

מבוא למדעי המחשב

חזרה,

מצביעים והעברת פרמטרים לפונקציות

חזרה – חישוב פאי

מצביעים

■ מצביע הוא משתנה שמכיל כתובת

```
int* p;
```

```
int a = 5; // מאוחסן במקום כלשהו בזיכרון
```

```
p = &a; // הכתובת של תא הזיכרון שמכיל את הערך
```

```
int* p2 = p;
```

```
int b = *p2; // הערך שנמצא בכתובת - במקרה זה 5
```

שימושים נפוצים למצביעים

- לאחסון כתובת של מערך דינמי פרימיטיבי:

```
int * arr = new int[10];
```

- העברת פרמטרים לפונקציות ללא העתקתם
- אחסון כתובת של חומרה (קלט עכבר, מקלדת, פלט מסך...)

תזכורת פונקציות:

מנגנון הקריאה לפונקציות:

- ◆ שמירה של מצב התוכנית.
- ◆ קריאה לפונקציה העברת פרמטרים, פתיחה של סביבה חדשה במחסנית.
- ◆ חזרה מפונקציה, ערך מוחזר, שחזור מצב התוכנית (pop).
- ◆ פונקציה יכולה להקצות זיכרון (מערך) ולהחזיר מצביע עליו (לערימה) - דורש שחרור!

העברת פרמטרים לפונקציות:

נניח שאנו מעוניינים לכתוב פונקציה שמחליפה בין שני ערכים של משתנים שלמים:

```
int x=2,y=5;  
cout<<"Orig: x: "<<x<<" y: "<<y<<endl;  
swap(x,y);  
cout<<"Swap: x: "<<x<<" y: "<<y<<endl;
```

העברת פרמטרים לפונקציות:

דוגמא ראשונה: שימוש במצביעים (pointer):

```
void swap1(int* p, int* q) {  
    int t=*p;  
    *p=*q;  
    *q=t;  
}
```

```
swap1(&x, &y);
```

קריאה לפונקציה:

העברת פרמטרים לפונקציות:

דוגמא שנייה: שימוש ברפרנס (אפשר לחשוב על זה כאילו המשתנה המקורי שנשלח מקבל שם חדש בפונקציה – ממומש בד"כ ע"י מצביע חבוי)

```
void swap2(int& p, int& q) {  
    int t=p;  
    p=q;  
    q=t;  
}
```

קריאה לפונקציה:

```
swap2(x, y);
```


העברת פרמטרים לפונקציות:

דוגמא שלישית: רק ערכים – הפונקציה לא עושה כלום!!!

```
void swap3(int p, int q) {  
    int t=p;  
    p=q;  
    q=t;  
}
```

קריאה לפונקציה:

```
swap3(x, y); // does nothing!!!
```

מערכים – העברה לפונקציה - תזכורת:

- העברה לפונקציה מתבצעת תמיד ע"י העברת מצביע לתחילת המערך ולכן ניתן לשנות את איברי המערך בתוך הפונקציה.

העברת פרמטרים לפונקציות:

דוגמא שנייה: שימוש בconst רפרנס: קלט שלא מועתק ואי אפשר לשנות בפונקציה

```
void swap2(const int& a, int& b) {  
    b = 2*a; // compiles ok  
    a = -3; // does not compile!!!  
}
```

למה זה טוב? להימנע משינויים בטעות.

קודים:

insertion sort

merge_sort