كلية العلوم

القسم : الرياضيات

السنة : الاولى

المادة : لغات البرمجة1

المحاضرة : الاولى/عملي/

تنزيل الانسة

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

الجمهورية العربية السورية

جامعة طرطوس

كلية العلوم قسم الرياضيات

السنة الاولى

المادة: لغات البرمجة 1 – عملي

المحاضرة الأولى

التعريف بلغة **C++**

2026

لمحة تاريخية:

ظهرت لغة C++ عام 1979 في مختبرات شركة بل، قام بتطويرها Bjarne Stroustrup كاستمرار للغة الام C وهي لغة عالية المستوى، تتميز C++ بانها لغة كائنية التوجه وبذلك يمكن كتابة البرامج وصيانتها بشكل أسهل، كما إنها تستخدم على نطاق واسع في برمجة نظم التشغل وبرامج التعامل مع العتاد الصلب Hardware بالاضافة الى برمجة برامج الحاسب.

مكونات اللغة:

رموز اللغة: الرموز المستخدمة في لغة C++:

1. الحروف الإنجليزية الكبيرة A.B.C
2. الحروف الإنجليزية الصغيرة a.b.c
3. الأرقام العربية الأصل 1.2.3
4. رموز خاصة مثل:

[]	"	!	<	-	+
*	،	||	>	( )	_
>>	<>	<=	>=	\	/
!=	&	%	$	#	<<

الجدول١-١

ملاحظة: لغة C++ حساسة لحالة الأحرف اي هناك فرق بين الحروف الأبجدية الصغيرة والكبيرة.

الأسماء التعريفية: وتطلق الأسماء التعريفية على المتغيرات، الدوال والمؤشرات.

قواعد تسمية الأسماء التعريفية في لغة C++:

1. اسم التعريفي لا يبدأ برقم.
2. اسم التعريفي لا يحوي رموز خاصة.
3. لا يبدأ بتسطيره غير سفلية .
4. لا يحوي على فراغ.
5. لا يجب ان يكون كلمة محجوزة.

بعض الأمثلة الصحيحة على الأسماء التعريفية:

Matrix ، _new ، TotalValue , Sum3_6 , X , Z315 , _RecorD

بعض الأمثلة غير الصحيحة على الأسماء التعريفية:

5new: يبدأ برقم.

math-: اسم يبدأ بتسطيرة غير سفلية.

Value%: اسم يحوي الرمز الخاص (%).

Red d: يحتوي على فراغ.

do, char,new: أسماء خاطئة لأنها كلمات محجوزة.

Var!: لأنه يحوي الرمز الخاص (!).

Od.d: لأنه يحوي الرمز الخاص (.).

الكلمات المحجوزة:

وهي كلمات قياسية معروفة مسبقاً لمترجم C++ وتكتب عادة بحروف صغيرة، ولها معان خاصة بها تؤديها في البرنامج.

الكلمات المحجوزة في لغة++C:

near	static	asm	double	long	sizeof
do	int	while	new	auto	else
for	this	void	delete	goto	if
const	entry	char	class	public	case
continue	extern	struct	inline	float	private
virtual	volatile	frinde	enum	return	static
cdecl	default	register	overload	unsigned	typedef
signed	pascal	operator	switch	template	union
protected	far	Catch	break		

الجدول١-٢

أنواع البيانات الأساسية في لغة ++C:

هناك عدة أنواع من متغيرات البيانات في لغة C++ منها يمثل المحارف ، الأعداد الصحيحة، الأعداد الحقيقية.

المحارف char:

يتم تخزين المحرف في متغيرات من النوع char، يحجز 8 bits من حجم الذاكرة مجال القيم [-128,127].

الأعداد الصحيحة:

هنالك ثلاثة أنواع من متغيرات الأعداد الصحيحة في لغة C++ هي: قصير short و عدد صحيح int و طويل long.

والفرق بين الثوابت الطويلة والقصيرة هو في عدد الوحدات التخزينية المطلوبة لكل نوع في الذاكرة ، فالطويلة تأخذ حيزاً اكبر، والقصيرة توفر عدد الوحدات التخزينية المستعملة ، أما الثوابت الصحيحة بدون إشارة int unsigned ، فان استعمالها يوفر وحدة تخزينية واحدة تستعمل للاشاره عندما تذكر كلمة unsigned قبل int ،وذلك بإزاحة القيمة إلى قيمة موجبة بدون إشارة ، ولكل نوع من الأنواع السابقة تطبيقاته المناسبة.

short يحجز 8 bits من حجم الذاكرة مجال القيم [32767+ ,32768-]، **int** يحجز مثل short في الأنظمة

التي تعمل ب 16 bits ومثل long في الأنظمة التي تعمل ب 32 bits، **long** يحجز32 bits مجال القيم [-2147483648, +2147483647].

متغيرات الأعداد الصحيحة التي ليس لها إشارة:

المتغيرات التي ليس لها إشارة لا تستطيع تخزين قيم سالبة لكن حجم مجال قيمها الموجبة يساوي ضعف حجم المجال للتي لها إشارة، تكون الأعداد الصحيحة الاعتيادية بدون ميزة unsigned لها إشارة بشكل افتراضي.

short مجال القيم [0, 65535]، **int** يحجز مثل short في الأنظمة التي تعمل ب 16 bits ومثل long في الأنظمة التي تعمل ب 32 bits، **long** يحجز 32 bitsمجال القيم [0, 42944967295].

الأعداد الحقيقية:

تستخدم لتمثيل الأعداد العشرية والكسور و بدلاً من النقطة العشرية يمكن استعمال الأسي لذا القيمة 124.965 في الشكل العادي هي 1.2465e2 في الشكل الأسي حيث يشير الرقم الذي يلي الحرف e إلى كم مرة يجب نقل النقطة العشرية إلى اليمين لاسترجاع القيمة إلى الشكل العادي.

هنالك ثلاثة أنواع من متغيرات الأعداد الحقيقية في لغة C++ هي:

- float: الحجم 32 bits مجال القيم [10e-38,10e38]، الدقة 5 عدد.
- double: الحجم 64 bits مجال القيم [10e-308,10e308]، الدقة 15 عدد.
- long double: الحجم 80 bits مجال القيم [10e-4932,10e4932]، الدقة 19 عدد.

يتطلب النوع float ذاكرة أقل من النوع double ويعد أسرع في إنجاز العمليات الحسابية، يستعمل النوع ى long double في معالج الأرقام الكبيرة.

المتغيرات:

المتغيرات هي عبارة عن مواقع في الذاكرة لها حجم معين (كل نوع له حجم معين) تقوم بتخزين البيانات، تعد من أساسيات البرمجة ولايمكن لبرنامج أن يعمل من دونها، تتكون من عدة انواع لكل نوع حجم مخصص له في الذاكرة، مثل متغيرات من النوع الصحيح int ومتغيرات من النوع النصي string char الخ.

وتنقسم المتغيرات إلى:

1. متغيرات عددية : وهي مواقع في الذاكرة تخزن بها أعداد .
2. متغيرات رمزية : وهي مواقع في الذاكرة تخزن بها رموز (محارف ونصوص).
3. متغيرات منطقية : وتخزن بها قيمة منطقية أما TRUE= 1 FALSE = 0 or

الشكل العام لتعريف المتغيرات:

```
Type Name = Value;
```

Type:وهو نوع المتغير اما عدد (int,double,..) او نص(string,char.. الخ)

Name:اسم المتغير ويجب ان يراعي شروط كتابة الأسماء التعريفية.

value: قيمة المتغير.

```
int n;
```

اسم المتغير و n هو هذا المتغير من نوع الأعداد الصحيحة ، كل متغير يجب أن يكون له نوع.

و يمكن أن نعلن عن أكثر من متغير من ذات النوع كما يلي:

```
int x,y,z;
```

ملاحظة: كل أمر يجب أن ينتهي بفاصلة منقوطة (;)، يتم الفصل بين المتغيرات من نفس النوع المعلن عنها بفاصلة (,)، يمكن أن يكون الإعلان في أي مكان بالبرنامج و لا يمكن استخدام متغير قبل الإعلان عنه.

عمليات الإسناد والتخصيص:

تتم باستخدام الرمز (=) ويمكن أن تسند قيمة ابتدائية للمتغير عند الإعلان عنه، وتأخذ الشكل الآتي:

```
variable = Value;
```

value هي القيمة المسندة للمتغير ويمكن أن تأخذ أحد الأشكال:

1. قيمة ما:

```
int x;
x=66;
```

2. اسم متغير يحمل قيمة:

```
int x,y;
x=8;
y=x;
```

3. تعبير حسابي:

```
int x,y,z;
int x=5;
int y=7;
int z= x*y+2;
```

4. دالة رياضية:

```
int x=9, y;
```

y=abs(x);

abs: هي دالة رياضية لإيجاد القيمة المطلقة.

إعطاء قيم ابتدائية لأكثر من متغير عند الإعلان عنهم:

int x,y=5,z=6,a,b,m=4;

أعلن عن المتغيرات x,a,bعلى أنها أعداد صحيحة ولكن المتغيرات y,z,m تم إسناد قيم لهم.

مثال لتخزين متغير محرفي:

char aa_ch;

aa_ch='a';

الأحرف الثابتة مثل 'a', 'b', '$','55' يجب أن تكون محصورة بين علامتي اقتباس فردية.

يمكن استعمال النتغيرات من النوع char لتخزين أرقام صحيحة بدلاً من الأحرف مثل:

char c=65;

عند طباعة المتغير c يظهر الرمز A لأن الرقم 65 يمثل الرمز A في جدول الآسكي.

محارف الهروب:

هي عبارة عن رموز تستخدم للقيام بمهمة معينة.

محرف الهروب	العمل المنفذ
'\n'	تنقل المؤشر إلى بداية السطر التالي
' \\ '	إدراج شرطة مائلة
'\b'	تنقل المشيرة إلى الوراء Backspace
'\r'	تنقل المشيرة إلى بداية السطر الحالي
' \' ,	إدراج اشارة تنصيص علوية
' \" '	إدراج إشارتي تنصيص علويتين
'\t'	تحريك المشيرة او رأس الطباعة لطباعة الأسماء التي تليها على شكل جدولة أفقية (بمقدار المفتاح Tab)
'\f'	للتقدم صفحة واحدة للأمام عند طباعة ملف
'\a'	إصدار صوت الجرس

محرف الهروب '\n' تنقل المؤشر إلى بداية السطر التالي (سطر جديد).

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<" Hello \n";
 cout<<" Hellen";
}
```

```
 Hello
 Hellen
[Program finished]
```

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<"Hello \nmy new line";
}
```

```
 Hello
 my new line
[Program finished]
```

محرف الهروب ’\t‘ تحريك المشيرة او رأس الطباعة لطباعة الأسماء التي تليها على شكل جدولة أفقية (بمقدار المفتاح Tab)

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<" Student \t ";
 cout<<" Raghad";
}
```

```
 Student         Raghad
[Program finished]
```

محرف الهروب ’\a‘ إصدار صوت الجرس (إنذار).

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<" Student\a";
 cout<<" Raghad";
}
```

```
 Student Raghad
[Program finished]
```

محرف الهروب ’\b‘ تنقل المشيرة إلى الوراء Backspace.

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<" Hello Worl\bd";
}
```

```
 Hello Word
[Program finished]
```

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<" Studenn\bts Clsss ";
}
```

```
 Students Clsss
[Program finished]
```

محرف الهروب ’\r‘ تنقل المشيرة إلى بداية السطر الحالي.

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<"ma in\rnbl";
}
```

```
nblin
[Program finished]
```

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<"Hello\rshe";
}
```

```
shelo
[Program finished]
```

محرف الهروب ' \\ ' إدراج شرطة مائلة.

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<"\\Ahmad";
}
```

```
\Ahmad
[Program finished]
```

محرف الهروب ' \ ' إدراج اشارة تنصيص علوية.

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<"\'Ali\' Lama";
}
```

```
'Ali' Lama
[Program finished]
```

محرف الهروب ' \" ' إدراج إشارتي تنصيص علويتين.

```cpp
#include<iostream>
using namespace std;
int main()
{
 cout<<"\"Ahmad\"";
}
```

```
"Ahmad"
[Program finished]
```

مثال على استخدام محارف الهروب:

cout: هو أحد الأوامر التي توفرها مكتبة iostream.h، يقوم بإخراج ما يأتي بعده ويظهره بعلى الشاشة.

>>: يسمى معامل الإخراج أي مانكتبه بعده سيقوم بإخراجه، كل مانريد إخراجه وعرضه على الشاشة يجب كتابته ضمن إشارتي تنصيص.

الشكل العام لعملية الطباعة:

```
cout<<’ hello world’;
```

حيث ستظهر عبارة hello world على الشاشة.

البرنامج:

```
#include<iostream>
using namespace std;
int main()
{
 cout<<"syrian arab republic ";
 cout<<"tartous uneversity ";
 cout<<"Math ";
 cout<<"\n .....\n";
 cout<<"syrian arab republic \n";
 cout<<"tartous uneversity \n";
 cout<<"Math \a";
 cout<<"\n .....\n";
 cout<<"syrian arab republic \r";
 cout<<"tartous uneversity \t";
 cout<<"Math\t\n";
 cout<<"Math\b";
 cout<<"Math \'s programming lecture";

}
```

الخرج:

```
syrian arab republic tartous uneversity Math
 ..................
syrian arab republic
tartous uneversity
Math
 ..................
tartous uneversity c    Math
MatMath 's programming lecture
[Program finished]
```

العمليات الحسابية:

العمليات الحسابية الثنائية:

تتضمن لغة C++ العمليات الحسابية الأربعة بالإضافة إلى عملية باقي القسمة.

الجمع (+) الطرح (-) الضرب (*) القسمة (/) باقي القسمة (%).

إن عملية باقي القسمة تقوم بحساب باقي قسمة عدد صحيح على عدد صحيح آخر، فمثلاً التعبير (9%2) يساوي 1.

العمليات الحسابية الفردية:

عملية الزيادة:

يرمز لها بالرمز ++ وهي عملية تنفذ زيادة بمقدار (1) على المتغير ولها نوعان:

زيادة سابقة: اي الزيادة على المتغير قبل التنفيذ:

```
int i ;
++i ;
```

مثال:

```
int x =4, y =2;
int z = ++x + y;
int c = x - y;
```

تم تعريف المتغيرين x,y وإسناد قيمة ابتدائية لهما.

يتم زيادة x بمقدار 1، ثم استخدام القيمة الجديدة. تصبح قيمة x هي: (4+1) و هي قيمة x الجديدة والمخزنة في الذاكرة.

z= 5 + 2 =7.

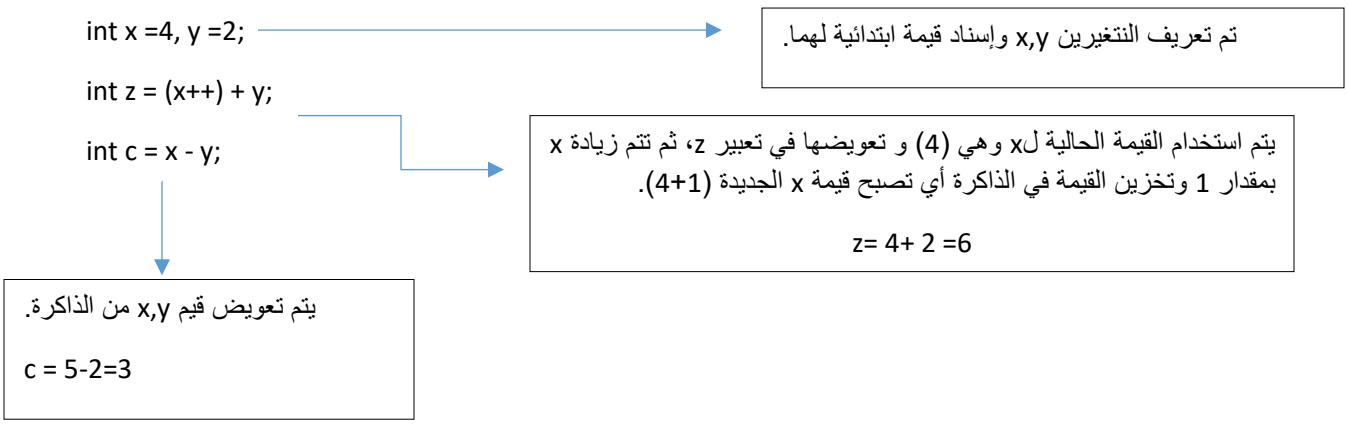
يتم تعويض قيم x,y من الذاكرة.

c = 5-2=3

زيادة لاحقة: اي الزيادة على المتغير بعد التنفيذ:

```
int i ;
i++ ;
```

مثال:



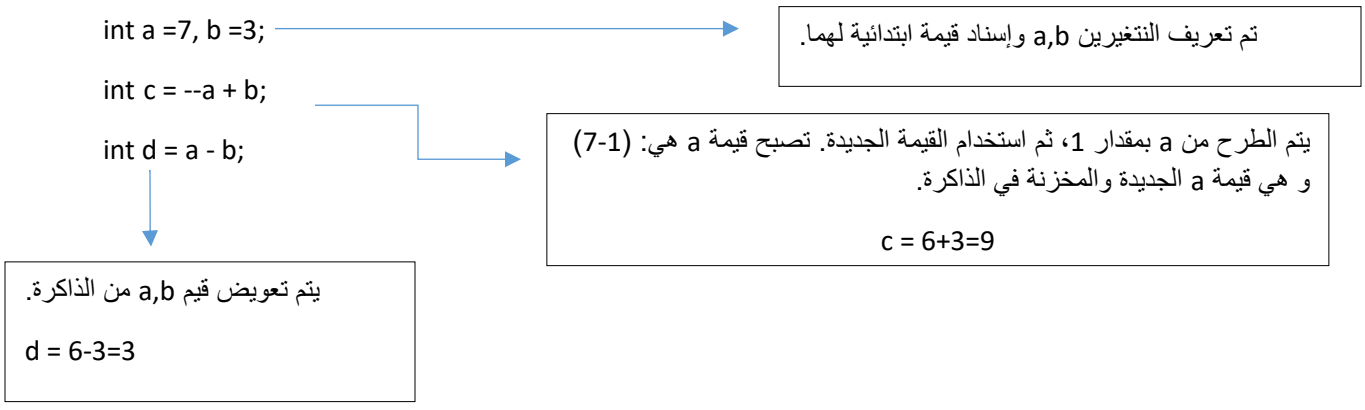
عملية النقصان:

يرمز لها بالرمز -- وهي عملية تنفذ طرح بمقدار (1) من المتغير ولها نوعان:

نقصان سابق: اي الطرح من المتغير قبل التنفيذ:

```
int i ;
--i ;
```

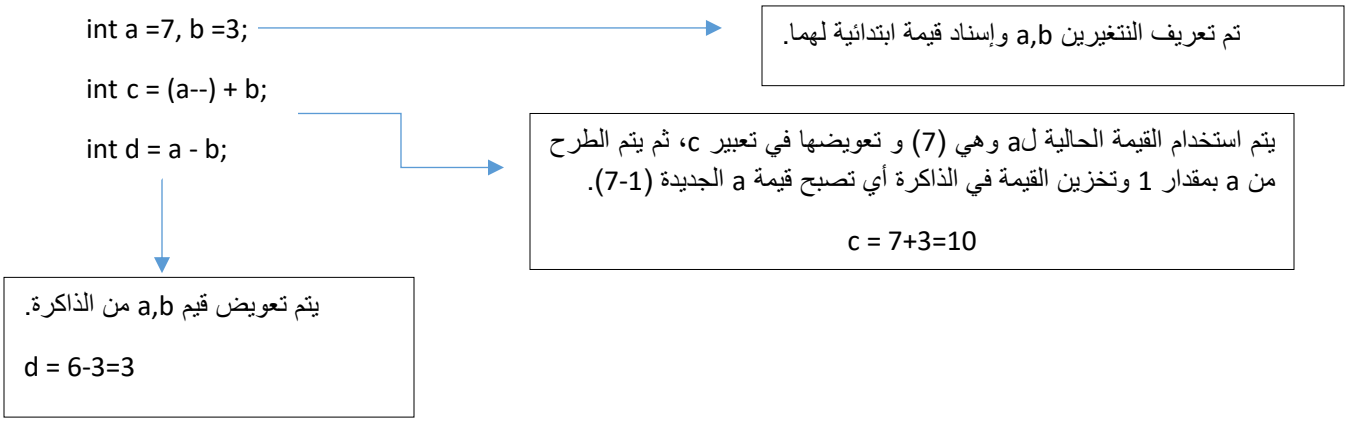
مثال:



نقصان لاحق: اي الطرح من المتغير بعد التنفيذ:

```
int i ;
i-- ;
```

مثال:



الأولويات في العمليات الحسابية:

1. الأقواس ().
2. العمليات الفردية الزيادة والنقصان (++، --).
3. الضرب، القسمة و باقي القسمة (*، /، %).
4. الجمع و الطرح (+،-).
5. الإسناد (=).

ملحوظة:

إذا تساوت أوليتان مثل الجمع والطرح في تعبير ، فتقدم العملية الأقرب إلى يسار التعبير ، وعند استعمال الأقواس لأي تعبير فان الأقواس تأخذ الأولوية الأولى في التنفيذ قبل (الزيادة أو النقصان) ، كما في لغات البرمجة الأخرى ، مثال:

X + y / z * a

يأخذ تسلسل أولويات عملياته الشكل والخطوات التالية:

العملية الأولى القسمة (y/z)، العملية الثانية هي الضرب ((a*(y/z))، العملية الثالثة هي الجمع ((X+(a*(y/z))).

4+10/5*2

العملية الأولى 10/5=2، العملية الثانية 2*2=4، العملية الثالثة 4+4=8، الناتج هو 8.

العمليات العلائقية

أصغر تماماً (<)، أكبر تماماً (>)، أصغر أو يساوي (=<)، أكبر أو يساوي (=>)، يساوي (==)، لا يساوي (!=).

نتيجة كل عملية هي إما true=1 أو false=0.

int a=b=3 ;

فان التعبير a < 3 نتيجته false أي 0.
التعبير a <= 3 نتيجته true أي 1.
التعبير a > b نتيجته false أي 0.
التعبير a != b نتيجته false أي 0.
التعبير a = = b نتيجته true أي 1.

العمليات المنطقية

تضمن لغة c++ ثلاث أدوات منطقية:

الأداة AND (&&) تكون العبارة صحيحة إذا كان التعبيران الموجودان على جانبي المعامل صحيحين.

X > 5&& X<=15 تكون العباة صحيحة إذا كان المتغيرضمن المجال [6,15].

الأداة OR (||) تكون العبارة صحيحة إذا كان أحد التعبيرين أو كلاهما صحيح.

X > 10 || X<13 تكون صحيحة إذا كان المتغير يقع خارج المجال [10,13].

الأداة NOT (!) يكون المعامل صحيحاً إذا كان التعبير بعده خطأ و يكون خطأ إذا كان التعبير صحيحاً.

```
int h =8, k=6;
int g =!(h <k);
```

إن العملية h < k خاطئة وبالتالي قيمتها تساوي 0، المعامل ! ستكون قيمته بعكس قيمة التعبير بعده وبالتالي سيكون صحيح وقيمته هي 1، أي قيمة g هي 1.

ترتيب أولويات العمليات الحسابية والمنطقية:

1. الأقواس.
2. الضرب القسمة، باقي القسمة.
3. الجمع والطرح.
4. العمليات العلائقية.
5. العمليات المنطقية.
6. الإسناد أو التخصيص.

أمثلة:

```
int a =3;
int b =a++;
```

قيمة b هي 3، تستخدم قيمة a ما هي ثم بعد ذلك تصبح a=4.

```
int x = 5;
int res =(x>3)&&(x<10);
```

x>3 =1

x<10=1

قيمة res هي 1، لأن الشرطين كلاهما صحيح.

```
int x = 3+4*2;
```

4*2=8

3+8 =11

قيمة x هي 11

```
int k =3, f=2;
int b =f+k*2>6;
```

k*2=6

f+6=8

8>6 =1(true)

قيمة b هي 1.

```
int a =4;
int res =(a++>4)&&(++a<7)
```

a++>4 تستخدم قيمة a كما هي: 4>4، عبارة خاطئة، وبالتالي عبارة and خاطئة. res =0.

```
int a=1, b =2, c=3;
int r = a+b>c&&b++;
```

a+b=3

3>c = 0 (false)

الشرط الأول من العملة AND هو خاطئ و بالتالي الشرط الثاني لن ينفذ، أي أن b++ لا تنفذ.

r=0