

Assignment

Home work II

EX 7

TAO 17

Interview trick

Swap Two numbers without temp Variable.

المطلوب ← معتبر تبديل القيمة المحفوظة في var 2 بدون استخدام

متغير 3 معتبر ثالث

الحل ← حلول وليس هنترج فكره حل بـ اضافة المباقي زرع
 ١) لا نستخدم عملية الجمع والطرح

ويزيلان بدل القيمة بتاعتهم →

) صنع جمع الاستثنى وحط قيمة الجمع في أول متغير منه

$$x = x + y - (x - y) \rightarrow x = 25 + 10 - 25 = 35$$

ب) نطرح المجموع من المتغير الثاني فهو يطلع قيمة المتغير الأول وتحطها في المتغير الثاني

$$y = x - y \quad y = 35 - 10 = 25$$

ج) نطرح المجموع من القيمة اللي ادنا لسه دطلبها في المتغير الثاني فتطلعلينا

القيمة الأصلية للمتغير الثاني وتحطها داخل المتغير الأول

$$x = x - y \quad x = 35 - 25 = 10$$

) دوكما تكون عملنا Swap لقيمة الـ var

$$(x = 10) \oplus (y = 25) \rightarrow (y = 25) \oplus (x = 10)$$

تلخيص N ٣ حلول

$$x = 25 \quad y = 10$$

$$x = x + y$$

$$y = x - y$$

$$x = x - y$$

$$x = 25 \quad y = 10$$

$$x = x * y$$

$$y = x / y$$

$$x = x / y$$

$$x = 25 \quad y = 10$$

$$x = x ^ y$$

$$y = x ^ y$$

$$x = x ^ y$$

$$\therefore x = 10$$

$$y = 25 \quad \#$$

$$x = 10 + 25 - 10 = 25$$

$$y = 25 \quad \#$$

$$y = 25 \quad \times$$

* C programming is → Case sensitive language

Float \neq FLOAT
 ↓
 keywording ↓ not in C

Float abc=1.5; int big=3;

→ Float g=big/2; g=3/2 $\frac{\text{int}}{\text{int}} = \text{int} \rightarrow 1$
 ↓
 g → float (g = 1.000.000)

→ Float g=big/2.0; g=3/2.0 $\frac{\text{int}}{\text{float}} \rightarrow \text{float} \rightarrow 1.5$
 ↓
 g → float (g = 1.500.000)

when → abc = 1.5 big = 3 g → float

$$\begin{aligned} \rightarrow g &= \text{big}/2 + \text{big}*4/\text{big} - \text{big} + \text{abc}/3 \\ &= 3/2 + 3*4/3 - 3 + 1.5/3 \\ &= 1 + 12/3 - 3 + 0.5 \\ &= 1 + 4 - 3 + 0.5 \quad \rightarrow \therefore g \rightarrow \text{float} \\ (g &= 2.5) \end{aligned}$$

when → ink = 3 act = 2 tig = 3/2 on → int

$$\begin{aligned} \rightarrow on &= \text{ink} * \text{act}/2 + 3/2 * \text{act} + 2 + \text{tig} \\ &= 3 * 2/2 + 3/2 * 2 + 2 + 3/2 \\ &= 6/2 + 1 * 2 + 2 + 3/2 \\ &= 3 + 2 + 2 + 3/2 \quad \rightarrow \therefore on \rightarrow \text{int} \\ (on &= 10) \end{aligned}$$

when → qui = 2 add = 4 god = 3 s → int

$$\rightarrow s = \text{qui} * \text{add}/4 - 6/2 + 2/3 * 6/\text{god}$$

$$(s = -1)$$

int i=2, j=3, k, l;

float a, b;

$$k = i \mid j * j$$

$$k = 2 / 3 * 3 \\ = 0 * 3$$

$$k = 0$$

$$l = j / i * i$$

$$l = 3 / 2 * 2 \\ = 1 * 2$$

$$l = 2$$

$$a = i / j * j - 2$$

$$a = 2 / 3 * 3 \\ = 0 * 3$$

$$a = 0.0$$

$$b = j / i * i$$

$$b = 3 / 2 * 2 \\ = 1 * 2$$

$$b = 2.0$$

The real Constant in C can be expressed in which of

the following forms \rightarrow

a) Fractional Form

b) Exponential Form

R.N. $\rightarrow 1.5$ FF

$\rightarrow 4e^3$ eF

assume that \rightarrow a is int with 2 bytes

$$a = 30 * 1000 + 2768$$

$$a = 30000 + 2768 = 32768$$

but range of $\rightarrow -32768 \rightarrow 32767$

- this number is out of range

لما يزيد عن عدد بتات المخزن أو يقل عن عدد

\leftarrow range \rightarrow عدد المخزن

$$+32768 \rightarrow 1000\ 0000\ 0000\ 0000$$

$$0111\ 1111\ 1111\ 1111$$

1's Complement

! +

$$a =$$

$$(-32768) \rightarrow 1000\ 0000\ 0000\ 0000$$

2's Complement

$$(2a = -32768) \#$$

Note

Float abc = 1.5

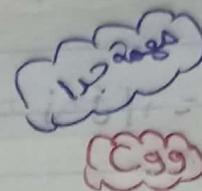
int big = 3

(int)

$$\left\{ \begin{array}{l} g_1 = 5 \% (-3) \rightarrow 2 \\ g_2 = (-5) \% (3) \rightarrow -2 \\ g_3 = (-5) \% (-3) \rightarrow -2 \end{array} \right.$$

(int)

$$\left\{ \begin{array}{l} g_1 = 5 / (-3) \rightarrow -1 \\ g_2 = (-5) / (3) \rightarrow -1 \\ g_3 = (-5) / (-3) \rightarrow -1 \end{array} \right.$$



الـ الرخاقيه ← remainder ← ميتاشرس بالسائلين إذا كان موجود في الباقى (N) إذا الـ division (/) يتكرر عارى بوجود السالب سود فى الباقى (N) أو العقام (D)

$$\text{ex} \Rightarrow x = 4 + 2 \% - 8$$

$$x = 4 + 2 = 6 \#$$

$$\text{ex} \Rightarrow x = 4 + 8 \% - 2$$

$$x = 4 - 4 = 0 \#$$

char → 1 byte

C = 125

$$C = C + 10$$

$$C = 125 + 10 = 135$$

$$\text{range of } C \Rightarrow -128 \rightarrow 127$$

this number is out of range ↗

$$135 \rightarrow 1000 \ 0111$$

$$0111 \ 1000 \quad 1^{\text{'}} \text{s Comp}$$

1 +

$$-121 \rightarrow 0111 \ 1001 \quad 2^{\text{'}} \text{s Comp}$$

$$-C = -121$$

```
unsigned int x = 1
```

```
int y = ~0
```

```
if(x == y)
```

```
printf("same");
```

```
else
```

```
printf("not same");
```

$x \rightarrow 0000\ 0000\ 0000\ 0001$

$y \rightarrow 1111\ 1111\ 1111\ 1110$

$(x, y) = True$

$-1 \rightarrow 1111\ 1111\ 1111\ 1111$

$= 0x FFFF \rightarrow -1$

على اختلاف الـ byte \leftarrow int

$y = ~0 = N(0000\ 0000\ 0000\ 0000)$

$y = (1111\ 1111\ 1111\ 1111) = 0x FFFF$

$\therefore x = y \quad \Leftarrow$

$28 = 9 + di (d)$

Same

طبع

note

char a = '1012'

ASCII with octa number

age

'101' \rightarrow Octa

0x \rightarrow hexa

هذا هو ما يتم تخزينه في الـ ASCII بـ 1012

بـ 3 اطلاعات غير الـ a

OCT (12) \rightarrow Dec (10)

printf("%d", a); \Rightarrow (a = 10) $\#$ + di (0)

$2 + 1 \ll 3$

$2 + 1 * 8$

$3 \ll 3$

$2 + 8$

$3 \rightarrow (3) * 2^3$

$= 10$

$= 3 * 8 = 24$

$3 \ll 3$

ex $1 \ll 2 + 3 \ll 4$

$0000\ 0011$

$= 1 \ll 5 \ll 4$

$0\ 001\ 1000$

$= 1 \ll 9$

$\hookrightarrow 16 + 8 = 24 \#$

$= 1 * (2^9)$

shift left priority (الجمع)

$= 512 \#$

(1) Comma as an operator

a) `int i = (5, 10, 7);`

← listas لكتور فكتور اخوات فكتور comma دل مع listas دل اخوات

`int i = 5;`

`i = 10;`

`i = 7;`

$\{i=7\}$

$i = 32$

← output دل اخوات

b) `int p = 35;`

`int q = (P++, ++P);`

$q = 32$

$P = 35$

$q = P++$

$q = 35$

$P = 36$

$q = ++P$

$q = 37$

$P = 37$

$\{q = 37\}$

← output دل اخوات

c) `int j = (F1(), F2());`

هنا هنا دل return دل F1() دل Call دل return دل F2() دل Call دل return دل F1() دل F2()

الآن دل return دل F1() دل F2() دل return دل F2() دل F1()

و دل اخوات من المفهوم يكون متخزن في ال ذاكرة صورة

. F2() دل return value دل

(2) Cinema as a separator

a) int a=3 , b=4 , c=5 ;

the α -azide β -keto ester has the following structure:

$$b = 4 \rightarrow$$

c-5

$(\forall x d_1 \neq d_2 \rightarrow (\exists i \geq 0 (d_1[i] \neq d_2[i]) \vee \exists j < \underline{x}))$

b) definition of function call lexical scoping

int Fun(x) {
 if (x < 0) return -1;
 else if (x == 0) return 0;
 else if (x > 0) return 1;
}

{

$$\text{int } z = x + 7 \text{ ;}$$

return z;

3 ~~1983~~ ~~1983~~

int main()

3

```
int a=1, b=2, c=3;
```

```
printf("%d", fun(a,b,c));
```

۳

c) int ai

$$a = 3, 4, 8,$$

PrintF("asd", a);

discord cele

discard glasses و میکاریم و دل نماید ↳

discard *aglaie* و *يَا تَرَا* و *مَدِّ* \rightarrow

d) `int a=3, 4, 8;`

$$\int \text{int } a = 3$$

inty;

int 8;

دی تک عین دے

الخط العرضي المترافق مع خط الاستواء (5)
int 4; / int 8; **Leave**

Comma operator & separator

for loop

الله عبارة عن

$\leftarrow \{g \circ s\}$

Ex \Rightarrow For ($a \geq 0, b \geq 0$; $a < 10, b < 5$; $a++$, $b++$)

```
printf("%d %d", a, b);
```

the condition is given as $\frac{dy}{dx} = 0$

لعدد ٥ كل دور على فم الحال

المكتوب بـ (ج) ← المكتوب بـ (ج) ← المكتوب بـ (ج)

أعما لو هو خطب أتواس وجواهم

الـ Condition هي صيغة فرض العالة في

$b < 5$

الحاله دي هى من اكشن

`sizeof()`

خالد ناهضه هر من هیشوف آئی عملیات جواه هر چند هیشوف الخرج بتایهم کلهه صیغه نویه ایه رسک انسان کدا هنارف نایج (size of) .

ex → int i=5, j=10, k=15;

```
printf("%d", sizeof(k) - i + j);
```

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

Optimization

هذا يعني أن التأثير التعبدي هو تأثيرات الـ output

$\leftarrow \text{Sizeof(int)} = 4 \quad \leftarrow \text{int} \quad \leftarrow \text{پیکچر یعنی ۴ باینری}$

دال ک هنگه همیز و میانه ۱۵ هستن یا دنامه همانش

العملية الموجودة جوا الـ (أ) من حزام

Bunkers 10/19 x 3 net 6 - ext 202 290

میتوانیم اینجا `int a, b, c;` را بگذاریم و آنرا در کنار `main` قرار دهیم.

$a = b = c = 100$ i.e. the memory will be filled with 1's.

a, b, c n

و بعد كذا نجده في كل مكان

c b q

int a = b = c = 100; auto single_val d, e using

In memory of all who made us, we a

error java will not declare $\leftarrow b, c \parallel$

int a, b, c;
if(a == b == c) → TRUE

`a = b = c = 100; with a) this else` \Rightarrow False

$$a = b \quad t = c$$

$$100 = 100 + c$$

True

(1 == 100) \Rightarrow False

二〇〇九

`int a=10, b=20, c=30;`

`if (c>b>a) → true`

`else → False`

`(c>b)>a`

`30 > 20 > 10`

`True → 1 > a → (1 > 10)`

`False`

Explicit conversion is done programmatically

Implicit conversion is done automatically.

Type casting is to Explicit

Conversion of large number to small number is Explicit

Conversion of small number to large number is implicit

note

حالة خاصة بالـ Prefix, PostFix

`int a=1, b=0;`

`b = (a++) + (a++);`

هنا ادنا لو استخدناها صيغ عينانية اد a,b ففيهم

بـ 3,3 سـ من مـ نـ كـ لـ ةـ ءـ اـ دـ اـ تـ اـ مـ شـ عـ اـ رـ فـ يـ اـ دـ اـ

هـ يـ هـ اـ ئـ تـ رـ مـ خـ الـ اـ لـ اـ وـ تـ مـ ①ـ وـ لـ اـ رـ قـ ②ـ وـ نـ اـ

فـ نـ خـ تـ اـ رـ اـ Compiler dependent لـ اوـ مـ وـ جـ وـ رـ ةـ فـ يـ الاـ خـ تـ يـ اـ رـ اـ

عـ اـ طـ اـ دـ اـ بـ وـ وـ تـ كـ يـ طـ بـ لـ وـ هـ مـ شـ مـ وـ جـ وـ دـ هـ تـ حـ مـ

اـ يـ ← لـ اـ اـ لـ ا~ Statement مـ كـ تـ وـ يـ ئـ زـ سـ طـ رـ دـ اـ دـ ا~

هـ نـ تـ عـ لـ مـ اـ لـ ا~ الـ يـ مـ يـ اـ دـ ا~ لـ او~ صـ مـ مـ طـ و~ لـ ا~

فـ جـ مـ ا~ PrintF خـ اـ لـ ا~ الـ يـ مـ يـ ا~ دـ ا~

الـ يـ مـ يـ ا~

ex → int a = 1, b = 0;
 $b = a++ + a++$

$i++ (i++)$

a =

b =

يتبع على اختلاف المبرمجs من حيث الاختيارات Compiler dependent

بـ $b = a++ + a++$ كمكتوب في statement مارك بكن مثلاً

printf جوا

ـ يستعمل من الممكن للبيئات

$$\begin{aligned} b &= a++ + a++; \\ b &= \downarrow \quad \downarrow \\ b &= 1 + 2 = 3 \end{aligned}$$

a = 2 a = 3

$$\therefore \boxed{b = 3} \quad \boxed{a = 3}$$

ex → int i = 10;

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

الحل ← على اختلاف المبرمجs من حيث الاختيارات Compiler dependent

ـ الملام مكتوب جوا جملة printf

ـ يستعمل من الممكن للمبرمج

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

ـ يرجع في المخرج $i+1$ ، $i+1$ ، $i+1$

\therefore طبعاً $i+1$

ـ متزوج الأول ويدرس $i+1$ ، $i+1$ ، $i+1$

ـ نطبع الـ i

$i = 12$

print $\Rightarrow 12$)

ـ نطبع الـ i
 \therefore print $\Rightarrow 12$)

$i = 11$

outPut $\rightarrow 12 10$

(note)

 $(++i) + +;$

Compile error because
in C, prefix and postfix operators
all need L-value to perform operation
return R-value

 $(++i) + +;$ $(5) + +;$ $5 = 5 + 1$

not invalid L-value → memory location
error gives different result

 $i = 4$

(note)

OR حالات $true || \boxed{\quad}$ $= true$

unless (unless) optimization algorithm does ←
true unless there is a condition
true or with OR

ex → $i = 5, j = 6$ $i || ++j \rightarrow \textcircled{5} || ++6 \rightarrow \text{true} || \boxed{\quad} = \text{true}$ at end → $i = 5, j = 6$ $\textcircled{j} \rightarrow \text{false because } j \neq 0 \text{ at end}$ AND حالات $\text{False} \& \& \boxed{\quad} = \text{False}$

unless (unless) optimization algorithm does ←
false and false unless there is a condition

 $\cdot \text{False and with AND}$ ex → $i = 0, j = 3$ $i \& \& ++j \rightarrow \textcircled{0} \& \& ++3 \rightarrow \text{false} \& \& \boxed{\quad} = \text{false}$ at end → $i = 0, j = 3$

$(-1 \ll 4)$

if ① represented in 2Byte

 $1 \rightarrow 0000\ 0000\ 10000\ 0001$
 $1111\ 1111\ 11111111\ 11111111$ 1's comp
 $(-1) \rightarrow 1111\ 1111\ 1111\ 1111$ $-(-1) = 0x(FFFF)$ $(-1 \ll 4)$

the 4th bit will be 0000

result = $0x(-FFFF)$ then result $\rightarrow 0000\ 0000\ 0000\ 0000$

(note) write "Hello world" without semi colon

الحل \rightarrow حوا انا صدقا داخل شرط لـ if

if (printf("Hello,world") == 3)

التحمير \rightarrow جوا ال condition الخاص بـ if ملائم

جملة printf فالجملة هي ترجع us

بعد حرف (الكلمة ان هنطبعها) يعني هنا ترجع

زايدة خلفها $\{0\}$ space يتعدد أتفع

حرف زائد حرف داتا

if (1) \rightarrow true

داتا صفر و يقوم طبعarakma

Hello world

linker error

دف في حالة لما تجنس تحمل link file

execution file على المكتبات lib file مع

ذاتي مكتبات lib file الذي يطلب

في عملية الربط وهو صواب

 \rightarrow Can not be generated the link of wrong

Function prototyping or incorrect header files.

بـ error

Semantic error

هذا النوع من الأخطاء ينبع من خطأ في التحويل من نوع البيانات Function لـ type casting.

إذاً هنا نظرنا إلى Function ، فـ casting عملت كما يدور في المفهوم.

Semantic error هو خطأ في المفهوم.

ما تدل على خطأ في المفهوم؟

في الواقع هناك خطأ في المفهوم.

ما هي هذه الخطأ؟

warning (رسالة تحذير) هي الخطأ التي تقوف البرنامج ولا يمكّنه العمل.

في الغالب خطأ في المفهوم هو خطأ في التحويل من نوع البيانات.

Note

الرقم الدي قبله \leftarrow hexa \leftarrow 10 \leftarrow 0x

الرقم الدي قبله \leftarrow bin \leftarrow 10 \leftarrow 0b

الرقم الدي قبله \leftarrow OCT \leftarrow 10 \leftarrow 0

الرقم المكتوب بـ باس \leftarrow Dec \leftarrow 10 \leftarrow 0

$$\text{ex} \quad \text{int } i = 0x10 + 0o10 + 10$$

hexa OCT Dec

$$i = 16 + 8 + 10 = (34)_{10}$$

$$(34)_b \Rightarrow (22)_{16}$$

$$= (22)_{16} \#$$

$0x10 \Rightarrow 16$
$0o10 \Rightarrow 8$
$10 \Rightarrow 10$

int x;

لـ declaration مكان في الـ memory وـ definition

definition \leftarrow corrupted data

ولماً إهنا ينكر أن var

يـ corrupt the program وـ accept user input

دـ declaration \leftarrow

Corrupted data

byte

int x = 0;

دـ initialization

بـ corrupted data

تحاول var

التاريخ:

Ex-→ extern int x;

declaration

الـ File في definition يختلف عن File في declaration في أنه يحتوي على معلومات عن المنهجيات والمتغيرات التي تم إنشاؤها.

ex → type varname = value;

دانيه الـ ۳ حاجات

بيانات عوقبة var وجزئيات مكان في الذاكرة definition (١)
بيانات المتغيرات المعرفة في الملف declaration (٢)
بيانات ادخال ارثها ارثها قيمة ابتدائية (Value) أو initialization (٣)
بيانات او البيانات المفتوحة corruption data (٤)

note

$$a=5, b=4$$

~~just mentioned~~ (v)

$\text{if } (a == b); \rightarrow$ statements like Null statements

```
printf("Equal");
```

رسانایران (۹) ای خوش

body (مَنْجِلٌ) if glottis is closed body مَنْجِلٌ

- last last last

```
int i=3;
```

Float y=3.0;

```
if(x==y) { printf("true"); }
```

Condition $\Rightarrow (z = \bar{z} \cdot 0)$

(int == float) ← implicit conversion

الناتج (Float)، (الكتاب) معاوی، (2007)

يُقْرَأُ الْمُكَبِّلُ

L output \rightarrow true

note**switch case** في حالة الـ

switch(x) عارض تبعه ←
Case X1 ده فلسط كائن في اد ←
 ↗ Constant ثابت int ← دزمه يكون رقم ←
 ↗Float يكون ← مسافر ←
Case 15: case → مسافر ←
case 15: دنطفعهم نفس المقد ←
 ↗ duplicate case value دواعي ← compile error

note**int i=1; if(i++ && (i==1))**

a) condition true

b) condition False

i(d==0)?i

i++ ← Postfix هنا هي قيمة الأول الـ

(i=1) → (i=2)

1 && (i==1)

1 && (z==1)

true && False

out put => **False**

: E: 700