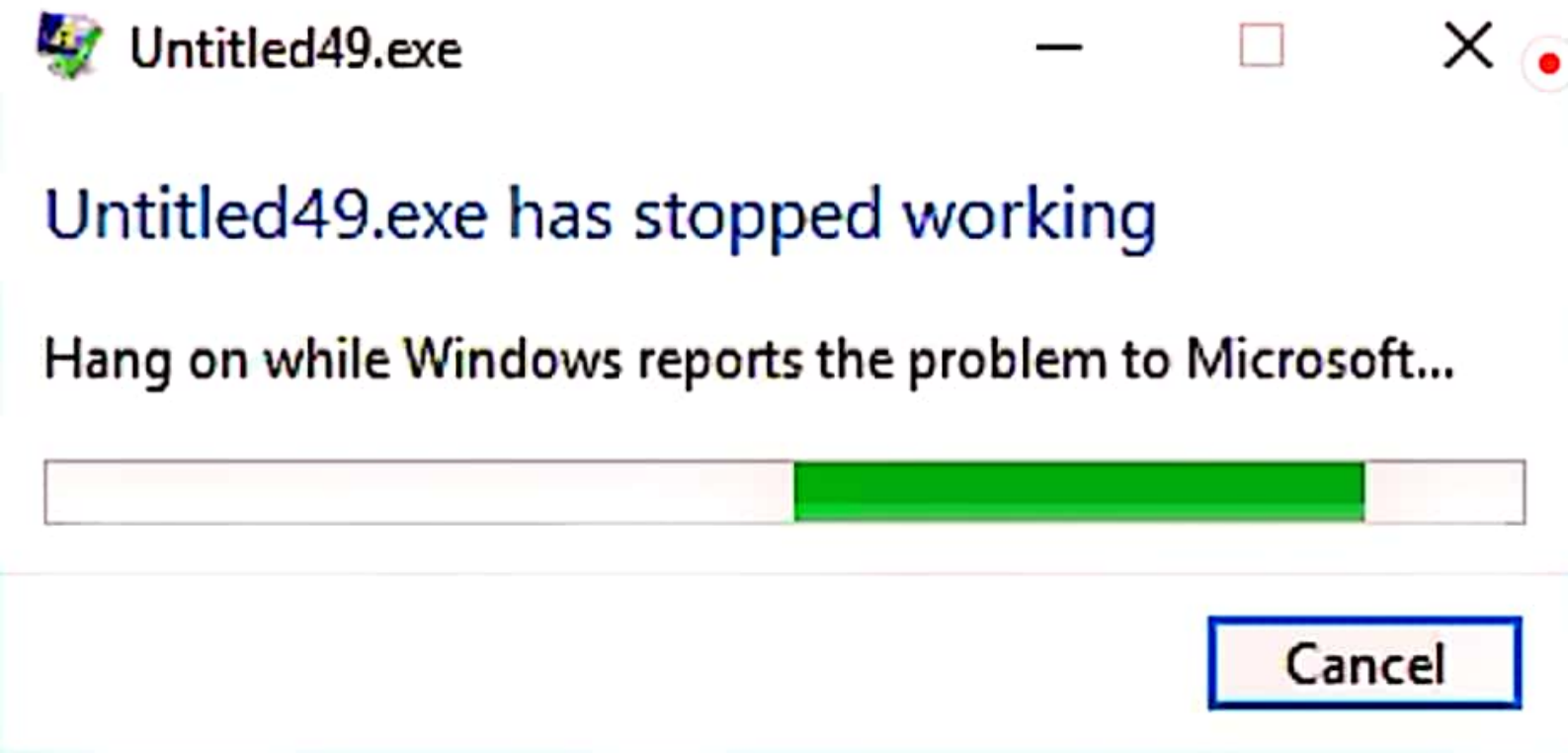
main | 100000000 | Go

```c
#include <stdio.h>
int main()
{
    int a[100000000];
    return 0;
}
```

uments\C-Free\Temp\Untitled49.c:6:2: warning: no newline at end of file

9. 0 error(s), 1 warning(s)
e\Documents\C-Free\Temp\Untitled49.exe

Insert Line ANSI DOS Modified: 14-Sep-20, 10:57:18 Ln: 62


Untitled49.exe
Untitled49.exe has stopped working
Hang on while Windows reports the problem to Microsoft...
Cancel

main | 100000000000 | Go

```c
#include <stdio.h>
int main()
{
    int a[100000000000];
    return 0;
}
```

e\Documents\C-Free\Temp\Untitled49.c...

main | 100000000000 | Go

```c
#include <stdio.h>
int main()
{
    //int a[100000000000];
    int *p;

    p = (int*) malloc( sizeof(int)*100000000000);

    return 0;
}
```

uments\C-Free\Temp\Untitled49.c:7: warning: large integer implicitly truncated to unsigned type

9. 0 error(s), 2 warning(s)
e\Documents\C-Free\Temp\Untitled49.exe

Insert Line ANSI DOS Modified: 14-Sep-20, 10:59:20 Ln: 139

int *P < named loc [stack]
unnamed loc {Heap}

int .

top

Structure pointer

The structure pointer operator -> is used to access members of a structure via a pointer.

-> is typed on the keyboard as a minus sign followed by a greater than sign.

If a pointer variable is assigned the address of a structure, then a member of the structure can be accessed by:

```
pointer-to-structure -> member_name
```

An equivalent construct is:

```
(*pointer_to_structure).member_name
```

Examples of the Two Accessing Modes

Declarations and Assignments

```
struct student temp, *p = &temp;
temp.grade = 'A';
temp.last_name = "Superman";
temp.student_id = 51;
```

Expression	Equivalent Expression	Conceptual Value
temp.grade	p -> grade	A
temp.last_name	p -> last_name	Superman
temp.student_id	p -> student_id	51
(*p).student_id	p -> student_id	51

Examples of the Two Accessing Modes

Declarations and Assignments

int a, *p = &a;

```
struct student temp, *p = &temp;
temp.grade = 'A';
temp.last_name = "Superman";
temp.student_id = 51;
```

Expression	Equivalent Expression	Conceptual Value
temp.grade	p -> grade	A
temp.last_name	p -> last_name	Superman
temp.student_id	p -> student_id	51
(*p).student_id	p -> student_id	51

Dynamic Memory Allocation

Data type integer is predefined

```
int  *p;
p = (int *)malloc( sizeof(int) );
```

4

```
struct one
{
        int a;
};

struct one *p;
p =(struct one *)malloc(sizeof(struct one));
```

4

Dynamic Memory Allocation

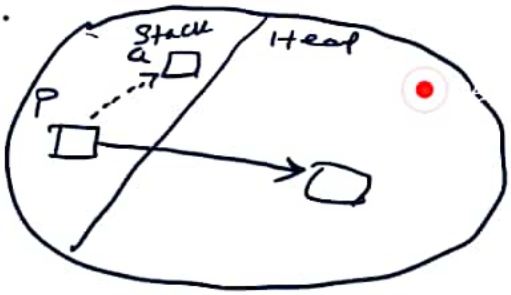
Data type integer is predefined

```
int  *p;
p = (int *)malloc( sizeof(int) );
*p=10;
```

Stack Heap

pointer → 10

```
struct one
{
        int a;
};

struct one *p;
p =(struct one *)malloc(sizeof(struct one));
p->a=10;
```



pointer → 10

Dynamic Memory Allocation

Exclusive: SRM KTR

Data type integer is predefined

```
int  *p;
p = (int *)malloc( sizeof(i
*p=10;
```

Stack

Heap

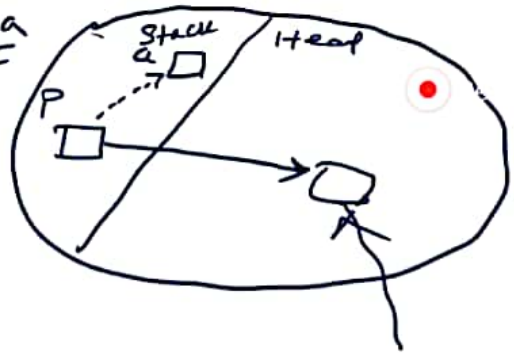
pointer → 10

data

Zoom

```
struct one
{
        int a;

ruct one *p;
=(struct one *)malloc(sizeof(struct one));
```

p=&a



pointer → 10

data

Next
Previous
Last Viewed
See All Slides
Zoom In
Custom Show
Show Presenter View
Screen
Pointer Options
Help
Pause
End Show

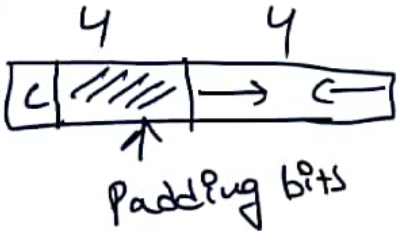
Laser Pointer
Pen
Highlighter
Ink Color
Eraser
Erase All Ink on Slide
Arrow Options

one.b

```c
#include <stdio.h>

struct one
{
    char a;
    int b;
}x;

int main()
{
    printf("%d\n", sizeof(x));
    return 0;
}
```

```
struct one
{
        char c;
        int a;
};
```

— sizeof(?)



one.c

```c
#include <stdio.h>

struct one
{
    char c;
    char a;
    int b;
}x;

int main()
{
    printf("%d\n", sizeof(x));
    return 0;
```

"C:\Users\home\Documents\C-Free\Temp\Untitled49.exe"

Zoom

```
Press any key to continue . . .
```

Unmute Start Video Share Participants More

"C:\Users\home\Documents\C-Free\Temp\Untitled49.exe"

)12

Zoom

Press any key to continue . . .

Unmute Start Video Share Participants More

one.c

```c
#include <stdio.h>

struct one
{
    char a;
    int b;
    char c;
}x;

int main()
{
    printf("%d\n", sizeof(x));
    return 0;
```

Documents\C-Free\Temp\Untitled49.c...

0 error(s), 0 warning(s)
Documents\C-Free\Temp\Untitled49.exe

"C:\Users\home\Documents\C-Free\Temp\Untitled49.exe"

Zoom

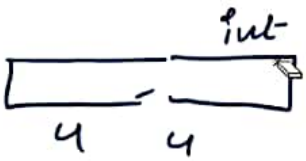
```
Press any key to continue . . .
```

Unmute Start Video Share Participants More

union one | union | Go

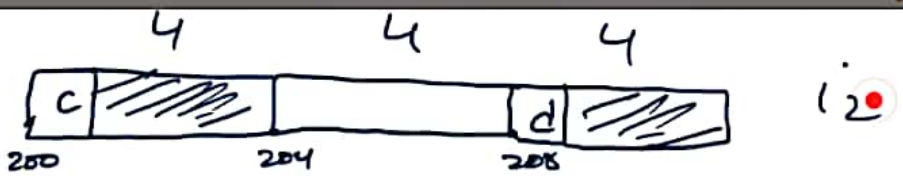
```c
#include <stdio.h>

union one
{
    char a;
    int b;
    char c;
}x;

int main()
{
    printf("%d\n", sizeof(x));
    return 0;
```

```
struct one
{
        char c;
        int a;
        char d;
};
```



= 8.

```
struct one
{
        char c;
        int a;
        char d;
};
```



```
{
    char c;
    int a;
};
```

double
char } → 1c

```
struct one
{
        char c;
        int a;
};
```

Union

last qualification

X - int

XII - int

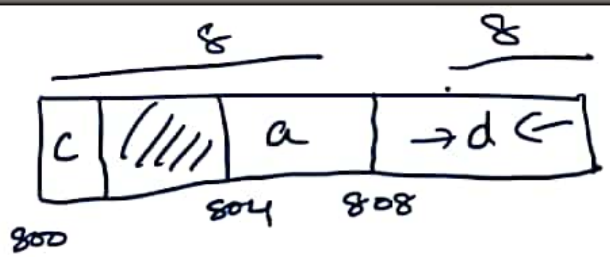
UG - int

PG -

DOC -

Post -

```
struct one
{
        char c;
        int a;
};      double d;
```



Dynamic Memory Allocation


```
struct one
{
        char c;
        int a;
};
```

```
struct one *p;
p = (struct one *) malloc(sizeof(struct one));
```

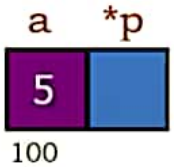
```
p->c='c';
p->a=10;
```

C	10

Dynamic Memory Allocation


```
struct one
{
        int a;
        int *p;
}*x;
```

```
int main()
{
        x = (struct one*) malloc(sizeof(struct one));

        x->a=5;
```



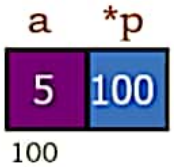
x->p = [ ]

addr of int loc

```
struct one
{
        int a;
        int *p;
}*x;
```

```
int main()
{
        x = (struct one*) malloc(sizeof(struct one));

        x->a=5;
        x->p=&(x->a);

        printf("%u %d\n",x->p,*(x->p));
```

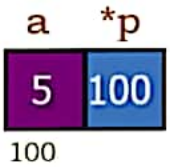
%p → Hex with prec



Dynamic Memory Allocation


```
struct one
{
        int a;
        int *p;
}*x;
```

```
int main()
{
        x = (struct one*) malloc(sizeof(struct one));

        x->a=5;
        x->p=&(x->a);

        printf("%u %d\n",x->p,*(x->p));
```



Dynamic Memory Allocation

Zoom

Exclusive: SRM KTR

```
struct one
{
        int a;
        int *p;
}*x;
```

```
int main()
{
        x = (struct one*) malloc(sizeof(struct one));

        x->a=5;
        x->p=&(x->a);

        printf("%u %d\r...x->p));
        return 0;
}
```

%d", *x)

*x

a	*p
5	100

→ 100

1 2 - - -

*a

Next
Previous
Last Viewed
See All Slides
Zoom In
Custom Show
Show Presenter View
Screen
Pointer Options
Help
Pause
End Show

Dynamic Memory Allocation


```
struct one
{
        int a;
        int *p;
}*x;
```

Struct one
{
int a; → 4
char c; → 1
Struct one z; ?
}x;

Data Size

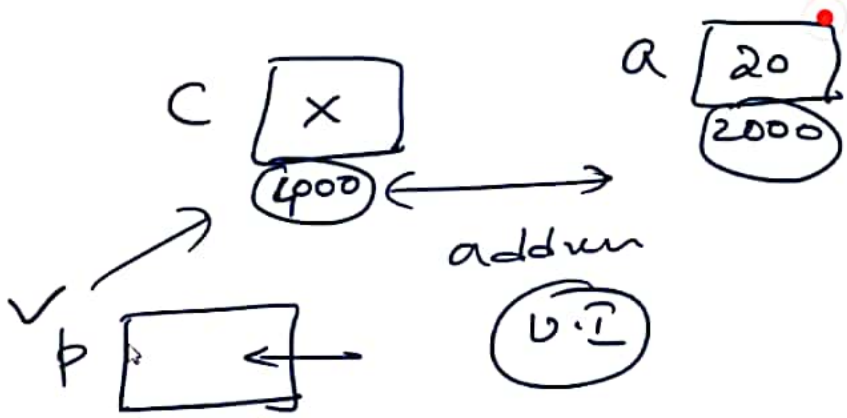
```
struct one
{
    int a;
    struct one *p;
}*x;
```

4

4

char c = 'x'

int a = 20;



```
struct one
{
    int a;
    struct one *p;
}*x;
```

```
int main()
{
    x = (struct one*) malloc(sizeof(struct one));

    x->a=5;
    x->p=x;
```

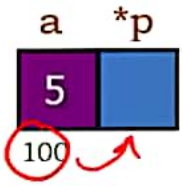
x -> p = x

add of struct

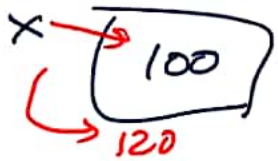
x -> p

int

add of int



struct



```
struct one
{
    int a;
    struct one *p;
}*x;
```

U.I 4

sizeof(int*)
(char*)
(void*)
(double*)

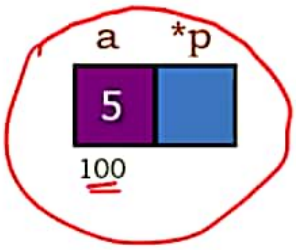
Dynamic Memory Allocation

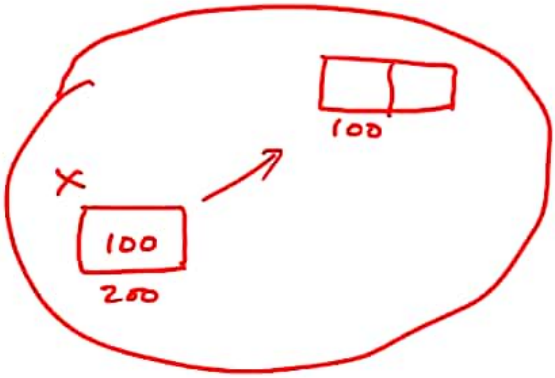
Zoom

Exclusive: SRM KTR

```
struct one
{
    int a;
    struct one *p;
}*x;
```

```
int main()
{
        x = (struct one*) malloc(sizeof(struct one));

        x->a=5;
```





?

x->P =

Dynamic Memory Allocation

Exclusive: SRM KTR

```
struct one
{
    int a;
    struct one *p;
}*x;
```

```
int main()
{
    x = (struct one*) malloc(sizeof(struct one));

    x->a=5;
    x->p=x;
```

a	*p
5	100

100

?

x->p = x

&x