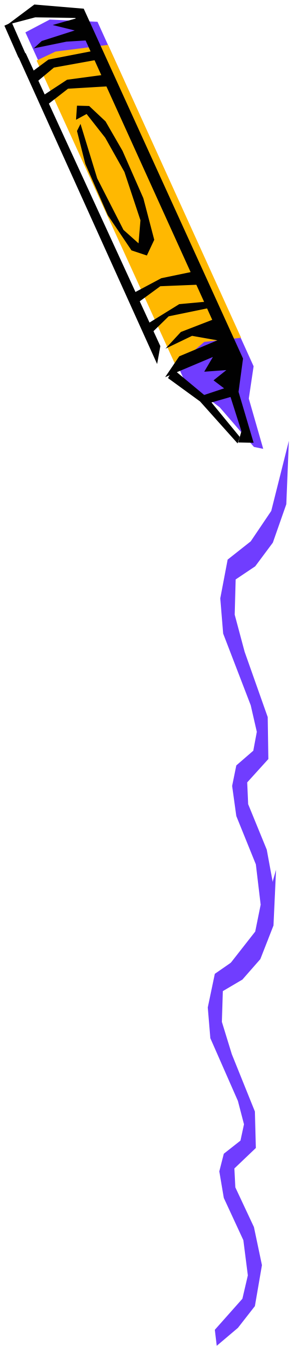
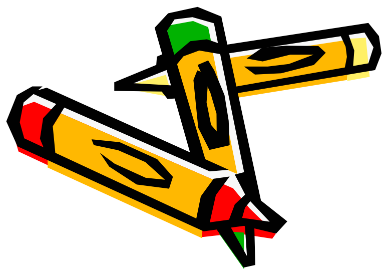
Relational Operators פעולות של יחסים

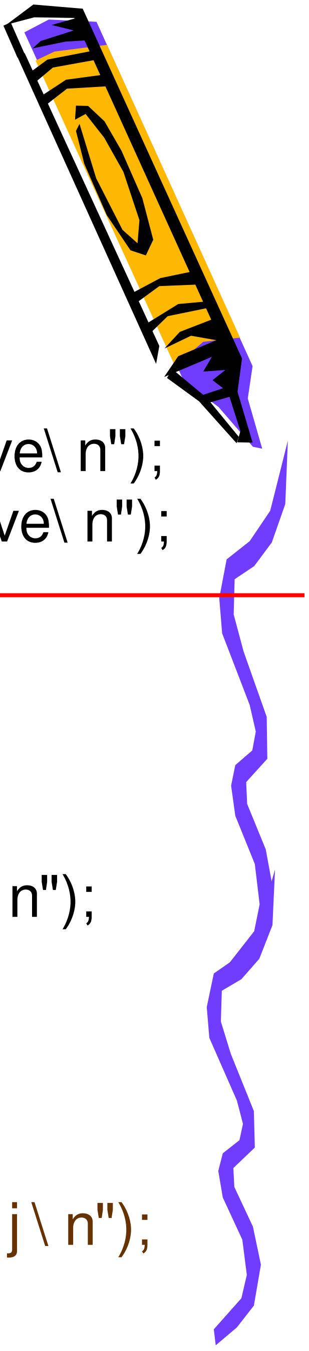


סימן ב-C	תיאור	סימן מתמטי
==	שווה	=
>	גדול	>
<	קטן	<
>=	גדול או שווה	≥
<=	קטן או שווה	≤
!=	לא שווה	≠

1) נכון <= 1
2) לא נכון <= 0
3) לא אפס <= נכון



if -else משפטי בקרה:



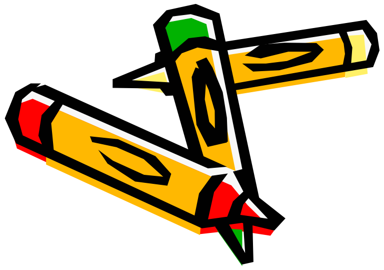
```
if (condition) statement;

if (total>0) printf("OK\n");
```

```
if (condition) statement 1;
else statement 2;

if (a<0) printf("a is negative\n");
else printf("a is nonnegative\n");
```

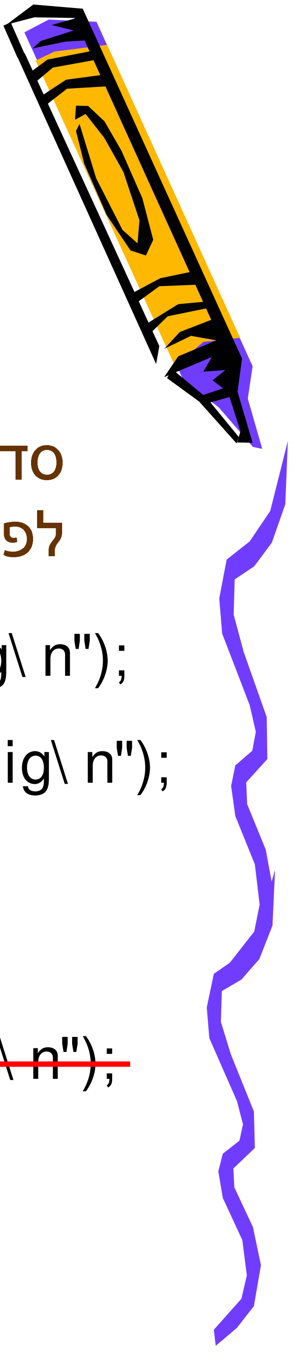
```
if (condition)
{ block of statements }
else
{ block of statements }
```

```
if (i<j) printf("i is smaller\n");
else
{
    if (i == j) printf("i equals j\n");
    else printf("i is larger\n");
}
```

```
if (i<j) printf("i is smaller\n");
else if (i == j) printf("i equals j\n");
else printf("i is larger\n");
```



if - else משפט



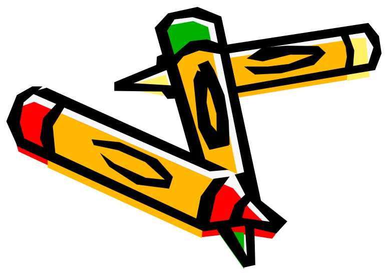
```
if (condition1)
   { if (condition2)
        { block 2 yes; }
      block of statements; }
else
   { if (condition3)
       { block 3 yes; }
      else
       { block3 no; }
      block of statements; }
```

סדר פעולות: אריתמטיקה
לפני יחסים

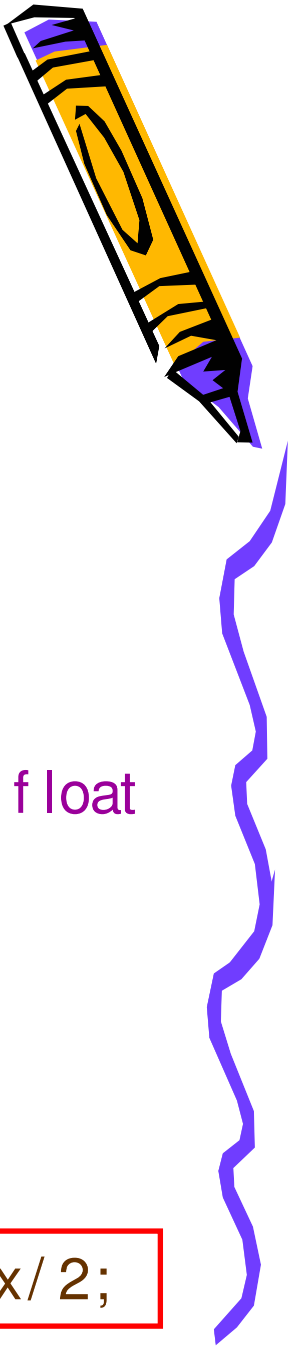
```
if (i>j+1) printf ("i is big\n");

if (i>(j+1)) printf ("i is big\n");
```

```
float a,b;
```
~~if (a == b) printf (“a = b\n");~~
```
if (fabs(a-b) < 1.e-6)
    printf (“a = b\n");
```

```
#include <math.h>
```



משפט בקרה מקוצר



```
if (condition) statement 1;
else statement 2;

(condition) ? expression1 : expression2;
```

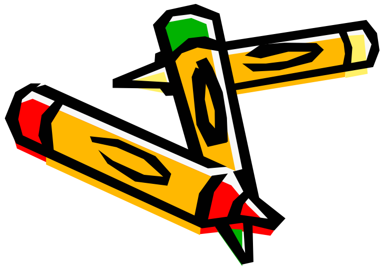
```
max = (a > b) ? a : b;

if (a > b)
   max=a;
else
   max=b;
```

```
int n;
float f;
(n > 0) ? f : n;
```

$\longrightarrow$ float

```
# include <math.h>
a = (x != 0) ? (exp(x)-1)/x : 1;

a = (fabs(x) < 1.e-6) ? (exp(x)-1)/x : 1;
```

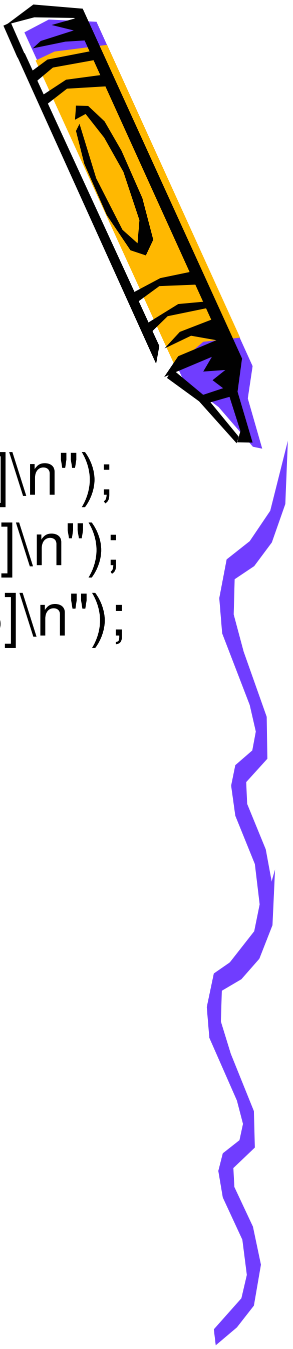
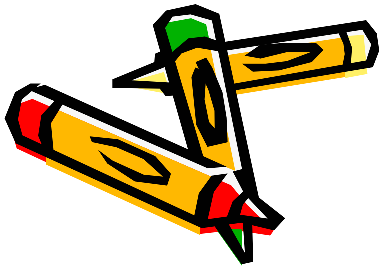
$$\boxed{\texttt{a = (fabs(x) < 1.e-6) ? (exp(x)-1)/x : 1+x/2;}}$$



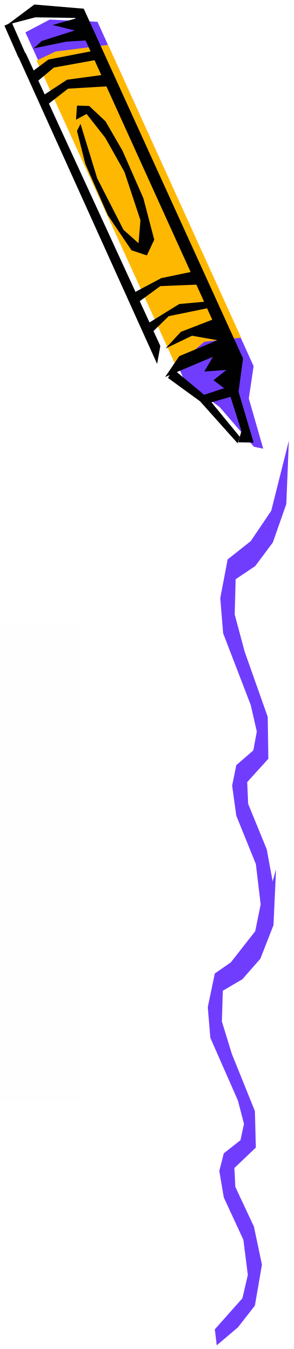
switch משפט

```
switch (expression)
   { case const-expr:
          statements;
     case const-expr:
          statements;
     default:
          statements; }
```

```
int menu;

printf("Main Menu\n");
printf("Rotate a vector      [1]\n");
printf("Invert a matrix        [2]\n");
printf("Find a determinant [3]\n");
scanf("%i",&menu);
switch (menu)
   {
     case 1:
     /* Vectors: */
     break;
     case 2: case 3:
     /* Matrices: */
     break;
   }
```

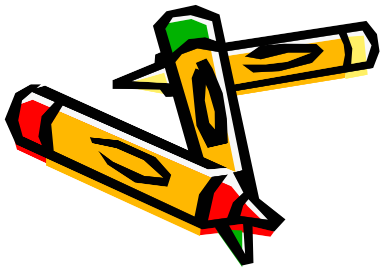



Logical Operators פעולות לוגיות



Operator	Meaning
&&	AND
||	OR
!	NOT

X1	X2	!X1	X1 && X2	X1 || X2
TRUE	TRUE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE
FALSE	FALSE	TRUE	FALSE	FALSE



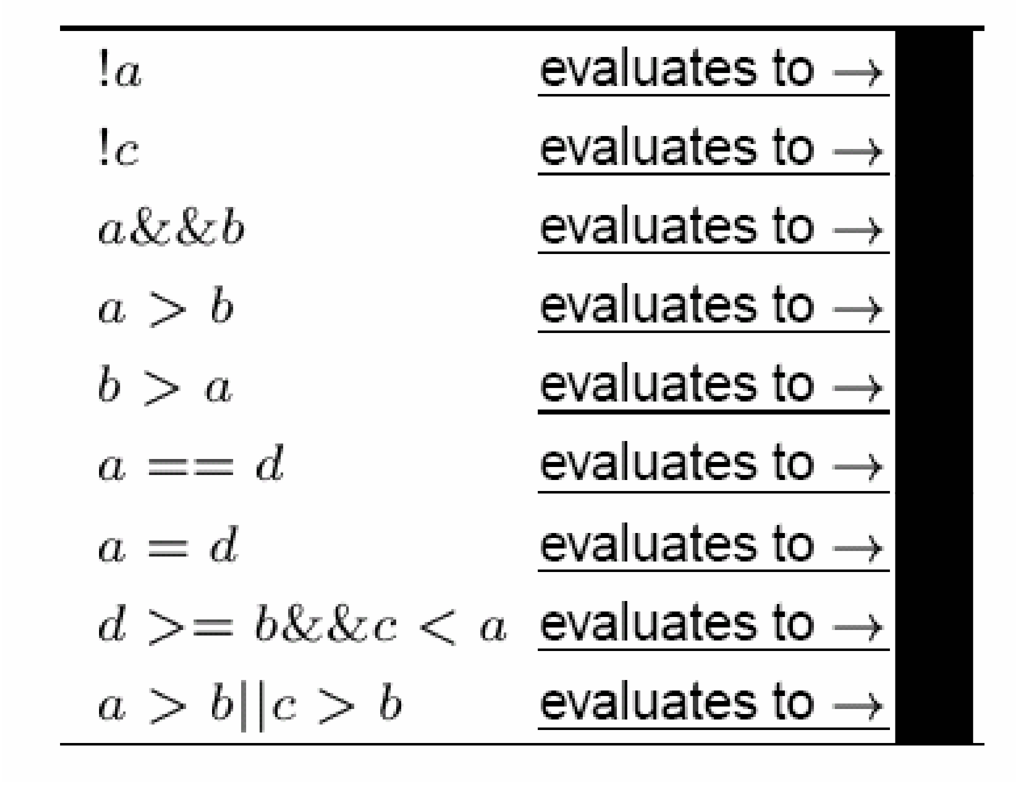
Logical Operators פעולות לוגיות

סדר פעולות:יחסים לפני לוגיקה

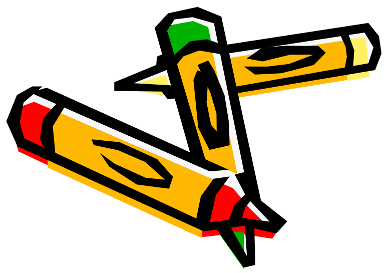
```
if (first_initial == 'A' && last_initial == 'G' || id == 321)...
```

```
if (((first_initial == 'A') &&
    (last_initial == 'G')) || (id==321))...
```

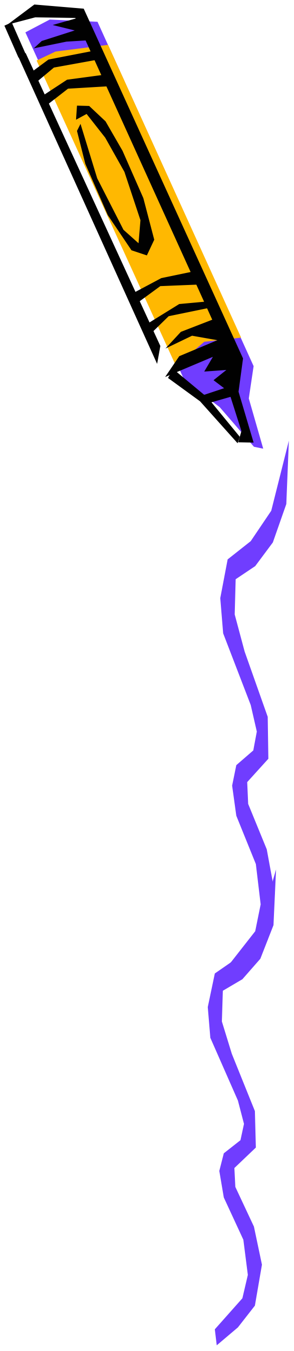
a=10, b=5,
c=0, d=5



$!a$	evaluates to $\rightarrow$	[redacted]
$!c$	evaluates to $\rightarrow$	[redacted]
$a\&\&b$	evaluates to $\rightarrow$	[redacted]
$a > b$	evaluates to $\rightarrow$	[redacted]
$b > a$	evaluates to $\rightarrow$	[redacted]
$a == d$	evaluates to $\rightarrow$	[redacted]
$a = d$	evaluates to $\rightarrow$	[redacted]
$d >= b\&\&c < a$	evaluates to $\rightarrow$	[redacted]
$a > b\|\|c > b$	evaluates to $\rightarrow$	[redacted]



Loops לולאות



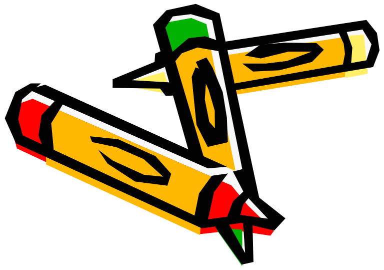
while (condition) statement;

do statement while (condition);

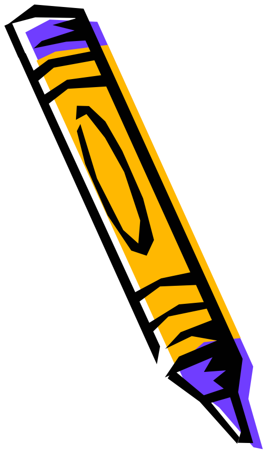
```
i=1;
while (i<10)
 {
   printf("%i \n",i);
   i++;
 }
```

$\boxed{i=10;}$

```
i=1;
do
 {
   printf("%i \n",i);
   i++;
 }
while (i<10);
```



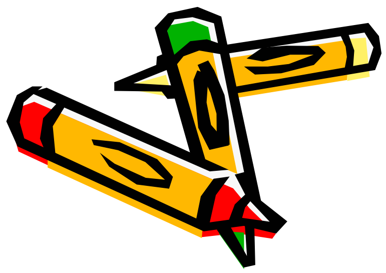
Loops לולאות



```
for (initial; condition; statement 1)
    statement 2;

initial;
while (condition)
   {statement 2;
    statement 1;}
```

```
i=1;
while (i<10)
 {
   printf ("%i \n",i);
   i++;
 }
```

```
for (i=1; i<10; i++)
   printf ("%i \n",i);
```



```
for (i=1, j=5; i<j; ++i)
     printf ("%i \n",i) ;
```