



Assumptions

1. System or machine: (32 bit machine)
2. char (1 byte)
3. short int (2 bytes)
4. int (4 bytes)
5. double (8 bytes)

GEEKSFORGEEKS

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpStoryboardingSearch

ShareComments

New Reuse Slide Slides

Table

Pictures

Online Pictures

Screenshot

Photo Album

Shapes

Icons

3D Models

SmartArt

Chart

Forms

My Add-ins

Zoom

Link

Action

Comment

Text Box

Header & Footer

WordArt

Equation

Symbol

Video

Audio

Screen Recording

1

2

3

4

5

6

Structure member alignment and padding

Memory architecture:

The diagram illustrates two memory architectures. On the left, 'Byte Addressable Memory' shows a single vertical stack of memory cells with a data bus labeled 'D7 data bus D0'. On the right, 'Banked Memory' shows four separate memory banks labeled 'BANK 3', 'BANK 2', 'BANK 1', and 'BANK 0'. Each bank has its own data bus, with 'BANK 0' specifically labeled 'D31' and 'D0'. A horizontal bar at the bottom of the banks is labeled 'data bus (one memory cycle)'. To the right of the banks, an upward-pointing arrow indicates memory addresses: 0x0000, 0x0004, 0x0008, and 0x000C. The source 'GEEKSFORGEEKS' is noted at the bottom.

Slide 3 of 9

Notes

86%

6:22 PM 5/23/2019



Some Rules

1. Any data type will always store its value at the address which is multiple of the size of that data type.
Example: int will always start from 4,8,12,.....
2. any structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

GEEKSFORGEEKS



Sequential memory structure

```
typedef struct structa_tag
{
    char        c;
    short int    s;
} structa_t;    I
```

Bytes	0	1	2	3	4	5	6	7	8
Allocation	char		Short int	Short int					

Total size: 4bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

GEEKSFORGEEKS



Sequential memory structure

```
typedef struct structa_tag
{
    char      c;
    short int  s;
} structa_t;
```

Bytes	0	1	2	3	4	5	6	7	8
Allocation	char		Short int	Short int					

Total size: 4bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

GEEKSFORGEEKS



```
typedef struct structb_tag
{
    short int    s;
    char         c;
    int          i;
} structb_t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

GEEKSFORGEEKS



```
typedef struct structb_tag
{
    short int    s;
    char         c;
    int          i;
} structb_t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

```
typedef struct structb_tag
{
    short int    s;
    char         c;
    int          i;
} structb t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

GEEKSFORGEEKS

AutoSave Off

pointer_to_array_1

Vishal Gulia

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

programming questions bd

ramdump.py

gpu_info_script.py

new

basic.cpp

Code

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

// int 4 bytes

// double 8 bytes

// structure A

typedef struct structa_tag

{

char c;

short int s;

} structa_t;

// structure B

typedef struct structb_tag

{

short int s;

char c;

int i;

} structb_t;

// structure C

typedef struct structc_tag

{

char c;

double d;

int s;

} structc_t;

// structure D

typedef struct structd_tag

{

double d;

int s;

char c;

} structd_t;

int main()

{

printf("sizeof(structa_t) = %d\n", sizeof(structa_t));

//printf("sizeof(structb_t) = %d\n", sizeof(structb_t));

//printf("sizeof(structc_t) = %d\n", sizeof(structc_t));

//printf("sizeof(structd_t) = %d\n", sizeof(structd_t));

return 0;

}

typedef struct structb_tag

{

short int s;

char c;

int i;

} structb_t;

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

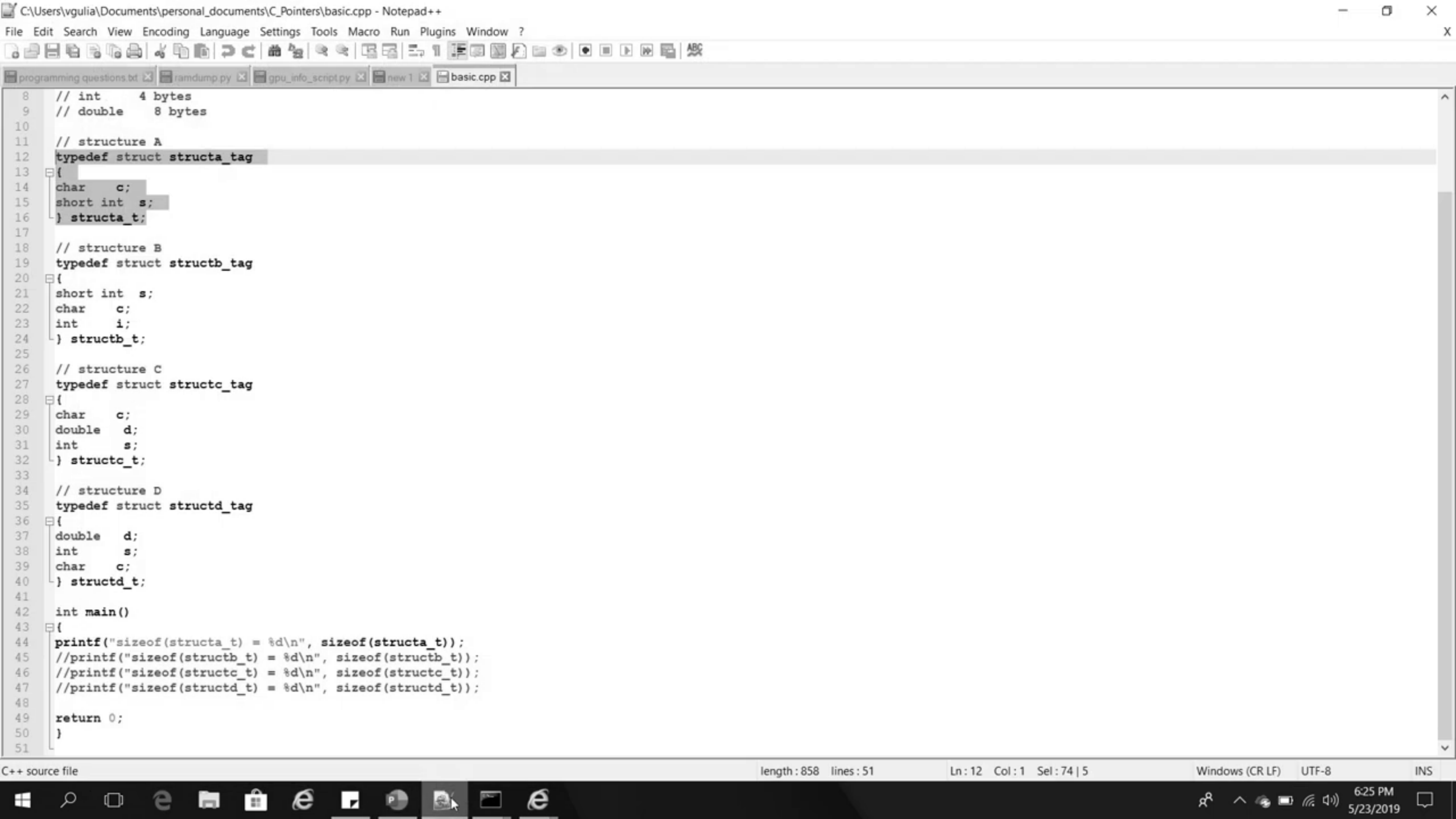
length: 858 lines: 51 Ln: 12 Col: 1 Sel: 74 | 5

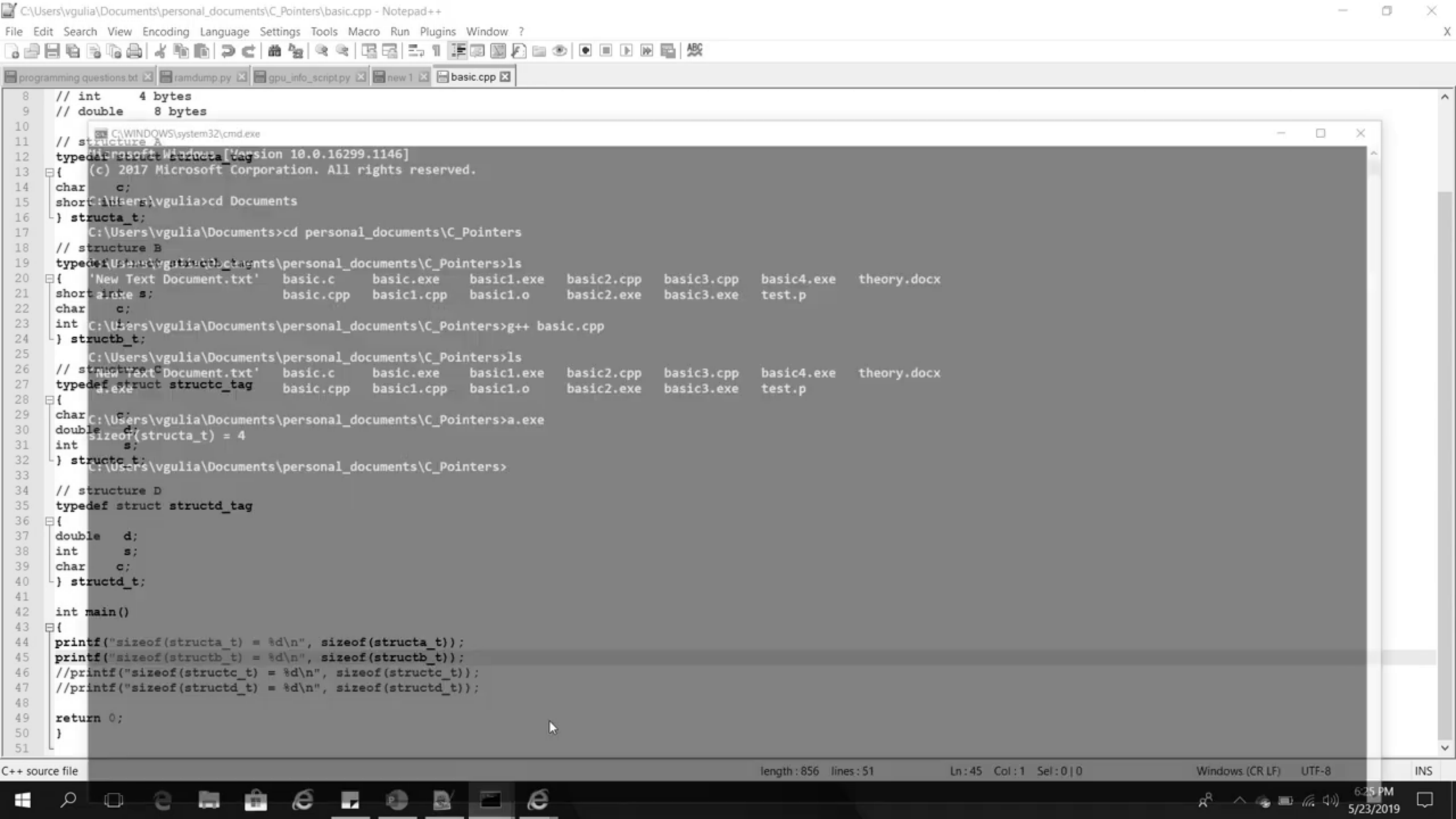
C++ source file

length: 858 lines: 51 Ln: 12 Col: 1 Sel: 74 | 5

Windows (CR LF) UTF-8

6:25 PM 5/23/2019





C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.16299.1146]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents

C:\Users\vgulia\Documents>cd personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
'New Text Document.txt' basic.c basic.exe basic1.exe basic2.cpp basic3.cpp basic4.exe theory.docx
a.exe basic.cpp basic1.cpp basic1.o basic2.exe basic3.exe test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp

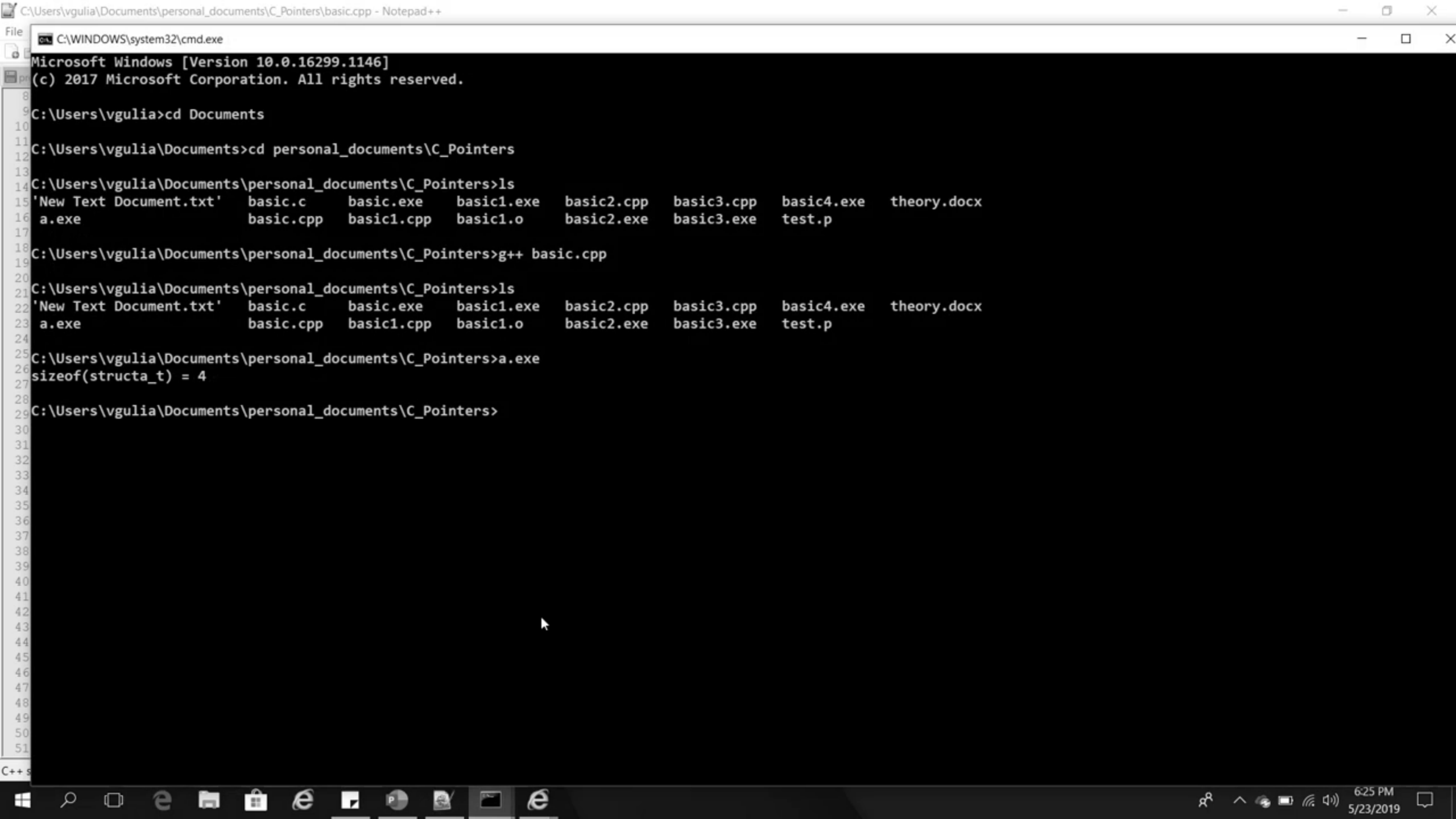
C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
'New Text Document.txt' basic.c basic.exe basic1.exe basic2.cpp basic3.cpp basic4.exe theory.docx
a.exe basic.cpp basic1.cpp basic1.o basic2.exe basic3.exe test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>a.exe
sizeof(structa_t) = 4

C:\Users\vgulia\Documents\personal_documents\C_Pointers>

```
1 // structure A
2 struct a {
3     int    a;
4 } structa_t;
5
6 // structure B
7 struct b {
8     int    a;
9     char    c;
10 } structb_t;
11
12 // structure D
13 typedef struct {
14     double  d;
15     int     a;
16     char    c;
17 } structd_t;
18
19 int main()
20 {
21     printf("sizeof(structa_t) = %d\n", sizeof(structa_t));
22     printf("sizeof(structb_t) = %d\n", sizeof(structb_t));
23     //printf("sizeof(structc_t) = %d\n", sizeof(structc_t));
24     //printf("sizeof(structd_t) = %d\n", sizeof(structd_t));
25
26     return 0;
27 }
```

C++ source filelength: 856 lines: 51Ln: 45 Col: 1 Sel: 0 | 0Windows (CR LF)UTF-8INS





```
typedef struct structb_tag
{
    short int    s;
    char         c;
    int          i;
} structb_t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.