCE שערכו הוא 4 (תצוגה בבינארי 00000100) ונעשה & ביניהם נקבל 00000100 שאינה 0 המייצג שהמשפחה כן משתתפת בחוג מחול.

ד. פונקציה לעדכון רישום בשם update. הפונקציה מקבלת כפרמטר הפניה לקובץ בינארי, פרמטר שני המהווה מספר משפחה ופרמטר שלישי תור (STL) המכיל את משפחות הממתינים לפינוי מקום בחוג. במידה והמשפחה אינה מופיעה במערכת, הפונקציה תזרוק את החרגה: "ERROR: Family is not in the file"

עבור מספר משפחה שאינו בטווח התקין (1-100) תזרוק החרגה: "ERROR: Invalid family number"

במידה והכל תקין (מספר המשפחה מופיע בקובץ הנתונים), הפונקציה תקלוט ערכים עבור ה-8 החוגים מהקלט הסטנדרטי. הפונקציה תשאל עבור כל חוג האם המשפחה מעוניינת בחוג ותקלוט מהמשתמש Y או y אם המשפחה מעוניינת בחוג, ו-N או n אם המשפחה לא מעוניינת בו.

עבור כל תשובה אחרת הפונקציה תזרוק את ההודעה: "ERROR: Invalid response"

פלט הפונקציה יהיה:

Do you want to register for swimming?
Do you want to register for gymnastics?
Do you want to register for dance?
Do you want to register for art?
Do you want to register for self defense?
Do you want to register for music?
Do you want to register for drama?
Do you want to register basketball?

מספר המשתתפים בכל חוג מוגבל ל-10 ילדים. ולכן, עבור כל חוג שהמשפחה רוצה להשתתף בהו, על הפונקציה לבדוק שאכן יש מקום פנוי בחוג. (יש להשתמש בפונקציה שהגדרת בסעיף הקודם הבדוקת כמה נרשמים יש לחוג מסוים)

במידה וקיים מקום, הפונקציה תעדכן את רשימת החוגים הנבחרים (את התכונה חוגים) של המשפחה שמספרה התקבל כפרמטר. שימו לב: ניתן לבצע עדכון חלקי – כלומר אם רוצים 3 חוגים ורק ב 2 מתוכם יש מקום פנוי, אז יש לרשום את המשפחה רק לשני החוגים שבהם יש מקום פנוי.

את הנתונים החדשים עבור המשפחה אחרי הוספת חוגים, יש להכניס לקובץ הבינארי.

במידה והמשפחה רצתה להירשם לחוג בו לא קיים מקום, לאחר עדכון הקובץ, יש לעדכן את רשימת ההמתנה עבור החוגים בהם אין כרגע מקום. כך: הפונקציה תוסיף את המשפחה לרשימת המתנה שהתקבלה כפרמטר מסוג תור של STL. (לשיקולכם כיצד לשמור: ניתן לשמור בתור משפחה כאשר תכונת החוגים מייצגת את החוגים שאליהם המשפחה רוצה להירשם [בדומה לצורה בה הערכים שמורים בקובץ]. לחילופין, ניתן לשמור את אותה

משפחה מס' פעמים עבור כל בקשה לחוג, כאשר ערך החוגים מאותחל בכל פעם לחוג מסויים בלבד)

ה. פונקציה לטיפול ברשימת המתנה waiting. הפונקציה מקבלת כפרמטר הפניה לתור המשפחות הממתינות לחוג, הפונקציה תדפיס (למסך) את פרטי המשפחה: שם משפחה, מס' נפשות, מס' טלפון ומיד אח"כ רשימת כל החוגים שהמשפחה ממתינה להם.

ו. פונקציה להדפסת פרטי משפחה בשם print. הפונקציה מקבלת כפרמטר הפניה לקובץ בינארי, ופרמטר שני המהווה מספר משפחה. במידה ומספר המשפחה קיים בקובץ הנתונים, הפונקציה תדפיס לפלט הסטנדרטי (מסך) את פרטי המשפחה: שם משפחה, מס' נפשות, מס' טלפון ושם של כל החוגים שהיא משתתפת בהם. במידה והמשפחה אינה מופיעה במערכת, הפונקציה תזרוק את ההודעה:
"ERROR: Family is not in the file" ועבור מספר משפחה שאינו בטווח התקין (1-100): "ERROR: Invalid family number".

לפניך תוכנית ראשית הבוחנת את נכונות הפונקציות שכתבת:

בשדה result שומרים את תוצאת הפעולה, לשם כך הגדירו את ה- enum הבא:

```
enum ACTIVITY {
    NONE, // טרם בחר חוג
    SWIMMING, // שחיה
    GYMNASTICS, // התעמלות קרקע
    DANCE = 4, // מחול
    ART = 8, // אומנות
    SELF_DEFENSE = 16, // הגנה עצמית
    MUSIC = 32, // מוסיקה
    DRAMA = 64 // דרמה
    BASKETBALL = 128; // כדור סל
};
```

```
#include "Family.h"
#include <queue>
#include <iostream>
#include <fstream>
#include <exception>

using namespace std;

enum { EXIT, ADD, DEL, COUNT, UPDATE, WAITING, PRINT };

enum ACTIVITY {
    NONE, // טרם בחר חוג
    SWIMMING, // שחייה
    GYMNASTICS, // התעמלות קרקע
    DANCE = 4, // מחול
    ART = 8, // אומנות
    SELF_DEFENSE = 16, // הגנה עצמית
    MUSIC = 32, // מוסיקה
    DRAMA = 64 // דרמה
    BASKETBALL = 128; // כדור סל
};

void handleCount(fstream& file) {
    int snum;
    cout << "enter activity number to count:\n";
    cout << "Choices are:\n1 SWIMMING\n2 GYMNASTICS\n3 DANCE\n4\n5 SELF DEFENSE\n6 MUSIC\n7 DRAMA\n8 BASKETBALL\n";
    cin >> snum;
    cout << "there are ";
    switch (snum) {
        case 1:
            cout << count(file, SWIMMING);
            break;
        case 2:
            cout << count(file, GYMNASTICS);
            break;
        case 3:
            cout << count(file, DANCE);
            break;
        case 4:
            cout << count(file, ART);
            break;
        case 5:
            cout << count(file, SELF_DEFENSE);
            break;
        case 6:
            cout << count(file, MUSIC);
            break;
        case 7:
            cout << count(file, DRAMA);
            break;
        case 8:
            cout << count(file, BASKETBALL);
            break;
    }
```

```
    }
    cout << " families registered\n";
}

int main()
{

    queue<Family> q;
    fstream file;
    file.open("families.txt", ios::binary | ios::out);

    if (!file)
    {
        cout << "ERROR: couldn't open file\n";
        return 0;
    }
    setFile(file);
    file.close();

    file.open("families.txt", ios::binary | ios::in | ios::out);
    int choice;
    int snum;
    int cnum;
    cout << "Choices are:\n0 to exit\n1 to add a family\n2 to
delete a family\n3 to count number of families signed up\n4 to
update the activities of a family \n5 print the waiting list \n6 to
print a family\n";
    cout << "enter 0-6:\n";
    cin >> choice;
    while (choice)
    {
        try {
            switch (choice)
            {
            case ADD://add to the file
                add(file);
                break;
            case DEL://delete from file
                cout << "enter number of family to delete:\n";
                cin >> snum;
                del(file, snum);
                break;
            case COUNT:
                handleCount(file);
                break;
            case UPDATE://update the list of classes of a family
                cout << "enter number of family to update:\n";
                cin >> snum;
                update(file, snum, q);
                break;
            case WAITING://update the list of classes for waiting fam

                waiting(q);
                break;

            case PRINT://print the details of a specific family
                cout << "enter number of family to print:\n";
                cin >> snum;
                print(file, snum);
                break;
            }
        }
    }
}
```

```
        default:
            cout << "ERROR: invalid choice\n";

        }
    } // try
    catch (const char* msg) {
        cout << msg;
    }
    cout << "\nenter 0-6:\n";
    cin >> choice;

}
file.close();
return 0;
}
```


דוגמה :

```
Choices are:
0 to exit
1 to add a family
2 to delete a family
3 to count number of families signed up
4 to update the activities of a family
5 print the waiting list
6 to print a family

enter 0-6:
1

enter the family's: id, name, amount of people and a phone number:
5 aaron 5 0505551235

enter 0-6:
1

enter the family's: id, name, amount of people and a phone number:
123 brian 7 0505551123

ERROR: Invalid family number

enter 0-6:
1

enter the family's: id, name, amount of people and a phone number:
5 aaron 5 0505551235

ERROR: Family is already in the file

enter 0-6:
1

enter the family's: id, name, amount of people and a phone number:
6 carly 8 0505551236

enter 0-6:
1

enter the family's: id, name, amount of people and a phone number:
7 daniella 9 0505551237

enter 0-6:
2

enter number of family to delete:
6
```

```
enter 0-6:
4
enter number of family to update:
6
ERROR: Family is not in the file

enter 0-6:
4

enter number of family to update:
5

Do you want to register for swimming?
Y
Do you want to register for gymnastics?
Y
Do you want to register for dance?
Y
Do you want to register for art?
Y
Do you want to register for self defense?
N
Do you want to register for music?
N
Do you want to register for drama?
N
Do you want to register basketball?
N

enter 0-6:
4
enter number of family to update:
7
Do you want to register for swimming?
Y
Do you want to register for gymnastics?
N
Do you want to register for dance?
Y
Do you want to register for art?
N
Do you want to register for self defense?
Y
Do you want to register for music?
N
Do you want to register for drama?
Y
Do you want to register basketball?
N
```

```
enter 0-6:
3

enter activity number to count:
Choices are:
1 SWIMMING
2 GYMNASTICS
3 DANCE
4 ART
5 SELF DEFENSE
6 MUSIC
7 DRAMA
8 BASKETBALL
1

there are 2 families registered

enter 0-6:
6

enter number of family to print:
5

family name: aaron
number of persons: 5
phone number: 505551235
!2!15
SWIMMING
GYMNASTICS
DANCE
ART

enter 0-6:
6

enter number of family to print:
6
ERROR: Family is not in the file

enter 0-6:
6

enter number of family to print:
7

family name: Daniella
number of persons: 9
phone number: 505551237
SWIMMING
DANCE
SELF DEFENSE
DRAMA
```

```
enter 0-6:  
6  
  
enter number of family to print:  
8  
ERROR: Family is not in the file  
  
enter 0-6:  
5  
  
enter 0-6:  
0
```