

① For (unsigned char i = 5; i < 10; i++)

في الأول الـ 5 وهو يكون مصفوفة الشرط وحققت

تقل لحد ما توصل لـ (Zero) بس بعد كذا بسبب

أن الـ i من نوع unsigned char فهو يكون (0-255)

بـ 255 وده مش مصفوفة الشرط وبالتالي

الـ output ← 543210

(note)

في حالة الـ switch case

لو إحنا كتبنا case وهو بيخاف بيخافه

بس إحنا ممكن نأش في آخرها break خالص

الكود هيكمل التنفيذ في الـ case الـ بعدا لحد

ما يقابل أول break توقفه

(note)

#include <stdio.h>

int i;

int main()

{

int u;

printf("%d %d", i, u);

}

→ output 0 garbage

الـ هنا ← global

الـ هنا ← local

الـ Global Variable بيأخذ قيمة ابتدائية

مصغر لو إحنا بيطبقوش قيمة

الـ local variable بيأخذ garbage

لو إحنا مخطئش قيمة لـ

(note)

int i = 1;

if (i);

else

printf("else");

هنا شرط if ← true بس هو خطأ

بدها دامعناه إنه مفيش الـ body

الـ if ده يندى if و else ويكمل الكود عادي

ومش هيبقى في أي دور هنا بس

الـ if هيبقى لو كتبنا أي سطر كود

بعد سطر الـ if وقبل الـ else هنا هيطوع أي دور

وهيقول وجود else بدون وجود if

(note)

int i = 1024;

في الـ For هنا منافذ

القيمة الـ i في البداية (1) $i \geq 1$ \leftarrow For(; i, i >>= 1) \leftarrow Condition
 printf("GeeksQuiz");

هنا في البداية قيمة الـ i هي 1

How many times it Print? shift right

لا i بمقدار 1

1) i = 1024 \rightarrow True \rightarrow V1024 \rightarrow [0100 0000 0000 >> 1] = 0010 0000 0000 (512)2) i = 512 \rightarrow True \rightarrow V(512 >> 1) \rightarrow 0001 0000 0000 (256)

3

وهكذا

1024 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 0

1 2 3 4 5 6 7 8 9 10 11

∴ printf Print = 11 times #

(note)

if (printf("one"))

هنا الـ printf يترجع عدد الحروف التي هي مطبوعا، لأنه هنا

هنا مطبوع الـ one

IF(3) + one printed

if (printf("")) \rightarrow if(0)

كنا نحتاج مطبوع ولا حتى مسافة

(note)

int c = 5, no = 10;

do {

no /= c;

} while (c--);

printf("%d", no);

no = $\frac{10}{5} = 2$ no = $\frac{2}{4} = 0$

وهكذا

no = $\frac{0}{0}$

∴ runtime error

Runtime error

هنا في الأول يكون


```

int i = 0;
for ( printf("1st\n"); i < 2 && printf("2nd\n"); ++i && printf("3rd\n") )
{
    printf(" *\n");
}

```

أول مرة من ال loop

صطبع ال initially التي فيها 1st و بعدين يروح

يشوف ال condition

3 → ✓ $(0 < 2) \&\& [printf("2nd\n");]$

الشرط ← True فتقوم طابع ال 2nd الأول و بعدين

يروح يطبع ال * التي في ال body و بعدين يروح

ال action في الأول هي زود ال i ب 1 $i = i + 1$

يعني ال i هتبقى بدل صفر ب 1 و بعدين

هيطبع 3rd

Output (1)

```

1st
2nd
*
3rd

```

ثاني مرة من ال loop

مش هيروح جب ال initially و بالتالي ال 1st مش هتطبع

ثاني و هيروح على ال condition

1 → ✓ $(1 < 2) \&\& [printf("2nd\n");]$

الشرط ← True فتطبع 2nd الأول و بعدين

يروح ينفذ ال body و يطبع * و بعدين يروح ال action

فهناك هي زود ال i ب 1 فتبقى ب (2) و هيطبع 3rd

Output (2)

```

2nd
*
3rd

```

ثالث مرة من ال loop

هيروح لل condition على طول ← $(2 < 2)$

أول طرف في ال && ب zero و يبقى مش هيكمل

ال condition و يخرج على طول من ال loop

Output → Output (1) + Output (2)

#

(note)

int i = 3;

output →

Nothing gets Printed

switch(i)

{

printf("outside");

case 1: printf("Geeks");

break;

default: printf("GeeksQuiz");

}

(note)

about Switch

Float x = 1.1;

switch(x)

{

// body

}

int x = 2;

int arr[] = {1, 2, 3};

switch(x)

{

case arr[0]: printf("choise 1\n");

case arr[1]: printf("choise 2\n");

case arr[2]: printf("choise 3\n");

}

case label أو قيمة متغيرة في ال Var

Case label must be reduce to an integer constant

note في السطر الثامن في (0) بعد ال if دا
 معنا ه بان جملة ال if مش هتأخر ولا هتكونا
 ليها لاهدا فهتخرج بشئ اللى بعد هادى
 فهتوط (x=0) وبعدين هتدخل ال if الثانية
 فالشرط بتاكد هتبقى False فهتخرج
 زحوق ال else فهتطلع عننا فى الآخر بان
`int x=3;`
`if (x==2); x=0;`
`if (x==3) x++;`
`else x+=2;`
`x = -2`

note `int i=0;`
`for (; i<=5; i++)` → **Null Statement**
`printf("%d", i);`
 ال for هتعمل سؤال لحد ما الشرط مبيقاش true
 واما يوصل لحد ما هتبقى ال i يقى ب [] فلما تنجى
 تم تطبع ال i هتلاقى بانها ب []
`OUTPUT → 6`

`int a=500, b=100, c;` `!a = !500 = zero`
`if (!a >= 400)` `(0 >= 400) → false`
`b=300;` ببقى body ال if مش هتفعل
`c=200;` وال body هنا هو سطر واحد
 وال c هتفعل عادى
`b → 100 c → 200`

`unsigned int i = 65535` /* `int → 2byte` */
`while (i++ != 0)` `unsigned int (2byte) range`
`printf("%d", ++i);` `From 0 → 65535`
 هتدخل جوا ال while ياخذ قيمة ال i فى الأول بيشترى بيها الشرط وبعدين بيشترى
 يزود `(65535 != 0)` بكون هتزيد هادى ب (1) فهتبقى ب (0)
 يدخل ب print فهتزيد ال i الاول ده خالص ب (1) وبعدين يلحق
 وىد كذا بيشترى الشرط خال (0) كمر هتبقى ب صفر فهتصل
infinite loop

note Short int i = 0;

For (i <= 5 && i >= -1; ++i; i > 5)

printf("%u", i);

في المثال ده ال compiler قسّم ال condition على جزأين

condition ال action ال

اللى هو (++i) وهيدأ يطبع من أول (1) وهينقل

يزود كل مرة بـ 1 (65535) فلما

يزود (6) هتبقى (6) نال condition بيقى False

فهتوقف ال loop (65535)

Output → 1, 65535

important note

ex → Float a = 0.7;

double b = 0.7;

if (b > a)

printf("hi");

الشرط بتاي if هتاي True

يعنى هيطبع عندنا (hi)

ليه بيقى (hi) ؟

Float بيتخزن 4 byte

double بيتخزن 8 byte

عدد ال bits المسئولة عن ال Floating point في حالة ال double أكبر

دقة من الموجودة في حالة ال Float وبالتالي ال a مش هتساوي

ال double هو ال ب أكبر من ال a

ex → Float a = 0.7;

if (0.7 > a)

double ←

printf("Hi");

else

printf("Hello");

← الرقم لما بيتكتب لو عدد (بتكون أى A)

متغير لو هو real num

فهو بيتخزن default ك double

double > Float

شرط ال if ب True

Output → Hi

قند الـ ← لو حظ (F) جسم (0.7) دامكنا صاiane
 همفزن الرقده ك Float مش double
 وبالتالي شرط ال if هيكون False
 فيقوم منه body ال else
 output → Hello

(note)

```
int a=5, b=10, c=1;
if (a && b > c)
    printf("c question bank");
else
    break;
```

Condition if → 5 && (10 > 1) → True && True → True

← في هيفكون ان c question bank هو اللي صليطع

بس اللي صليطع compiler error و (:) و

عشاك في break جوا ال else وال break

لا تستخدم ال switch مع ال loop

output → compiler error

(note)

في حالة ال if ال compiler بيدخل

يشون شرط ال if لو true

هيفنق ال body بتاعها ويخرج

ولو مش true هيدخل يشوف ال

شرط ال else if اللي بيدها هيفنق

وبالتالي هيفنق شرط

ال if ب true هيفنق

(A) ويدخل يخرج

note `int x = -1, y = -1; if (++x == ++y) printf("true");`

else

`printf("false");`

Compiler error

`if (!printf("Mostafa"))`

`if (printf("Edrees"))`

في أول if الـ printf هتتبع عدد الحروف وتطبع كلمة Mostafa بس بسبب وجود (!) الشرط هيبقى False وبالتالي مش هينفذ السطر الثاني مكانه هو الـ body يتاخر الـ if وبالتالي الـ

Output → Mostafa

`int x = 1;`

`if (x--)`

`printf("Mostafa");`

`--x;`

else

`printf("%d", x);`

الـ body بتاخر الـ if في الحالة دي هتطبع Mostafa والـ x في النهاية هتكون بـ (-1)

Compound Statement → دالة عبارة عن مجموعة

من الأكواد مكتوبة بين braces ← `{ and }`

note `int a=5, b=10;`
`if (a < ++a || b < ++b)`
`printf("%d %d", a, b);`
`else`
`printf("False");`

في الـ `conditional` يتبع الـ `if` ← `Prefix` هو الذي يحصل الأول ويغيرين
 $(a < ++a \parallel b < ++b)$
 $(6 < 6 \parallel 11 < 11)$
 $(False \parallel False) \rightarrow False$
 الـ `else` يتبع الـ `body` يخرج تطبيق الـ
 Output → **False**

note `int x=1, y=2;`
`if (--x && --y)`
`printf("%d %d", x, y);`
`else`
`printf("%d %d", x, y);`

شرح الـ `if` ← `--x && --y`
 $(--x \&\& --y)$

من هيكمل للجزء الثاني `--y` و `--x` هتكون بـ (0)
 الـ `y` من هتكون جينا جيب قيمتها
 و هتكون لسه بـ (2)

Output → **0 2** ✓

```

int a = 2;
if (a--, --a, a)
    printf("Mostafa");
else
    printf("Anas");

```

شرط الـ if ← أول مرة هنا نأخذ الـ a ونكسرنا بنفسها

فان (a=1) بعد كذا فنفس الـ a الأول فونتهن

بـ (a=0) وفي الأخير هيقى if(a) الـ ص

if(0) يعني الشرط بـ False ولنا

هين الـ body تابع الـ else

output → (Anas) #

```

int i, j;

```

```

i = j = 3, 4;

```

هنا قيمة الـ 4 هتوصل ←

```

while (--i && j++)

```

ignore يمكن إيتامش

```

    printf("%d %d", i, j);

```

هنا ضما في اعتبرنا

(i=3, j=4) ←

1st → --i && j++ بعد كذا يزداد الـ j

(2 && 3) → True → 2 4

2nd → --i && j++

(1 && 4) → True → 1 5

3rd → --i && j++

(0 && j++) → False

output → 2 4 1 5

هناك الأسريج الـ **Prefix** و الـ **PostFix** ؟

1) **PostFix** $y = x++$ (10 9 8 7 6 5 4 3 2 1)

دري هتتقول لـ **3 instructions** بلغة الـ **assembly**

1- بيخزن القيمة بتاعت الـ x في الـ y

2- بيخزن القيمة بتاعت الـ x بمقدار (1)

3- بيخزن قيمة الـ x الجديدة في الـ **memory**
 فكلما في السطر التالي بعد كذا

2) **PreFix** $y = ++x$

دري هتتقول لـ **2 instructions** بلغة الـ **assembly**

1- هنزود قيمة الـ (x) بمقدار (1) وفي نفس الخطوة هنخلي قيمة

الـ x في الـ **memory** بالقيمة القديمة + 1

2- هنخزن القيمة بتاعت الـ x في الـ y

وبالتالي ← الـ **Prefix** أسريج من الـ **PostFix**

int n;

for (n=9; n!=0; n--)

printf("n= %d", n--);

$n=9 \rightarrow 9 \neq 0 \rightarrow \text{True}$

print $\rightarrow 9 \rightarrow n=8$ action $\rightarrow n=7$

print $\rightarrow 7 \rightarrow n=6$ action $\rightarrow n=5$

print $\rightarrow 5 \rightarrow n=4$ action $\rightarrow n=3$

Print $\rightarrow 3 \rightarrow n=2$ action $\rightarrow n=1$

Print $\rightarrow 1 \rightarrow n=0$ action $\rightarrow n=-1$

كل مرة تحقق الشرط $n \neq 0$

infinite loop وبالتالي هيدخل في

int i=40 → global id

extern int i;

int main() {

do {

printf("%d", i++);

} while (5, 4, 3, 2, 1, 0);

return 0;

}

while condition التي في الـ while

while (5, 4, 3, 2, 1, 0) → while (0)

في الشرط ← False يعني يخرج من الـ loop

Output → 40

طوبو سئلنا أول سطر في الكود الى فوق واية الى هيحصل (؟)

يعني سئلنا جزء الـ global variable ؟

هيحصل عننا linking error ليه (؟)

كشان الكود عننا مكتوب في ملف واحد بس فيه معناه

لنا المتغير (i) مش معمول له definition في ملف تاني

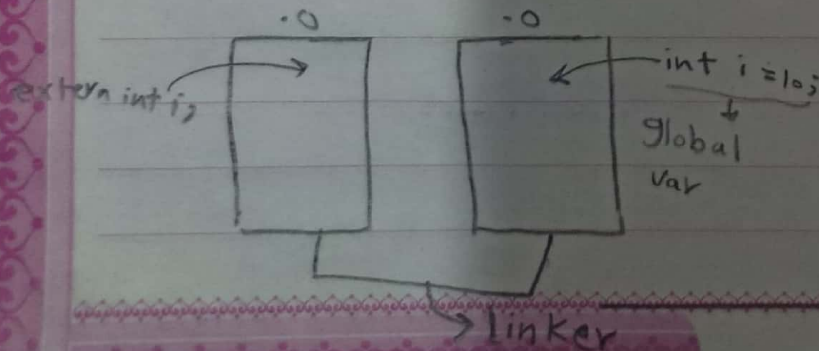
بس هو معمول له declaration عننا هنا في الملف

وبالتالي مفيش قيمة للمتغير (i) نقدر نجيبها

من ملف تاني فيحصل linker error

طب ليه مش هيحصل compilation error كشان هنا

هنا كاتبين كود C مع مفيش مشاكل



الـ linker error هيظهر على

شكل ld error


```
int i=2, j=2;
while(i+1? --i: j++)
    printf("%d", i);
```

Condition 1: $i+1? --i: j++$

1st $\rightarrow i=2 \rightarrow 3? --i: j++$
 True $\rightarrow --i \rightarrow i=1$ Condition $\rightarrow T$

Output $\rightarrow 1$

2nd $\rightarrow i=1 \rightarrow 2? --i: j++$
 True $\rightarrow --i \rightarrow i=0$ Condition $\rightarrow F$

يبقى ال loop مش هتتق وهتخرج برامها

والنتيجه ال Output $\rightarrow 1$