فرضا لو عندنا Array وانا عايز اعرف مساحتها اعمل ايه؟؟

الحل:

```
u8 size=sizeof(array)/sizeof (array[0]);
```

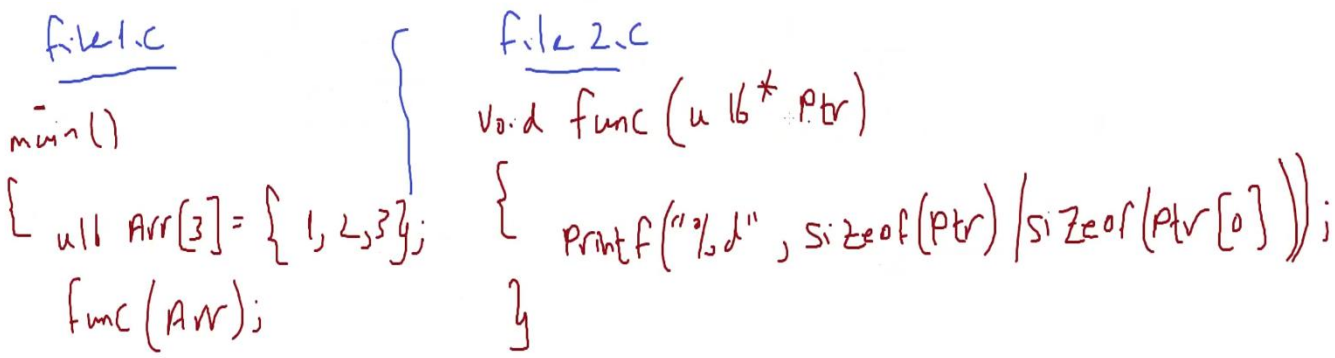
لكن ال sizeof ما هي الا operator بيفهمها البرنامج اثناء الbuilding time او بيترجمها اثناء الbuilding time

طيب انا في ال building time انا ببقي عارف ال size الخاص بالحاجه الي انت باعتها :والله علي حسب!!

يعني لو بتسالني علي int انا ببقي عارف ان ال compilerهنا بيكون مخصص 4byte مثلا بدون ما اشوف أي متغير int في الميموري ببقي عارفها من الاعدادات

لكن ولو دي حاجه ال compiler ميعرفهاش

مثال:



في المثال عندنا في الfile 2.c اثناء بناء البرنامج بيكون الsizeof بما انه operator بيكون ترجمته من مهام الcompiler

ونتذكر ان ال compiler بيدخل علي كل ملف لوحده بمتغيراته لوحده مش شايف أي حاجه في الملف التاني في الproject

في خلال ال compile time ال function مش بتكون معاها قيم ال parametar بتاعتها او ال input بتاعتها وبتحصل عليهم اثناء ال run time

فالمشكله الوقتي ال sizeof متفكتش من ال compiler (متعوضش بالاسمبلي او الmachine code المناظر ليها) لان مفيش قيمه يتنفذ عليها والملف الوقتي في ال run time في ايد الCPU لان ال compiler مطلعش Error وطلع رقم random من دماغه كده😂

فمينفعش استخدم الطريقه دي لمعرفه عدد عناصر ال Array في ال C

الحل : ممكن نتعامل معاها زي ما بنتعامل مع الString

بنكون عارفين اخر عنصر في ال String مثلا بيبقي 0\ وبالتالي بنبقي عارفين ال Size بتاع الString

project 1.c

```c
#include <stdio.h>
#include "STD_Types.h"
int main(void)
{
    printf("%d",sizeof(void));
    return 0;
}
```

```
C:\Windows\System32\cmd.exe

mo:gcc "project 1.c"

mo:a
1
mo:
```

Void pointer

احنا لو جربنا نشوف حجم ال void قد ايه : بتبقي Compiler dependent

Void * ptr ;

بوينتر بيشاور علي ال1byte ليه مستخدمناش ال char علشان هي كمان بتشاور علي 1byte

ليها استخدامات هنتكلم فيها السيشن الجايه بس هي بتبقي compiler dependent

فيه فانكشنز بترجع void pointer بدل ما يلزم نفسه بال int مثلا وانا عايزهم byte1 بس حاجه زي Stander block كلنا متفقين عليها

طيب انا عايز استخدمها كا int وهو باعتها void pointer اعملها ازاي

هتبقي عن طريق الcasting

Type cating

Type change:

u8 x=10;

x=10000 //مش هتنفع هنا لان ال10000 كبيره عليه

(Require type) variable

(u16) x=10000;

هتتغير في السطر ده بس وبعد كده بترجع لاصلها

اهم استخدام ليها بيبقي في الvoid pointer

فانكشن مثلا بـreturn ال void pointer وانا محتاج استخدمها كint pointer

int *ptr=(int*)func(4);

اول ما تشوف ال void pointer برمج مخك اعمله casting

مثلا لو جايلي من ال user address وانا مش عارف استقبله في أي نوع من ال pointer اعمله void pointer

User defined data type

هو type علي مزاجي بيعمل الي انا عايزه بالطريقه الي انا عايزها

Struct:

Syntax:

Struct name{

الي انت عايزه بقي

};

كده احنا عرفنا بس بس لسه ماخدش مساحه في الميموري

ازاي اعمل منه object

Syntax:

struct Struct_name object_name;

مثال:

```
Struct employee
{
u16 salary;
u8 deduction;
}
Int main(void)
{
struct employee ali;
}
```

مساحته بتبقي من مساحه المتغيرات الي بتبقي جواه

لو عايزين نضيف داتا اثناء التعريف

```
Struct employee ali ={3000,200};
Struct employee amged ={2000,100};
```

طيب بعد التعريف

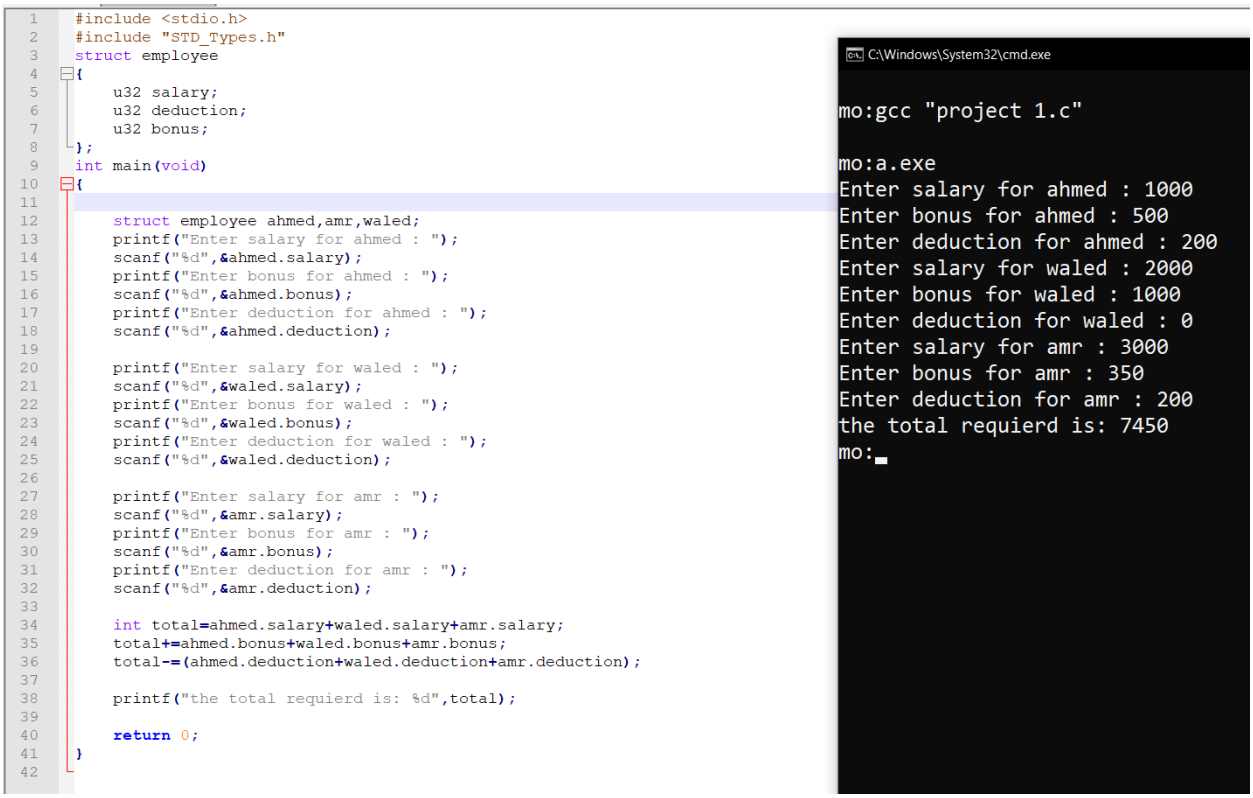
```
Struct employee ali;
ali.salary=3000;
ali.dedction =200;
```

انا ممكن اطبع أي حاجه عادي

```
Printf(“%d”,ali.salary);
```

وممكن اسجل فيه أي قيمه

```
scanf(“%d”,&ali.dedction);
```



```c
#include <stdio.h>
#include "STD_Types.h"
struct employee
{
    u32 salary;
    u32 deduction;
    u32 bonus;
};
int main(void)
{

    struct employee ahmed,amr,waled;
    printf("Enter salary for ahmed : ");
    scanf("%d",&ahmed.salary);
    printf("Enter bonus for ahmed : ");
    scanf("%d",&ahmed.bonus);
    printf("Enter deduction for ahmed : ");
    scanf("%d",&ahmed.deduction);

    printf("Enter salary for waled : ");
    scanf("%d",&waled.salary);
    printf("Enter bonus for waled : ");
    scanf("%d",&waled.bonus);
    printf("Enter deduction for waled : ");
    scanf("%d",&waled.deduction);

    printf("Enter salary for amr : ");
    scanf("%d",&amr.salary);
    printf("Enter bonus for amr : ");
    scanf("%d",&amr.bonus);
    printf("Enter deduction for amr : ");
    scanf("%d",&amr.deduction);

    int total=ahmed.salary+waled.salary+amr.salary;
    total+=ahmed.bonus+waled.bonus+amr.bonus;
    total-=(ahmed.deduction+waled.deduction+amr.deduction);

    printf("the total requierd is: %d",total);

    return 0;
}
```

```
C:\Windows\System32\cmd.exe

mo:gcc "project 1.c"

mo:a.exe
Enter salary for ahmed : 1000
Enter bonus for ahmed : 500
Enter deduction for ahmed : 200
Enter salary for waled : 2000
Enter bonus for waled : 1000
Enter deduction for waled : 0
Enter salary for amr : 3000
Enter bonus for amr : 350
Enter deduction for amr : 200
the total requierd is: 7450
mo:
```

طيب فرضا انا عندي ahmed كان نفس ال salaryوال bonusوال deduction بتاع ali ممكن اعمل ali= ahmed علطول

بس مينفع ازود حاجه او انقص حاجه ومفيش == ومفيش جمع او طرح او ضرب او اقسمه او أي عمليه منطقيه او حسابيه اخري ولازم يكونوا من نفس النوع

Passing struct to function

نعمل function بتاخد ال Struct كا input

u16 calc(stuct employee x)

{ statements }

كانه متغير عادي وممكن اعمل functionنوع struct employee عادي

طيب هو انا كل مره هستخدم employee هكتب قبلها Struct ؟؟

لا في طريقه تانيه وهي باستخدام typedef

typedef struct {

u16 salary;

u16 bonus;

}employee;

واستخدمه عادي

employee ahmed={2000,200};

ليه مبنكتبش اسم ال employee فوق جمب struct علشان ده بيبقي الاسم القديم الي typedef هيستبدله بالاسم الجديد الي هيجي بعد تعريف الstruct فبنكتبه بعد تعريف ال Struct كامل

Pointer to structure

زيه زي أي متغير عادي يعني

يتعمله Derefrancing ويتعمل منه pointer وعادي جدا

بس فيه حاجه مخصصه ليها وهي الArrow operator بيعمل بيها Derefrancing لل structure

Array of structure

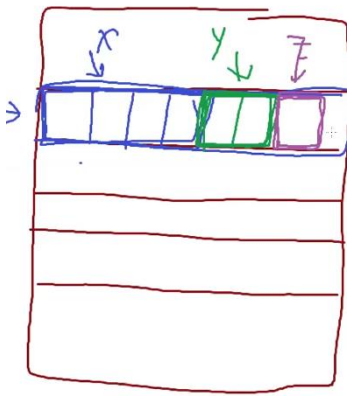
employee arr[3]={ahmed,ali,amr};

تعمل منه Array عادي زيه زي أي متغير

Bit field

انا عايز مثلا من المتغير bit4 مش ال8كلهم

مثال:



Typdef struct

{ u8 x:4; //use only 4bit

u8 y :2; //use only 2bit

u8 z:1; //use only 1bit

}my struct;

المساحه الي عايزينها 4+2+1 =7 bit

بس الي اتحجز في الميموري هو 1byte = 8bit

لان مفيش حاجه اسمها نص بايت ولو اخدت 9 بت هتبقي المساحه 2byte

بس مش هيستغل غير 9

طيب انا عملتها u32 في نفس المثال الي فات هيبدء ب byte4 ولواحتجت bit اكتر من 4byte هيذود هو byte 1 واحده بس زياده

Sizeof

Packing and padding of structure

لما بيعمل struct المساحه بتاعته بناء علي المتغيرات الي فيه بيبدء يحسب علي مساحه اكبر مساحه فيهم

فلو الكبير byte2 وعندي 5 متغيرات مهما كانت مساحتها فالمساحه الاجماليه هتبقي byte10

بس لو جه متغيرات ورا بعض مساحتهم الاجماليه= مساحه الكبير بيطحطوا في خانه واحده سوي غير كده ممكن يعمل خانه زياده يعني لو حصل متغير مساحته byte 2 جه بعد متغيرين اتنين مساحه كل واحد فيهم byte1 فالمساحه الاجماليه هتبقي byte4

طيب نفس المسال لكن الكبير جه ما بين الاتنين المساحه الكليه هتبقي byte6

حل مشكله زي دي اننا نرتب العناصر الكبير ورا بعض و الصغير ورا بعض.

كل الي سبق ده اسمهpadding

الي جاي هو االpacking

يعني هنطلب من ال compiler بدل ما يقسم بناء علي اكبر عنصر في الميموري

ده ازاي بقي؟

#pragma

دي key word بتروح لل compiler والي بيبدء ب # غالبا بيكون preprocessor ماعدا دي

فهتبقي الجمله الي بتعمل packing

#pragma pack(1)

وبتكون قبل تعريف ال structer وبتبقي في كل فايل لان ال compiler بيشوف ال project ملف ملف لوحده

لكن عيبها ان بدخل علي اكتر من memory location علشان اجيب value لمتغير عندي ثيمته اتقسمت علي اتنين byte

UNION

شبه ال Struct بس بتتبع قاعده

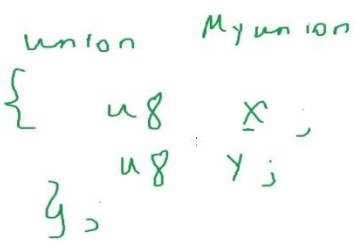
Only one element is valid at a time

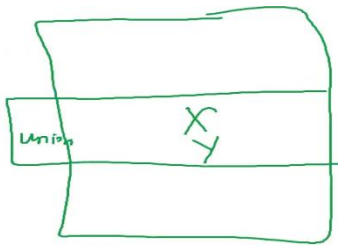
Syntax

union name

{ type name ; type name;};

```
union myunion
{
u8 x;
u8 y;
};
```



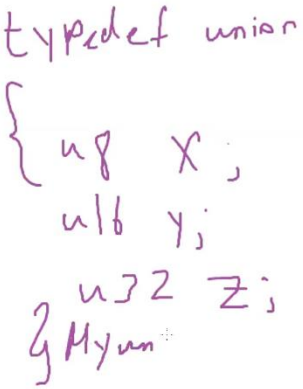


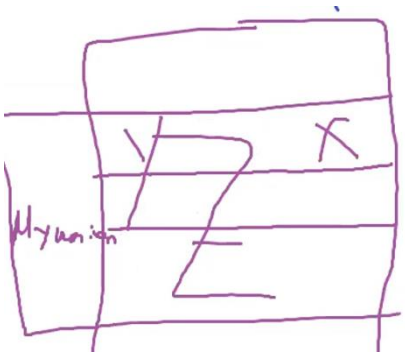
لو شوفنا حجم ال myunionهنلاقيه 1byte

وكمان لو كتبت في ال X قيمه وجيت اطبع ال y هيطبع القيمه الي اتخذنت في الx لانهم الاتنين بيتشاركوا في نفس المكان

في المثال ده المساحه بتتحدد بالكبير

يعني byte4





طيب بستخدمها في ايه؟

دي data type

اقدر اكتب فيها كلها مره واحده

واقدر اعمل اكسس لكل bit فيها