


Notes:

لما بنستخدم ال double بنحتاج ال format specifier ← %f وعند العرض ال %f
بيشغل عادي زيه زي ال float

```
1 #include <stdio.h>
2 int main()
3 {
4     double x;
5     double y;
6     printf("Enter two double num:");
7     scanf("%f %f",&x,&y);
8     printf("the Entered two double num:%0.1f and %0.1f",x,y);
9 }
10
```



نلاحظ ان لما استخدمنا متغير X من نوع double مع ال %f بس الناتج طلع ب 0.0 وده
معناه ان الانبوت مدخلش في المتغير زي ما احنا عايزين لكن لما استخدمنا متغير y مع
ال %f ظهرت النتيجة في الاوتبوت وده الي احنا عايزينه

2-Bitwise operators :

ملحوظه : نراجع ال number systems قبل ما تبدا

0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

1- AND



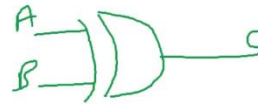
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

2- OR



A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

3- XOR



A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

4- Complement



A	B
0	1
1	0

ال and يشبه عملية الضرب بينما ال or يشبهه عملية الجمع

ال XOR المتشابه بصفر والمختلف بواحد (مهم لانه يستخدم كثير)

ال NOT هو Unary operator

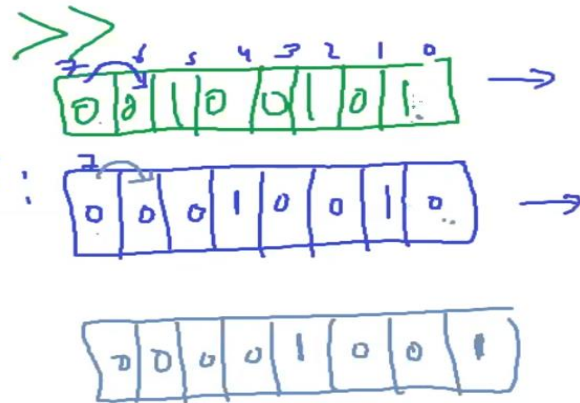
شكلهم في الكود ال & and ال و ال | OR و ال ^ XOR و ال ~ NOT

5- shift right

Assume Number:

execute one shift:
right

execute one more
shift right




بنستخدم RIGHTshift او ال leftshift باستخدام ال >> او <<

والأماكن الفاضيه بتبقي بصفر ممكن كمان يتنفذ ع الباينري او الديسمال او الهيكسا او غيرها من الانظمه العدديه

ملحوظه : %x بتطبع بال hex

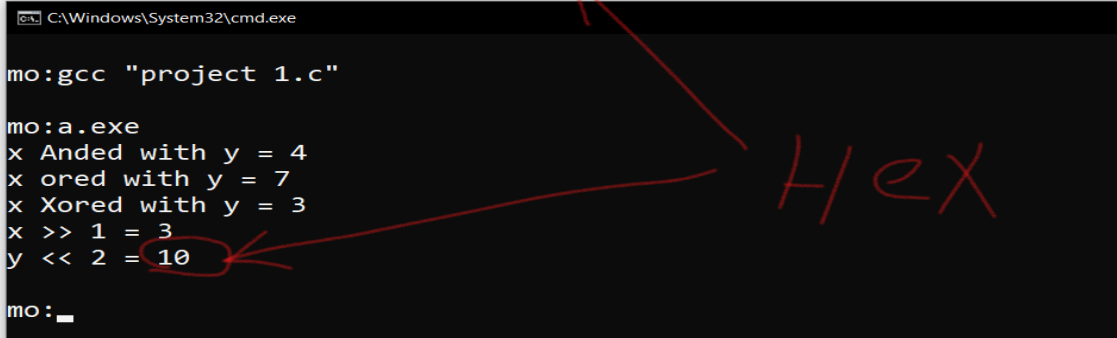
مثال:

```
1 #include <stdio.h>
2 int main()
3 {
4     int x=7,y=4;
5     printf("x Anded with y = %d\n",x&y);
6     printf("x ored with y = %d\n",x|y);
7     printf("x Xored with y = %d\n",x^y);
8     printf("x >> 1 = %d\n",x>>1);
9     printf("y << 2 = %d\n",y<<2);
10 }
11
```



ولو لاحظنا التغيري البسيط في الكود هنتأكد ان 16 في ال Decmal يقابلها 10 في ال hex

```
1 #include <stdio.h>
2 int main()
3 {
4     int x=7,y=4;
5     printf("x Anded with y = %d\n",x&y);
6     printf("x ored with y = %d\n",x|y);
7     printf("x Xored with y = %d\n",x^y);
8     printf("x >> 1 = %d\n",x>>1);
9     printf("y << 2 = %x\n",y<<2);
10 }
11
```



ملحوظه مفيش حاجه بتعرض ال binary represintaion زي ال %x لل hex او ال %d للديسمال

Logical operators:

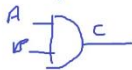
True → any value other than zero

False → zero

بس في الاخر الكومبايلر بيترجم الترو علي انه 1 او 0xff

1- Logic AND ($\&$)

A	B	C
F	F	F
F	T	F
T	F	F
T	T	T



2- Logic OR ($\|$)

A	B	C
F	F	F
F	T	T
T	F	T
T	T	T



3- Logic NOT (!)

A	B
T	F
F	T



مثال:

int X=5 , K=0 , Y=10 , Z ;

Z = $\overset{T}{X} \& \& \overset{T}{Y}$; → True

Z = $! \overset{T}{X}$; → false

Z = $\overset{f}{K} \| \overset{T}{X}$; → True

Z = $K \& X$; → false

Assignment operators

=	+=	-=	*=	=	%=
&=	=	^=	>>=	<<=	+=

في المثال ده احنا بناخد القيمه الي في اليمين وبنخذنها في المتغير الي في الشمال

```
Int x =5;
```

لكن في العمليات الثانيه احنا بندمج عمليتين مع بعض
في المثال ده احنا بنجمع ونعمل assigne في خطوه واحده

```
Int x +=5;
```

وده بيساوي انك تعمل

```
Int x =x+5;
```

Relational operators

= = != > < >= <=

الناتج ديما True or False

relation operators	
operator	effect
>	Greater than
>=	Greater or equal
<	Less than
<=	Less or equal
= =	Equal
! =	Not equal

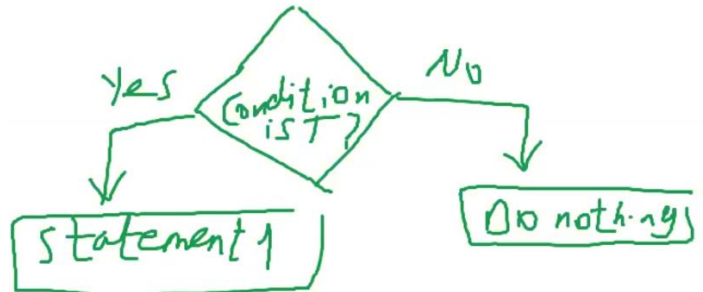
Conditional statement

عندنا في ال C طريقتين للاستخدام ال conditional statement

1-if statement:

Syntax:

```
If (<condition>)  
{  
    <statements>  
}  
else{  
    <statements>  
}
```



مثال:

Q:

Write a C code to calculate employee salary in a week based on the his working hours, hour rate is 50.

The program will ask the user to enter the working hours, then it will print his salary.

But if the working hours are less than 40 hours, a 10% deduction will be applied.

answer:

```
1 #include <stdio.h>
2 int main()
3 {   int h;
4     printf("Please Enter your Working houre: ");
5     scanf("%d",&h);
6     if(h>=40)
7     {
8         printf("your salary is %d",h*50);
9     }
10    else
11    {   h=h*50*0.9;
12        printf("your salary is %d",h);
13    }
14
15 }
16
```

```
C:\Windows\System32\cmd.exe
mo: gcc "project 1.c"
mo:a.exe
Please Enter your Working houre: 20
your salary is 900
mo:a.exe
Please Enter your Working houre: 60
your salary is 3000
mo:a.exe
Please Enter your Working houre: 50
your salary is 2500
mo:
```

نلاحظ: تم فصل في جملة ال else المتغير h عن الفانكشن printf علي عكس ال if

السبب: هو لو سبينها زي ال if printf("your salary is %d",h*50); القيمة عبارة عن int
تم ضربه في int وعرض القيمة الناتجة في الديسمل لأننا استخدمنا ال %d

لكن في الحاله الثانيه h=h*50*0.9; لو نفذناها زي المثال الي فات كان

```
printf("your salary is %d",h);
```

الي هيحصل انه قيمه int مضروبه في 50 الناتج int يضرب ثاني في قيمه float

0.9 يبقى معنا متغير float عايزين نعرضه باستخدام %d ده هينتج عنه ان القيمة الي
هتطلع هتبقى صفر

لكن بعد الفصل الي حصل هو كالاتي تم نفس الخطوات السابقه واصبح عندك ناتج float
هتأخذنه في متغير int وهتستخدم المتغير من النوع int مع %d فهتظهر الناتج .

و الحل الاخر : للخروج من حته ال float ممكن تمثّل الرقم علي هيئه كسر ديسما

```
1 #include <stdio.h>
2 int main()
3 {   int h;
4     printf("Please Enter your Working houre: ");
5     scanf("%d",&h);
6     if(h>=40)
7     {
8         printf("your salary is %d",h*50);
9     }
10    else
11    {   printf("your salary is %d",h*50*9/10);
12    }
13
14
15 }
16
```

```
C:\Windows\System32\cmd.exe
mo:a.exe
Please Enter your Working houre: 20
your salary is 900
mo:a.exe
Please Enter your Working houre: 50
your salary is 2500
mo:a.exe
Please Enter your Working houre: 60
your salary is 3000
mo:
```

ملحوظه : استخدام ال if ..else if استخداما اسرع من استخدام اكثر من if ورا بعض
لان ال if...else if لو شرط واحد اتحقق مش هيتشك علي الباقي
معلومه advance : لو انت عارف احتماليه الاحداث الي هي هتعملها في ال if..else if
تبدء بالي احتمال حدوثها اكبر ثم الأقل ثم الأقل .
ويبقى ال syntax النهائي لل if :

If(condition1)

{

Statements;

}

Else If(condition2)

{

Statements;

}

Else

{

Statements;

}

NOTES:

- In if ...else , else is optional.
- No code is available between the curly brackets and if or else or else if.
- Nesting is allowed , recommendation (no more than 3 nested if): for readability .

- If there is more than one condition in if stamen use round brackets () to make sure of the sequence of checked conditions.
- {} may be not be used if only one statement is required.

Switch

Syntax:

```
switch (<variable>)
{
case <case1>: statment1
break;
case <case2>: statment2
break;

default : statment3
break;

}
```

NOTES:

ال const مقصود بيه الحاجه الي بيتيجي بعد ال case

- Const must be unique
- Const can not be variable
- Only be decimal value

- Only one default is allowed(is also optional)
- Default can be any where in switch brackets

ال switch case يشتغل زي ال lockup table وده بيدليه سرعه عاليه جدا مقارنة بال if..else

يعني ليه بردو:

ال if انا بتتشك علي الكوندشن الاولاني ثم الكوندشن الثاني ثم الثالث وهكذا لحد ما أوصل للكوندشن بيبقي بTrue فكل ده وقت محسوب

بينما ال Switch بتروح تتتشك علي الكوندشن المطابق يعني علشان اروح ل case 1 هياخد نفس الوقت علشان اروح ل Case 9 بينما كان في ال if علشان اروح لكوندشن 9 كنت باخد وقت = كوندشن 1+2+3+4+5+6+7+8+9

امتي الاتنين بيتساو لما ال if بيبقي اول كوندشن فيه بTrue

وعيبها انها بتاخذ مساحه اكبر من الميموري

لو هتعمل برنامج بيعمل اكثر من اتنين equality == استخدم ال switch case