# הקצאת זיכרון באופן דינאמי
# (Dynamic memory allocation)

```
#include <stdlib.h>
void *malloc(size_t number_of_bytes);
void free(void* p);

ip = (int *) malloc(j*sizeof(int));
for (i=0; i<j; i++) ip[i]=10;
free(ip);
if (ip != NULL)
   for (i=0; i<j; i++) ip[i]=10;
```

← לשימוש רק אחרי malloc

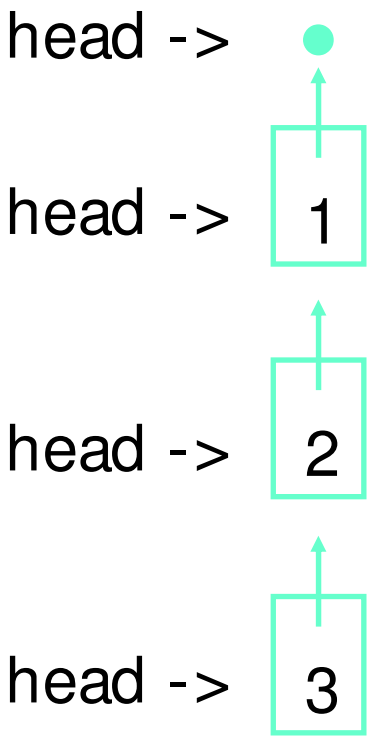
```
if (!ip) exit(1);
```

# דוגמא: שרשרת (Linked List)

```
#include<stdlib.h>
#include<stdio.h>

struct list_el {
   int val;
   struct list_el *next;
};

typedef struct list_el item;
```

```
main() {
   item *curr, *head;
   int i;

   head = NULL;

   for (i=1; i<=3; i++) {
      curr = (item *)malloc(sizeof(item));
      curr->val = i;
      curr->next  = head;
      head = curr;
   }

   curr = head;

   while(curr) {
      printf("%d\n", curr->val);
      curr = curr->next ;
   }
}
```

# דוגמא: שרשרת (Linked List)




3

2

1

```
main() {
   item *curr, *head;
   int i;

   head = NULL;

   for (i=1; i<=3; i++) {
      curr = (item *)malloc(sizeof(item));
      curr->val = i;
      curr->next  = head;
      head = curr;
   }

   curr = head;

   while(curr) {
      printf("%d\n", curr->val);
      curr = curr->next ;
   }
}
```

# אותיות ומחרוזות

דוגמא 1:

```
char c;
c='a';
```

דוגמא 2:

```
int i;
char a[5];
for(i=0; i<5; ++i) scanf("%c",&a[i]);
for(i=4;i>=0;--i) printf("%c",a[i]);
printf("++++\n");
for(i=4;i>=0;--i) printf("%i\n",a[i]);
```

קלט:

```
;
bc
```

השפעה:

a[0]=';'

a[1]='\n'

a[2]='b'

a[3]='c'

a[4]='\n'

פלט:

```
cb
;++++
10
99
98
10
59
```

# מחרוזות (strings)

- מחרוזת הינה מערך של איברים מסוג char — char a[5];
- במחרוזת n+1 תאים - האחרון קוד סיום — ‘r’ ‘u’ ‘n’
- מחרוזת היא מערך ולכן לא ניתן לבצע השוואה או השמה באופן ישיר — char a[5], b[5]; a=b; if (a==b) … false
- העברת נתונים בין מחרוזות – פונקציות עזר ב- string.h
- ניתן לאתחל:

char name[7];

name=“Rennan”;

char name[7]={‘R’,’e’,’n’,’n’,’a’,’n’,’\0’};

או

char name[7] =“Rennan”;

או

char name[] =“Rennan”;

# מחרוזות (strings)

$'a' \neq "a"$ $\qquad$ $"a" = \{'a','\backslash 0'\}$

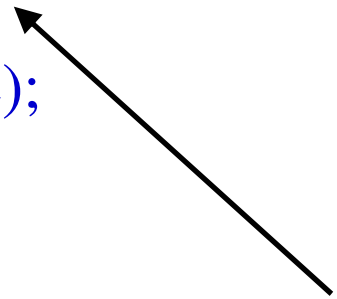
null character

```
char name[40]=“Rennan”;
printf("%s\n",name);
scanf("%s",name);
printf("%s\n",name);
```

Rennan

Barkana :קלט

Barkana



scanf עובד עם מצביע

# מחרוזות (strings)

```
#include <string.h>
char a[10]="Rennan",b[10];
strcpy(b,a);
```

a ==> b

```
char* mystrcpy2(char *s, char *t)
{
   char *save=s;
   while (*s++ = *t++) ;
   return save;
}
```

```
char* mystrcpy1(char *s, char *t)
{
   char *save=s;
   while ((*s = *t) != '\0') {
      s++;
      t++;
   }
   return save;
}
```

פעולות על משתנה אחד (כמו * או ++) מבוצעות מימין לשמאל.

# מחרוזות (strings)

- פונקציות מיוחדות ב- string.h
  - size_t strlen(s) : אורך לא כולל \0
  - int strcmp(s,t) : 0 אם שווים(!!)
  - char *strcpy(s,t) : t ל- s, כולל \0, מחזיר את s.
  - char *strcat(s,t) : מוסיף t ל- s, עם \0 בסוף, מחזיר את s.

```
char a[10]="One";
printf("%s\n",strcat(a,"Two"));
```

OneTwo