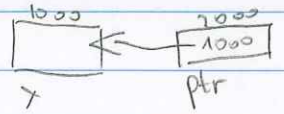
מצביעים - pointer

```
int x;
int *ptr;        // int *ptr=NULL;
  ptr=&x;
```



הגדרה של הערך הנמצא

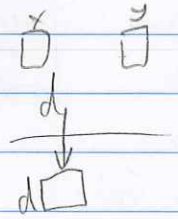
```
*ptr = 20 // x = 20;
```

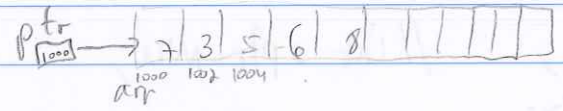
```
void swap(int *x, int *y)
{
  int tmp = *x;
  *x = *y;
  *y = tmp;
}

void main()
{int x=10, y=5;
 swap(&x, &y);
}
```

```
int fun(int x, int y, int *d)
{
  *d= x/y;
  return x*y;
}

void main()
{ int x=10, y=5, m, d;
  m= fun(x, y, &d)
}
```






```
int arr[10];
(arr+2*sizeof(int)) arr[2]=5;
int *ptr;
ptr=arr; / ptr=&arr[0];
```

הדפסה: הפלט הראשון כמעלה `cout<<arr[0]<<*ptr<<*arr<<ptr[0];`

```
x=arr[2]; / x=*(arr+2);
```

`ptr++;` מצביע אחרי הכל כמעלה

`cout<<*(ptr+2)<<ptr[2]<<arr[2]` הפרמטר לפונקציה שקבלה מערך.

`void fun(int a[3]) / void fun(int *a)` `ptr=ptr+2` מצביע על האיבר השני בכל

```
for(ptr=arr; ptr<arr+10; ptr++)
    cout<<*ptr;
```

```
for(ptr=arr; ptr<arr+10;)
    cout<<*ptr++;
    cout<<(*ptr)++;
    cout<<++*ptr;
    cout<<++(*ptr);
```

מקדם את המצביע ומדפיס את המצביע הקודם

מגדיל את המצביע ומדפיס אותו

מגדיל את המצביע ומקדם אותו

מגדיל את המצביע ומגדיל אותו (לא משנה הסדר)

הדפסת יחידות לזיכרון	הדפס
הדפס	הדפס
הדפס מצביע עכשיו לשלמות	הדפס
הדפס כמה מילות	הדפס
אז לחבר 2 מילות	אז לחבר 2 מילות

`ptr++*` - מקדם למצביע של הבתובת הבאה

* כשעוברים סדרתית על מחרוזת חייבים `while(*ptr)`

פעולות וקבועים

להלן מפת זיכרון:

2000	34	x
2010	17	y
2020	2000	ptr
2030		

ענו על השאלות הבאות:

- תא x נמצא בכתובת 2000
- תא y נמצא בכתובת 2010
- תא ptr נמצא בכתובת 2020
- רשמו על מפת הזיכרון את הפעולה y = 17
- כתבו במילים את הפעולה ptr = &x ל ptr יש כתובת x
- רשמו במפת הזיכרון את הפעולה ptr = &x
- כתבו במילים את הפעולה: *PTR = 34 השמה שנלך בכתובת שהכיל עליו ptr יקבל 34
- רשמו במפת הזיכרון את הפעולה הזו.
- רשמו פעולה אחרת השווה בתוכנה

x=34;

נתונים שני תאים Y,X ושני מצביעים PX ו-PY המצביעים על התאים X ו-Y בהתאמה, ז"א הציבו px=&x ו- py=&y.

להלן מספר פעולות רשמו את הסברן:

1.	printf ("%d",*px)	מדפיס את ערך x. שמוכלת ע"י px
2.	printf ("%p",&px)	מדפיס את הכתובת של px
3.	printf ("%p",px)	מדפיס את הכתובת של x שבה הערך של px
4.	*px=0	השמה אפס ל-x
5.	d=sqrt(*px)	שורש	השמה שורש של x לשמנה d
6.	py=px	ל py יש אותו הכתובת של x
7.	px=&y	ל px יש את הכתובת של y
8.	y= *px+1	y שווה ל px ש ל y + 1 אין שום אחרי אחרי x+1
9.	y= *(px+1)	שגיאה- צריך לכתוב הקאה ולא מערך
10.	*px+=1	מוסיף לערך שמוכלת ע"י px 1
11.	*px++	מקדם לכתובת הבאה
12.	(*px)++	מוסיף לתוכן המוכלת ע"י px 1

מצביעים

1) לפניך תמונת זיכרון של מחשב וקטע תוכנית. מה תהיה תמונת הזיכרון לאחר הרצת התוכנית?

Variable	Address	Value
ptr	2000	2000 ???
	2002	???
	2004	???
x	3000	2 2 8
arr	3002	2 0
	3004	0 3
	3006	20 3
	3008	17 7

```
Main()
{
   int x=0, arr[4]={0,3,20,17}, *ptr;
   ptr=&x;          ptr= 3000
   *ptr=x+2;        x=2
   ptr=arr;         ptr= 3002
   *ptr=x;          arr[0]= 2
   ptr++;           ptr= 3004
   x=*(ptr+1);      x=20
   *ptr=0;          arr[1]=0
}
```

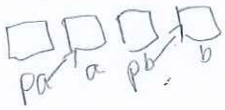
2) לפניך מערך שלמים ושני מצביעים. מה יהיו ערכי המערך לאחר הרצת התוכנית:

```
main()
{
   int array[]={1,2,3,4,5,10,20,30,40,50,60,70,80,90};
   int *ptr1, *ptr2;
   for(ptr1=ptr2=array; (ptr1-ptr2) < 10 ; ptr1+=2)
        *ptr1 = *ptr2;
   *(ptr2+12)=110;
}
```

מצביעים דף 2

3) נתונה התוכנית הבאה ומצב הזכרון בתחילתה:

```
int a, *pa, b, *pb
a=7;
b=5;
pa=&a;
pb=&b;
}
```



&pa: 1000
&a: 1002
&pb: 1004
&b: 1006

a, b, a++, b++, pa, pb, pa++, pb++, *pa++, *pb++, (* pa)++,
(*pb)++, ++*pa, ++*pb, ++(*pa), ++(*pb), &a, &b, &pa, &pb

3) א. תארי מה מבצעת התוכנית הבאה.

```
#include <stdio.h>
int x[10];
int y[10];

void sum(int *pnt_x,int *pnt_y);

main()
{
        sum(x,y);
}

void sum(int *pnt_x, int *pnt_y)
{
        int i;
        *(pnt_y) = *(pnt_x);
        for(i=1;i<10;i++)
                *(pnt_y+i) = *(pnt_y+i-1) + *(pnt_x+i);
}
```

ב. מה יהיו ערכי מערך y לאחר ביצוע התוכנית, אם מערך x יהיה:

x[10]={6,2,8,12,0,3,2,0,6,1};

תרגיל

נתבי פונקציה המקבלת מערך, אורך המערך ערך x ומחזירה את מצביע הראשון
ל x במערך (אם לא מחזיר NULL)

```
int*fun( int *arr, int x, int len)
{
    int *ptr = arr;
    for( ptr ; ptr - arr <len; ptr++)
        if (*ptr==x)
            return ptr;
    return NULL;
}
```

```
for(ptr=arr;ptr<arr+len&&*ptr!=x;ptr++);
return ptr<arr+len?ptr:NULL;
```

ש.ב.

1) נתבי פונקציה המקבלת מערך ואורך מערך ומחזירה את המערך האקסיטי והמערך הסינגלי
2) " " " במערכים הנ"ל ומחזירה שמצביע למערך שבו סכום האברים
הגדול ביותר, והשונות הכי גדולה מ main אל מקום המערך
3) נתבי פונקציה המקבלת מערך מספרים השלמים ו n- והפונקציה מחזיר מספר האורך
המשותף של כל המספרים הכי הרבה ספרות, אם כולם המספרים, שמאלה ומה מצביע המס
הגדול ביותר במערך
4) נתבי strlen באמצעות מצביעים
מקבלת מחרוזת ומחזירה את מס' התווים במחרוזת לא כולל ה NULL

תשובות - ש.ר.

מס'

```
int * fun( int * arr, int len, int *max)
{ int *ptr=arr, min = 32,767;
  for(;ptr < len; ptr++)
  if (*ptr < min)
      min = *ptr;
      if (*ptr > *max)
          *max= *ptr;
  }
  return min
void main ()
{ int arr[5] 1,2,3,4,5,6; max=0, len= 6
  cout<< fun(arr, len, &max << max;
}
```

(1)

```
int * fun(int * a , int *b)
{ int *p1=a, *p2=b;
  for(p1; p1+=*p1 && *p1!=-1; p1++);
  for(p2; p2+=*p2 && *p2!=-1; p2++);
    if (p1>p2)
        return
```

(2)

int * all (3

int mystrln(char *ch) (4
{
int ptr=*ch, i, j=0;
for(i; ch[j]; j++);
i=*ch+j;
return i-ptr;
}

void main()
{
char ch[]= " ";
cout << mystrln(ch);
getch()
}

סמני את ההוראות שאינן עוברות קומפילציה//

```
void main()
{
    int arr[20]={0}, *ptr, num;
    ptr=arr;

  ✗arr++;          arr שם של מערך הוא קבוע ולא ניתן לשנותו
                   לא יכול לקבל כתובת של משתנה
  ✗ arr=&num;
  ✓ptr++;          תקבור להצביע במקום הבא
  ✓ ptr=&num;      ptr מצביע למשתנה ptr קבל כתובת של משתנה
  ✓*ptr=8;         למקום המוצבע ע"י ptr ותשים בו 8
  ✗*num=8;
  ✗arr+2=8;        כתובת, ולכן אלה לשנות
  ✗&num=8;         מוחקים את הכתובת "לשתנה מצביע
                   לא ניתן לשנות)
  ✗ &num=&arr[5];  כתובת arr[5]-מוקי אבל אבל
  ✓num=*ptr;       שים את הערך של מה שמוצבע
  ✓*(arr+19)=8;    (ערך עצום בעצמו - num) ptr ע"י num
                   arr כתובת של מערך במקום הוא תשים
  ✓(*arr)++;       קידום סופי - תוכן של האינדקס arr
  ✗*arr++;         קידום לכתובת - לא ניתן לשנות
  ✓ *arr=*ptr==*arr;  == בימוי לוגי ולכן את הבוליאנ שמים באינדקס הראשון המערך
  ✓int *ptr2;
  ✗*ptr2=ptr;      לא יכול להיות *int לint ובגלל לא מצביע לשום דבר
  ✓ptr2=&*arr;     - האינדקס arr*-הראשון ומשתמשים את הכתובת  ptr2 = arr
  ✓ num=*++ptr2;
}
```

מצביע לאינדקס הבא במערך
ואז תקח את התוכן שלו
ותשים במשתנה