



מבוא למדעי המחשב

תרגול 5: לולאות ומערכים



תוכניה

- לולאת while, do while, for
- מערכים
- מערכים דו ממדיים





לולאות



מבוא למדעי המחשב מ' - תירגול 3

תזכורת: לולאת while

- גוף הלולאה מתבצע שוב ושוב כל עוד התנאי מתקיים

```
int number;  
scanf("%d", &number);  
while(number>0)  
{  
    printf("%d\n", number);  
    scanf("%d", &number);  
}
```

תנאי

גוף הלולאה

לולאת do-while

- בפעם הראשונה גוף הלולאה מתבצע בוודאות.
- לאחר מכן, גוף הלולאה מתבצע שוב ושוב כל עוד התנאי מתקיים

```
int x;
```

```
do  
{
```

```
    scanf ("%d" , &x) ;  
    printf ("%d\n" , x*x) ;
```

```
}while (x>0) ;
```

גוף הלולאה

תנאי

לולאת for

```
int factorial = 1;
int n, i;

printf("Enter n: ");
scanf("%d", &n);
```

אתחול

בדוק
התנאי

קידום

```
for (i=2 ;i<=n ;i++)
{
    factorial *= i;
}
```

גוף הלולאה

```
printf("%d!=%d\n", n, factorial);
```

שקילות לולאות

- האם ניתן להחליף את לולאת ה-for הזו בלולאת while? ומה לגבי do-while?
- (הניחו כי n הוגדר ואותחל בנקודה קודמת בתוכנית)

```
for (int i=0; i<n; i++){  
    printf("%d",i) ;  
}
```

שקילות לולאות

מה יותר ברור?

```
for (int i=0; i<n; i++){  
    printf("%d",i);  
}
```

האם זה
תמיד שקול?

```
int i=0;  
while (i<n){  
    printf("%d",i);  
    i++;  
}
```

```
int i=0;  
do {  
    printf("%d",i);  
    i++;  
} while (i<n)
```


שקילות לולאות

מה יותר ברור?

נניח ש x הוגדר
ואותחל קודם

```
while (x < 0) {  
    scanf("%d", &x);  
}
```

```
for ( ; x < 0 ; ) {  
    scanf("%d", &x);  
}
```

ניתן להשאיר
את גוף
הלולאה ריק

ניתן להשאיר
חלקים ריקים
בכותרת

```
for ( ; x < 0 ; scanf("%d", &x) ) {  
}
```

לולאות


תרגיל 1: כתבו תכנית הקולטת מספר אי-שלילי ומדפיסה את סכום הספרות. למשל, אם המספר הנקלט הוא 234114, הפלט יהיה 15.
אין צורך לוודא כי הקלט אי-שלילי.

באיזו סוג
לולאה כדאי
להשתמש?

```
int n, sum = 0;
printf("Please enter a non-negative number: ");
if( scanf("%d", &n) < 1)
    return 1;
/* each iteration we add the last digit of the
   number, and reducing it from the number. */
while (n > 0) {
    sum += n%10;
    n /= 10;
}
printf("sum of digits is %d\n", sum);
```

לולאות

תרגיל 2: כתבו תכנית הקולטת 10 מספרים, מחשבת ומדפיסה את סכומם.



באיזו סוג
לולאה כדאי
להשתמש
הפעם?

```
#define N 10

int main()
{
    int i;
    double x, sum = 0;
    printf("Please enter %d numbers: ", N);

    /* Gets N numbers from the user and sum them */
    for (i = 1; i <= N; i++)
    {
        if (scanf("%lf", &x) < 1)
            return 1;

        sum += x;
    }
    printf("sum = %f", sum);
    return 0;
}
```

לולאות

תרגיל 3: כתבו תכנית המחשבת את ממוצע המספרים החיוביים בקלט.

```
sum = 0; count = 0;
while (scanf("%d", &num)==1) {
    if (num < 0)
        continue;
    sum += num;
    count++;
}
average = sum / count;
```

continue מחשיך
לאיטרציה הבאה

תרגיל 4: כתבו תכנית המקבלת שני מספרים, טווח התחלה וטווח סיום, ומדפיסה את המספר הראשון בטווח שסכום ספרותיו הוא 8, אחרת תדפיס 1-.

ניתן להניח כי המספר הראשון שהתקבל קטן מהמספר השני.

```
int low,high,tmp, res = -1;
scanf("%d %d", &low,&high);
for(int num=low; num<=high; ++num){
/*check if the digit sum of @num is exactly 8*/
    tmp = num;
    int sum=0;
    while(tmp > 0){
        sum += tmp%10;
        tmp/=10;
    }
    if(sum == 8){
        res = num;
        break;
    }
}
printf("%d" , res);
```

break חסיים את
הלולאה/switch
הכי פנימיים

מי הג'יני הרע?

```
for (i = 0; i<3; ++i) {  
    printf("make a wish: ");  
    scanf("%s", the_wish);  
    if (toohard(the_wish))  
        continue;  
    dowish(the_wish);  
}
```



```
i = 0;  
while (i<3) {  
    printf("make a wish: ");  
    scanf("%s", the_wish);  
    if (toohard(the_wish))  
        continue;  
    dowish(the_wish);  
    i++;  
}
```



מערכים



מערכים

מערך הוא סדרה של משתנים מאותו טיפוס.
את המערך מגדירים באופן הבא:

```
int grades[5];
```

טיפוס משתני
המערך

שם המערך

גודל המערך

מערכים

אפשר לאתחל מערכים במספר דרכים:

- רשימת אתחול:

```
int grades[] = { 95, 45, 62, 80, 76 };
```

- אתחול המערך לאפסים:

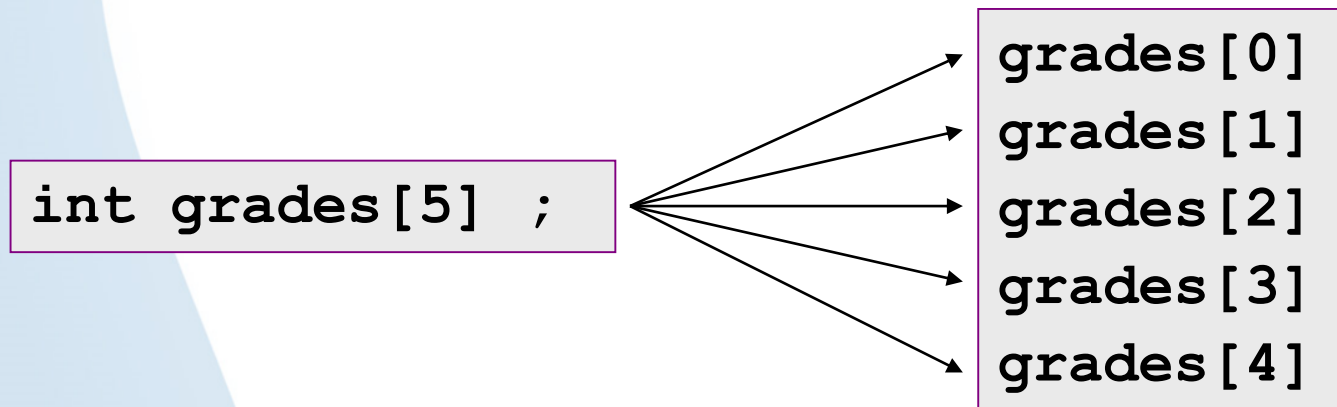
```
int grades[5] = { 0 };
```

- אתחול המערך בתחילה בערכים, ובסופו אפסים.

```
int grades[5] = { 95, 45 };
```

מערכים

איברי המערך ממוספרים מ-0 (ולא מ-1!) וניתן לגשת אליהם באמצעות אופרטור הסוגריים המרובעים:



שימו לב: חריגה מגבולות המערך אינה מהווה שגיאת קומפילציה ובמקרים מסוימים אף לא תגרום לשגיאת זמן ריצה (את הסיבה לכך נבין כשנלמד על מצביעים).

מערכים

תרגיל 5: כתבו תכנית המקבלת כקלט 10 משכורות (מספרים שלמים חיוביים), ומחזירה את הממוצע שלהן ואת המשכורת המקסימלית.

הערה: אנו נציג פתרון המשתמש במערכים, אולם ניתן היה לפתור את התרגיל גם ללא מערכים (נסו בעצמכם).

מערכים

```
#define NUM_SALARIES 10

int main()
{
    int salaries[NUM_SALARIES];
    int sum = 0, max_sal, i;
    // Initialize the array.
    for(int i=0; i<NUM_SALARIES; i++)
    {
        if (scanf("%d", &salaries[i]) < 1)
            return 1;
    }
}
```

נהוג להשתמש
ב-define כדי
להגדיר גודל
מערך

גישה לאיבר
במערך. אסור
לחרוג מגבולות
המערך!

מערכים

```
max_sal = salaries[0];  
// Sum the salaries, and finds max salary.  
for (sum=0, i=0; i<NUM_SALARIES; ++i)  
{  
    sum += salaries[i];  
    if (salaries[i] > max_sal)  
        max_sal = salaries[i];  
}  
  
printf("Max salary is %d, Average salary  
is %f",max_sal, (double)sum / NUM_SALARIES);
```

איך התכנית
תשתנה אם
נרצה לקבל
משכורות עם
ערכים
עשרוניים?

מערכים- תרגיל סיכום

תרגיל 6: כתבו תכנית הקולטת מהמשתמש מספרים שלמים (עד סוף הקלט) ולבסוף מדפיסה את חמשת המספרים האחרונים שנקלטו (לפי סדר קליטתם).

אם נקלטו עד סוף הקלט פחות מחמישה מספרים, יש להדפיס את כל המספרים שנקלטו.

ניתן להניח כי הקלט תקין (כלומר מוכנסים רק מספרים שלמים)

```
#define N 5
int counter = 0, num;
int memory[N];
while (scanf("%d", &num) != EOF) {
    if (counter == N) {
        for (int i = 0; i < N; i++) {
            memory[i] = memory[i+1];
        }
        memory[N-1] = num;
        continue;
    }
    memory[counter++] = num;
}
for (int i = 0; i < counter; i++) {
    printf("%d ", memory[i]);
}
```



מצאו את
הבעיה

```
#define N 5

int counter = 0, num;

int memory[N];

while (scanf("%d", &num) != EOF) {
    if (counter == N) {
        for (int i = 0; i < N-1 ; i++) {
            memory[i] = memory[i+1];
        }
        memory[N-1] = num;
        continue;
    }
    memory[counter++] = num;
}

for (int i = 0; i < counter ; i++) {
    printf("%d ", memory[i]);
}
```

לולאות ומערכים - תרגיל סיכום

תרגיל 7: כתבו תכנית רצף אותיות אנגליות גדולות, ובודקת האם כל האותיות ב ABC הופיעו

לולאות ומערכים

```
#define LETTERS ('Z'-'A'+1)
char c;
bool appeared[LETTERS]={false}; //all elements are 0==false
while(scanf("%c", &c)==1){
    appeared[c-'A']=true;
}
int i;
for(i=0; i<LETTERS; ++i)
    if(appeared[i]==false)
        break;
if(i==LETTERS)
    printf("All letters appeared\n");
else
    printf("Letter %c didn't appear\n", i+'A');
```

לולאות ומערכים

```
#define LETTERS ('Z'-'A'+1)
char c;
bool appeared[LETTERS]={false}; //all elements are 0==false
while(scanf("%c", &c)==1){
    if(c>='A' && c<='Z')
        appeared[c-'A']=true;
}
int i;
for(i=0; i<LETTERS; ++i)
    if(appeared[i]==false)
        break;
if(i==LETTERS)
    printf("All letters appeared\n");
else
    printf("Letter %c didn't appear\n", i+'A');
```



מערכים דו ממדיים

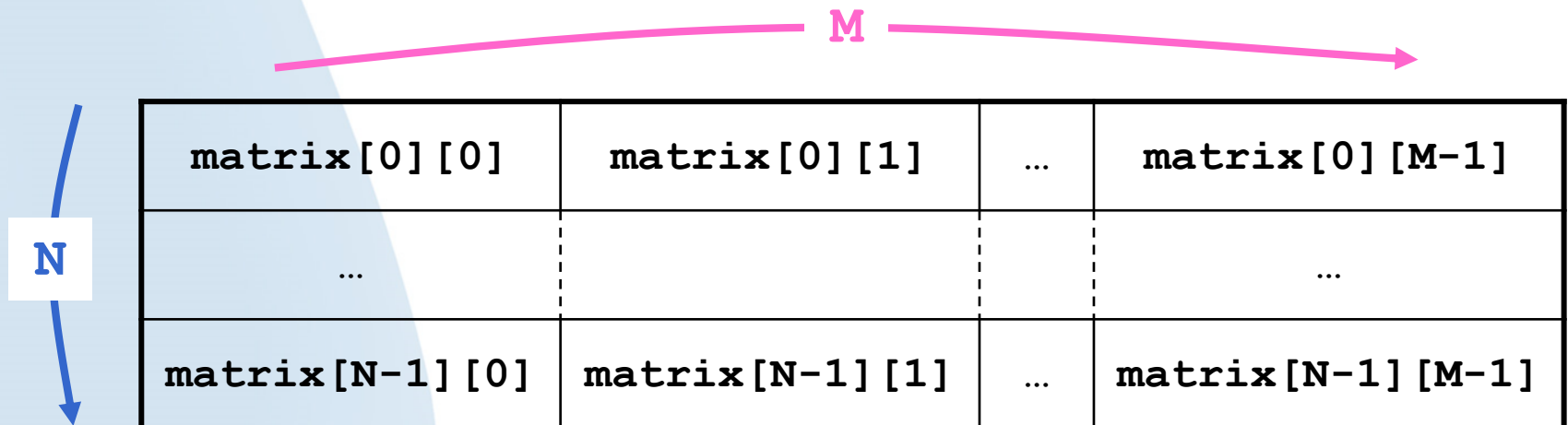


מערכים דו ממדיים

את המערך הדו-ממדי מגדירים כך:

```
double matrix[N][M];
```

נוח לחשוב על מערך זה כמטריצה דו-ממדית עם N שורות ו- M עמודות, ולשרטט את אברי המערך הדו-ממדי כך:



<code>matrix[0][0]</code>	<code>matrix[0][1]</code>	...	<code>matrix[0][M-1]</code>
...			...
<code>matrix[N-1][0]</code>	<code>matrix[N-1][1]</code>	...	<code>matrix[N-1][M-1]</code>

מערכים דו ממדיים

אתחול מערך דו-ממדי:
בעזרת רשימה:

```
int matrix[2][3] = { {5, 3, 8} ,  
                     {1, 0, 3} };
```

```
int matrix[][3] = { {5, 3, 8} ,  
                    {1, 0, 3} };
```

או כך:

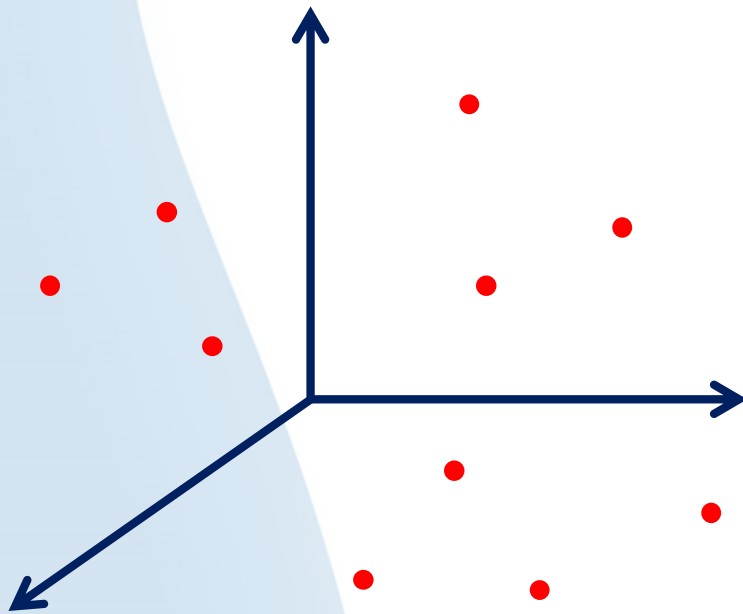
תמיד יש לציין
את אורך
השורה.

רשימה חלקית:

```
int matrix[2][3] = { {5, 3}, {1} };
```

מערכים דו-ממדיים

תרגיל 8: כתבו תכנית המקבלת מהשתמש 10 נקודות במרחב ומחשבת את מרכז המסה שלהן.



(מרכז המסה של הנקודות
היא הנקודה שערכה בכל
קורדינטה הוא הממוצע של
ערכי כל הנקודות באותה
קורדינטה)

תרגיל 8 - פתרון

```
#define N 10
double points[N][3];
double centerOfMass[3];
int i, j;
centerOfMass[0] = centerOfMass[1] = centerOfMass[2] = 0.0;
for (i = 0; i < N; i++) {
    printf("Please enter the next point: ");
    //Get a dot and adds it to the average of every axis.
    for (j = 0; j < 3; j++) {
        scanf("%lf", &points[i][j]);
        centerOfMass[j] += points[i][j]/N;
    }
}
printf("c.m. = (%f, %f, %f).\n",
        centerOfMass[0], centerOfMass[1], centerOfMass[2]);
```

תרגילי סכום

תרגיל 9: כתבו תכנית המקבלת כקלט מספר k ורשימה של 10 מספרים נוספים ומדפיסה שני מספרים שונים מהרשימה שסכומם הוא k במידה וקיימים, אחרת מדפיסה כי לא קיימים מספרים כאלה.

למשל: עבור $k=6$ והרשימה: 1,2,3,4,5,6,7,8,9,10
2 ו-4 מהווה פתרון חוקי אך 3 ו-3 לא (כי 3 מופיע פעם אחת)

עבור $k=6$ והרשימה: 1,2,3,4,3,6,7,8,9,10
3 ו-3 מהווה פתרון חוקי (כי 3 מופיע פעמיים)

```
#define N 10
int i, j, k;
int a[N];
printf("Enter k's value: ");
if (scanf("%d", &k) != 1) return 1;

printf("Please enter %d numbers: ", N);
for (i = 0; i < N; i++) {
    if (scanf("%d", &a[i]) != 1) return 1;
}
for (i = 0; i < N; i++) {
    for (j = i + 1; j < N; j++) {
        if (a[i] + a[j] == k) {
            printf("Found %d and %d.\n", a[i], a[j]);
            return 0;
        }
    }
}
printf("Not found.\n");
```

תרגילי סכום

תרגיל 10: כתבו תכנית המקבלת מערך דו-ממדי של מספרים שלמים בגודל 5×5 (המספרים מתקבלים שורה שורה, משמאל לימין), ומדפיסה את המספר הגדול ביותר של איברים סמוכים באותה עמודה שהם ממוינים בסדר לא יורד.

1	12	8	4	<u>7</u>
9	3	11	7	<u>9</u>
24	11	19	6	<u>12</u>
17	4	2	3	<u>16</u>
18	12	7	5	9

למשל, עבור המערך הבא יוחזר 4:

תרגיל 10 - פתרון

אתחול המערך:

```
#define N 5
int mat[N][N];
for (int i = 0; i < N; i++) {
    for(int j=0; j < N; j++){
        if(scanf("%d", &mat[i][j])<1) return 1;
    }
}
```

תרגיל 10 - פתרון

```
int max_len = 0, r, c, len;
for (c = 0; c < N; c++) {
    len = 1;
    for (r = 1; r < N; r++) {
        //The sequence ended at r-1
        //if current item is smaller
        if (mat[r][c] < mat[r-1][c]) {
            if (len > max_len) max_len = len;
            len = 1;
        } else
            len++;
    }
    if (len > max_len) max_len = len;
}
printf("Max length is %d", max_len);
```