

C Basics

GCC → GNO C Compiler

Tool chain

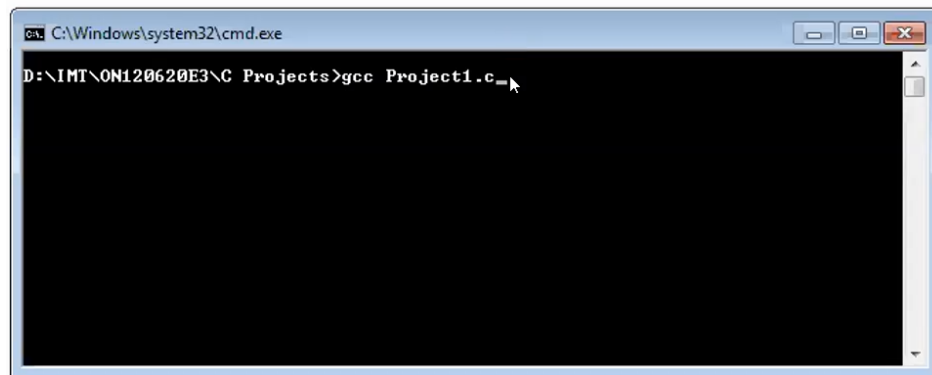
يعني ايه الكلام ده :

يعني بيعمل عملية البلدينج الي هي بيحول الكود ل

Exe file

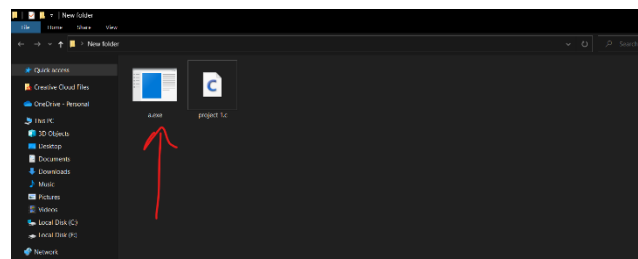
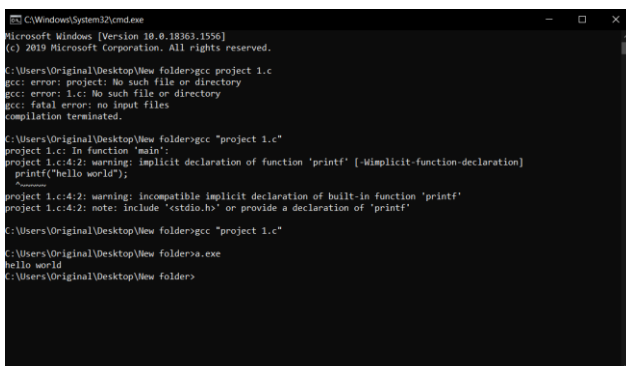
- C is structure programming language.
- Building block for C is {Function}.
- Entry point is → main function.
- TO run c file with cmd:

نستخدم الcmd في نفس الملف الي فيه الc file وبعد كده نستخدم الامر

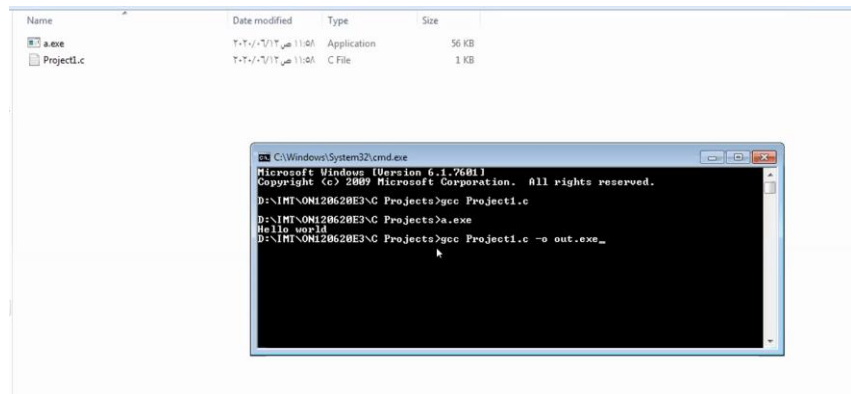


كده تم انشاء الملف ال exe

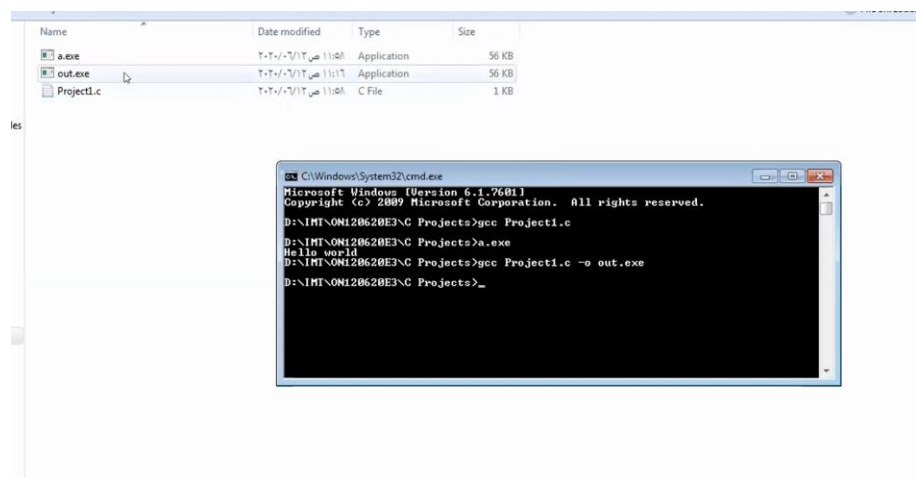
لتشغيله عليك بكتابه اسمه في الcmd



في العادي الملف بيطلع باسم a.exe لكن لو عايز تغيره:

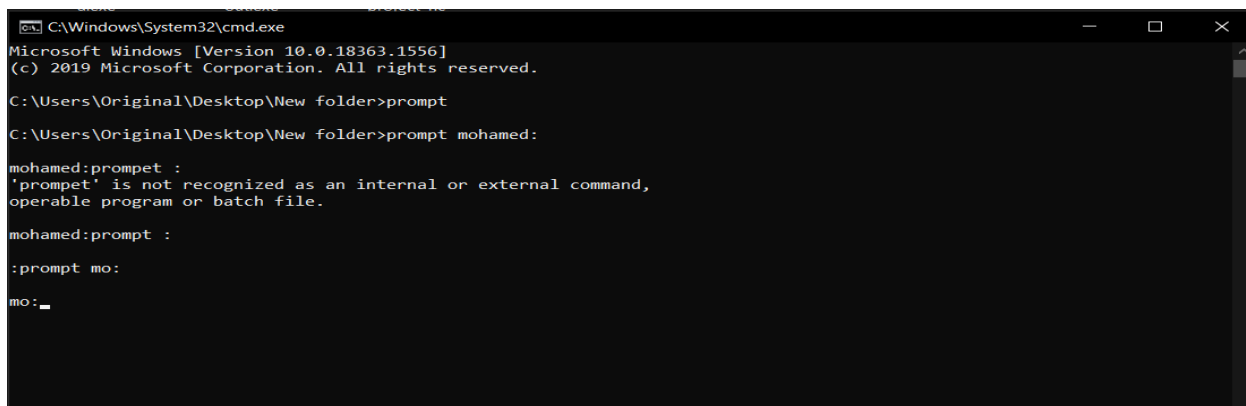


هياظهر عندك ملف جديد بالاسم الجديد

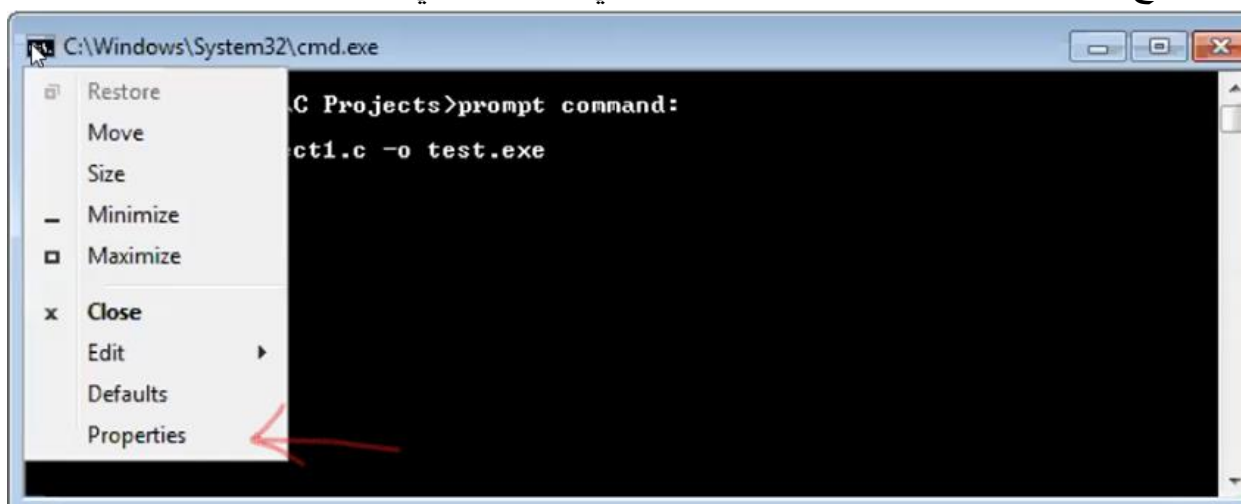


أوامر للCMD:

1. CLS: لومحتاج تشيل كل الوامر الي كتبتها من الشاشة
2. Prompt: لو المسار الي مكتوب مضايك وعازي تشيله بس ممكن تكتب بدل المسار الكلام الي انت عايزه



3. لو محتاج تعدل الخط او لونه او لون الخلفيه هيبقي Rclick علي ايكونه الcmd



الكومنت بيتم باستخدام ←

Comments

- 1- // → single line comment
- 2- /* comment */ → Multiline comment

ال"\" هي الرمز او الاوبريتور الي بنستخدمه في حاله الرموز المميزه مثل:

\n → لاضافه سطر جديد

\t → لاخت مسافه كبيره تقريبا 5 مسافات (بتفرق من تول اتشين للتانيه)

\\ → لاضافه \ واحده في النص

\' → لاضافه كوت واحده وبالمثل الدابل كوت

ال variables هي عباره عن عنوان بيشاور علي مكان في اليموري وبستخدمه في أداء العمليات الحسابيه والتعامل مع ال functions

ال Data type

عندنا 3 أنواع

- ال **primitive datatype**: وده الي جاي متعرف في اللغة زي {int,char,double,float,void}
- ال **Drived datatype**: زي {function,array,pointer} ودي أنواع مشتقة من الأنواع المعرفه في اللغة وبتيجي متعرفه أيضا
- ال **user define**: ودي أنواع بيعرفها اليوزر من خلال مثلا {struct,union,enum}

ال Primitive datatype

- ال interger عادات (لانه ممكن يتغير) بياخد 4 بايت (8 بت)
- ال char عادات بياخد 1 بايت
- ال float عادات بياخد 4 بايت
- ال double عادات بياخد 8 بايت

Syntax:

- **type name ;**
ex: `int x ;`
`x = 20 ;`
- **type name = Value ;**
ex: `int x = 20 ;`

لو حطيت مثلا

```
int x=30.25;
```

هتكون القيمه بتاعه ال `x = 30` بس لان `int` مسؤول بالجزء الصحيح من الرقم

علشان نستخدم الجزء العشري عندي نوعين float ,double

Naming rules:

1. NO spaces
2. NO special character except _
3. NO reserved words like {int ,double void }
4. 1st character is alphabet

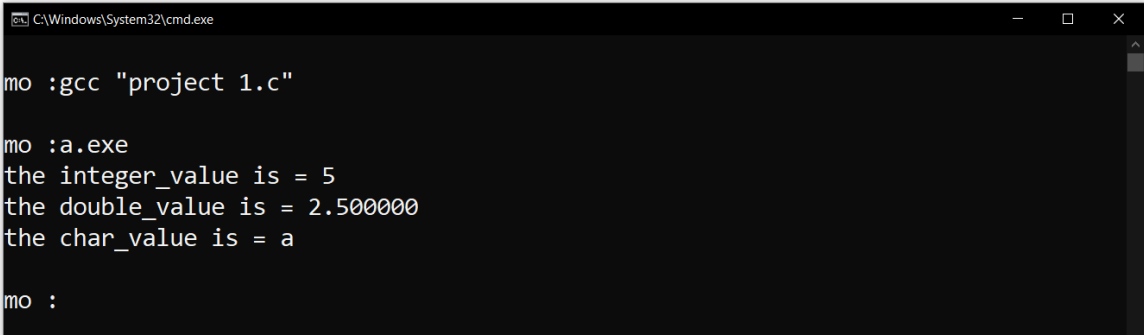
جمله الطباعه printf(); يمكن اتخدامه من مكتبة stdio.h

من خلال #include <stdio.h>

لطباعة قيم المتغيرات وليس اسماءها نستخدم = format specifier :

- %d → for decimal values
- %f → for floating values
- %c → for character values

```
#include <stdio.h>
int main()
{
    int integer_value= 5;
    double double_value=2.5;
    char char_value='a';
    printf("the integer_value is = %d\n",integer_value);
    printf("the double_value is = %f\n",double_value);
    printf("the char_value is = %c\n",char_value);
}
```



```
mo :gcc "project 1.c"

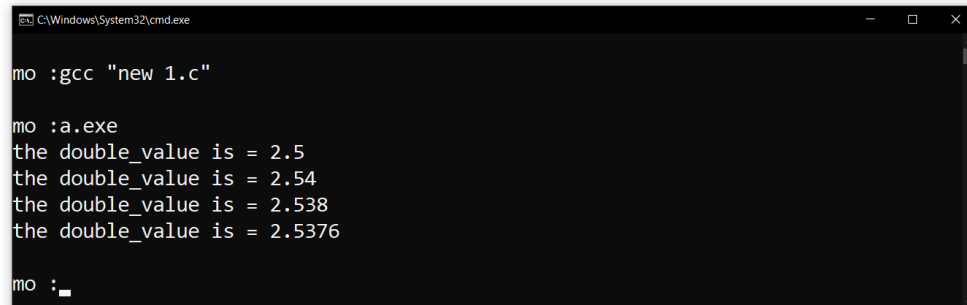
mo :a.exe
the integer_value is = 5
the double_value is = 2.500000
the char_value is = a

mo :
```

في حال اردت استخدام علامه عشريه واحده بدلا من 2.5000000 تريد انت تظهر علامه عشريه واحده 2.5

فأستخدم %.1f لتحديد عدد العلامات العشرية او %.2f لعلامتان عشريتان

```
1 #include <stdio.h>
2 int main()
3 {
4     double double_value=2.5376;
5     printf("the double_value is = %.1f\n",double_value);
6     printf("the double_value is = %.2f\n",double_value);
7     printf("the double_value is = %.3f\n",double_value);
8     printf("the double_value is = %.4f\n",double_value);
9 }
```



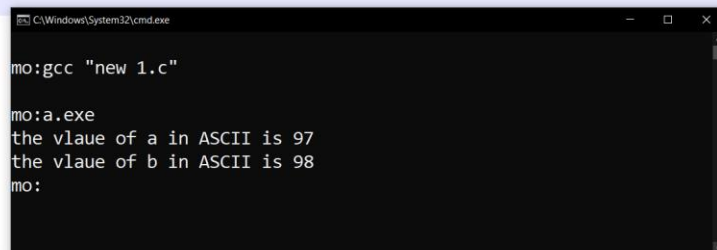
```
C:\Windows\System32\cmd.exe
mo :gcc "new 1.c"

mo :a.exe
the double_value is = 2.5
the double_value is = 2.54
the double_value is = 2.538
the double_value is = 2.5376

mo :_
```

طبعا كل حرف ليه تمثيل decimal للحرف ده في ال ASCII ممكن تظهرها باستخدام
الprintf();

```
1 #include <stdio.h>
2 int main()
3 {
4     char char1_value='a';
5     char char2_value='b';
6     printf("the vlaue of %c in ASCII is %d\nthe vlaue of %c in ASCII is %d",
7     char1_value,char1_value,char2_value,char2_value);
8 }
```



```
C:\Windows\System32\cmd.exe
mo: gcc "new 1.c"

mo: a.exe
the vlaue of a in ASCII is 97
the vlaue of b in ASCII is 98
mo:
```

To get input from user → **scanf();**

علشان نخزن متغير لازم نعرفه الأول قبل ما ناخده من اليوزير :

بنستخدم الأول ال format specifier

ثم بنستخدم ال address of operator ← & تفاصيل اكثر في جزء ال pointers

وبعد كده اسم المتغير الي هخزن فيه الداتا الي جايه من اليوزر

```
1 #include <stdio.h>
2 int main()
3 {
4     int x;
5     scanf("%d", &x);
6     printf("your input is = %d", x);
7 }
```



```
C:\Windows\System32\cmd.exe
mo: gcc "new 1.c"
mo: a.exe
5
your input is = 5
mo: _
```

Operators

Arthmitic operator:

1-binary:

+ → additional

- → subtraction

* → multiplication

/ → devision

% → moudlus

ناخد بالنا :

لو القسمه نتيجتيها ب float وانت بتخذيها في int المتغير هيشيل الجزء العشري

المودولوس هو الرقم الباقي من القسمة (مثل الكسر)
 مثال 2/9 سيكون الناتج فيها الـ 4 والباقي 1 الواحد ده هو ناتج عملية الـ modlus والناتج بيبقي integer

$$\text{int result} = 6 \% 4 \rightarrow 6 - 1 * 4 = 2$$

$$\text{int result} = 5 \% 10 \rightarrow 5$$

$$\text{int result} = 11 \% 12 \rightarrow 11$$

$$11 - 0 * 12 = 11$$

$$\text{int result} = 12 \% 11 \rightarrow 1$$

في قاعده عامه للريميندر

general rule

$$\text{dec}\left(\frac{A}{B}\right) * B + A \% B = A$$

مثال

$$\begin{array}{c} A \downarrow \quad \quad \downarrow B \\ 5 \% 3 : \end{array} \quad \text{dec}\left(\frac{5}{3}\right) * 3 + 5 \% 3 = 5$$

$$\underbrace{1 * 3}_3 + \underbrace{5 \% 3}_2 = 5$$

ملحوظه $\text{dec}(A/B)$ معناها ان ده الجزء الصحيح الديسيمل منه

مثال اخر

get -5 % -3

$1 * -3 = -3$

$-5 \% -3 = -2$

نلاحظ ان الناتج ديما بياخد اشاره ال A

2-Unary operator

++ → increment

-- → decrement

علي حسب مكان ال unary operator هيتغير سيكونس التنفيذ:
مثال

int x = 20 ;

int y;

y = x++; → post increment

// هنا هينفذ ال assingment قبل عمليه ال increment
الناتج هيكون X=21 , y=20

y = ++x; → pre increment

// هنا هينفذ ال assingment بعد عمليه ال increment
الناتج هيكون X=21 , y=21

ملحوظه ال ++x افضل في حجم الميموري واسرع في التنفيذ من ال x=x+1

لانه في ال x=x+1 بتتنفذ في 3 خطوات

Load X → modifay X (هجمع) → store X

ال inc انستراكشن في الاسمبلي بتعملها في خطوه واحده (بيعدله مكانه في الميموري علطول)

