

שאלות נחמדות בסגנון מבחן לתרגול . . .

שאלה מספר 1

בתכנית מוגדר ע"י `define N` קבוע `N`. כתבו פונקציה `int check_sum(unsigned arr[N], unsigned sum)` המקבלת כפרמטר מערך `(arr)` של מספרים שלמים ואי שליליים, ומספר עולם ואי שלילי `(sum)` ובדקת האם במערך ישנו קטע של מספרים רצופים שסכומם בדיוק `sum`. הפונקציה מחזירה את האינדקס בו מתחיל קטע כזה אם קיים ו `-1` אחרת.

שאלה מספר 2

כתבו פונקציה `ascend()` המקבלת כפרמטרים מערך של `int` וגודלו, ומחזירה את אורך הקטע העולה הארוך ביותר במערך.

כתבו פונקציה `sorted()` המקבלת כפרמטרים מערך של `int` וגודלו ומחזירה `true` אם המערך ממוין (בסדר עולה או בסדר יורד) ו `false` אחרת. `(bool מטיפוס true, false)`. הערה: אפשר להשתמש בפונקציה מסעיף א גם אם לא פתרתם את סעיף א.

שאלה מספר 3

מערך של `int`, נקרא הררי אם מורכב משני חלקים (לאו דווקא זהים בגודלם), כאשר החלק הראשון הוא סדרה עולה, והחלק השני הוא סדרה יורדת.

כתבו פונקציה בשם `mountain()` המקבלת כפרמטר מערך בגודל `N` של מספרים מטיפוס `int`, ומחזירה את האינדקס בו מתחילה הסדרה היורדת אם המערך הררי ו `0` אחרת. הערות:

✓ סדרה עולה / יורדת היא בת שני איברים לפחות.

✓ איבר יחיד יכול להיחשב גם כחלק של סדרה עולה וגם כחלק של סדרה יורדת (באיור, האיבר 3 הוא גם חלק של סדרה עולה: (1,3) וגם חלק של סדרה יורדת: (3,2)).

שאלה מספר 4

א. כתבו פונקציה `void move_max()` המקבלת מערך של עצמים מטיפוס `int` ואת אורך המערך, ומעבירה את האיבר הקטן ביותר במערך לתחילת המערך - בלי לשנות את הסדר הפנימי של שאר האיברים במערך.

ב. כתבו פונקציה `void sort()` המקבלת מערך של עצמים מטיפוס כלשהו ואת אורך המערך, ומסדרת את אברי המערך מהקטן לגדול. אפשר להשתמש בפונקציה מסעיף א גם אם לא פתרתם את סעיף א.

שאלה מספר 5

קובץ קלם מכיל מקסט עם תווים כלשהם.

מילה: תונוך כרצף של אותיות בלבד (קמנות ו\או גדולות) שמשמאלו (לפניו) יש תמיד TAB (`'\t'`) או רווח, ומימינו (אחריו) יש רווח או סוף שורה (`'\n'`).

הניחו שאין בקובץ מילה ארוכה מ- 20 אותיות.

כתבו תכנית הקוראת את קובץ הקלם הסטנדרטי, ומדפיסה את המילה הארוכה ביותר בקובץ. אם יש יותר ממילה אחת באורך המרבי, התוכנית תדפיס את האחרונה ביניהן.

שאלה מספר 6

קבץ קלם מכיל מקסט עם תווים כלשהם.

מספר שלם חיובי חוקי: יוגדר כרצף של ספרות בלבד כשממשאלו (לפניו)

יש תמיד TAB (`'\t'`) ומימינו (אחריו), TAB או סוף שורה (`'\n'`).

הניחו שאין בקובץ מספר שלם חיובי חוקי בעל יותר מ- 20 ספרות.

כתבו תכנית הקוראת את קובץ הקלם ומדפיסה את המספר השלם החיובי החוקי הגדול ביותר בקובץ. אם יש יותר ממספר שלם חיובי חוקי מקסימאלי אחד, התכנית תדפיס את האחרון מביניהם.



תרגול נעים ומהנה !!!

שאלה מספר 1

```

int
check_back(unsigned Vec[], unsigned Sum, unsigned TempSum)
{
    int j = 0;
    while(TempSum > Sum){
        TempSum -= Vec[j++];
        if(TempSum == Sum) return j;
    }
    return NOTEXIST;
}

int
check_sum(unsigned Vec[], unsigned Sum)
{
    int i, j, BackInd, TempSum = 0;
    for (i = 0; j = 0; i < N; i++, j++){
        TempSum += Vec[i];
        if (TempSum == Sum) return i-j;
        if (TempSum > Sum) {
            BackInd = check_back(Vec, Sum, TempSum);
            if (BackInd != NOTEXIST)
                return BackInd;
            if (i < SIZE-1) i += 1 - j;
            j = 0;
            TempSum = 0;
        }
    }
    return NOTEXIST;
}

```

הערה: NOTEXIST מוגדר כך: #define NOTEXIST -1

שאלה מספר 5

```

void
Print(char Vec[], int Len)
{
    int i;
    for (i = 0; i < Len; i++)
        printf("%c", Vec[i]);
    return;
}

```

```

void
Copy(char s1[], char s2[],
int Len)
{
    int i;
    for (i = 0; i < Len; i++)
        s1[i] = s2[i];
    return;
}

```

```

#include <stdio.h>
#include <ctype.h>
#define SIZE 20
void Copy(char[], char[], int);
void Print(char[], int);
int
main()
{
    char Word[SIZE], MaxWord[SIZE];
    int c, InWord = 0, CurrLen = 0, Max = 0;
    c = getchar();
    while (c != EOF) {
        if ((InWord && (c == ' ' || c == '\t')) {
            c = getchar();
            if (isalpha(c))
                InWord = 1;
            else {
                c = getchar();
                continue;
            }
        }
        else if (InWord) {
            if (isalpha(c)) {
                Word[CurrLen++] = (char) c;
                c = getchar();
            }
            else if (c == ' ' || c == '\n') {
                if (Max < CurrLen) {
                    Copy(MaxWord, Word, CurrLen);
                    Max = CurrLen;
                }
                CurrLen = 0;
                InWord = 0;
            }
            else {
                InWord = 0;
                CurrLen = 0;
            }
        }
        else c = getchar();
    }
    Print(MaxWord, Max);
    return 0;
}

```

שאלה מספר 3

```

int
SortedRange(int Vec[], int Ind, bool Up)
{
    for (; Ind < N - 1; Ind++){
        if(Up){
            if ((Vec[Ind]) <= (Vec[Ind + 1]))
                continue;
            else break;
        }
        else {
            if ((Vec[Ind]) >= (Vec[Ind + 1]))
                continue;
            else break;
        }
    }
    return Ind;
}

int
mountain(int Vec[])
{
    int EndOfMount, Peak;
    Peak = SortedRange(Vec, 0, TRUE);
    if ((Peak == 0) || (Peak == N - 1)) return 0;
    EndOfMount = SortedRange(Vec, Peak, FALSE);
    if (EndOfMount < N - 1) return 0;
    return Peak;
}

```

הערה: בתחנה שהתבצעו ההגדרות הבאות:

```

typedef int bool;    #define N 10
#define TRUE 1       #define FALSE 0

```

שאלה מספר 4

```

int
FindMinInd(int Vec[], int Len)
{
    int i, Min = Vec[0], MinInd = 0;
    for (i = 1; i < Len; i++){
        if (Vec[i] < Min){
            Min = Vec[i];
            MinInd = i;
        }
    }
    return MinInd;
}

void
MovMin(int Vec[], int Len)
{
    int i, TempMin;
    for (i = FindMinInd(Vec, Len), TempMin = Vec[i]; i > 0; i--)
        Vec[i] = Vec[i-1];
    Vec[0] = TempMin;
    return;
}

void
Sort(int Vec[], int Len)
{
    int i;
    for (i = 0; i < Len; i++)
        MovMin(Vec + i, Len - i);
    return;
}

```