

(8) C_117 \Rightarrow Structure Packing in C

* In structure padding; data alignment is done automatically by compiler to increase the speed and performance in parity of memory.

* But what if; we don't waste the memory and how not to waste memory?

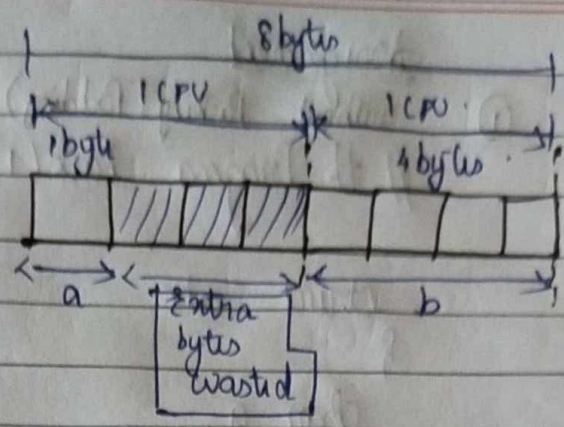
Structure Padding.

```

struct abc
{
    char a;
    int b;
};

main()
{

```



```

    struct abc s;
    printf("%d", sizeof(s)); // 8 bytes
}

```

* We don't want to waste those extra bits by Compiler and we want to pack the structure; then we should code

#Pragma Pack(1) → We turn on packing

* When we turn on packing by using this code; then Compiler will do packing but no padding, i.e. compiler will pack the structure members.

* By using structure packing; we can save the memory.

Structure Packing

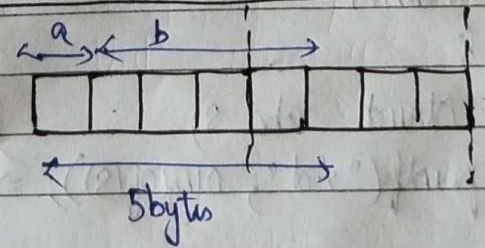
```

#Pragma Pack(1)
struct abc
{
    char a;
    int b;
};

main()
{

```

Note (X)(X)
only one byte alignment for 1 CPU



```

    struct abc s; printf("%d", sizeof(s)); // 5 bytes
}

```


Assignment

①

without Packing (Padding)

// #pragma Pack(1)

→ solve

Both without structure packing and also with structure padding.

```
struct abc
```

```
{
```

```
char a;
```

```
int b;
```

```
double c;
```

```
};
```

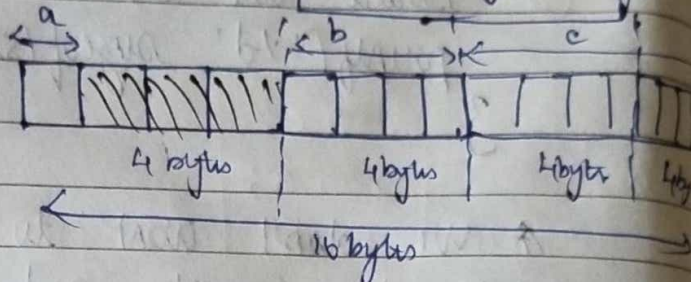
```
main()
```

```
{
```

```
struct abc s;
```

```
printf("%d", sizeof(s));
```

```
}
```



24 bytes

②

with packing

#pragma Pack(1)

```
struct abc
```

```
{
```

```
char a; -1
```

```
int b; -4
```

```
double c; -8
```

```
};
```

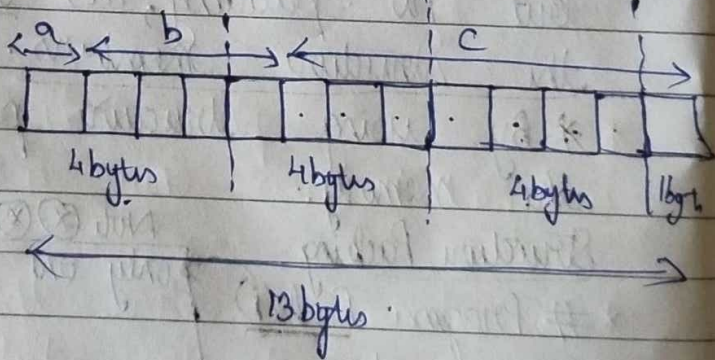
```
main()
```

```
{
```

```
struct abc s;
```

```
printf("%d", sizeof(s));
```

```
}
```



13 bytes

```

2  #include <stdlib.h>
3  /** 22-STRUCTURE PACKING IN C (1) **/
4  /* Structure Padding */
5  struct abc
6  {
7      char a;
8      int b;
9  };
10 int main()
11 {
12     struct abc s;
13     printf("size of memory allocated for object s of structure abc is %d", sizeof(s));
14     getch();
15 }
16 /* Structure Packing */
17 #pragma pack(1)
18 struct abc
19 {
20     char a;
21     int b;
22 };
23 int main()
24 {
25     struct abc s;
26     printf("size of memory allocated for object s of structure abc is %d", sizeof(s));
27     getch();
28 }
29 */
30

```

```

D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUCTURES\22_STR...
size of memory allocated for object s of structure abc is 8

```

```

3  /** 22-STRUCTURE PACKING IN C (1) **/
4  /* Structure Padding */
5  struct abc
6  {
7      char a;
8      int b;
9  };
10 int main()
11 {
12     struct abc s;
13     printf("size of memory allocated for object s of structure abc is %d", sizeof(s));
14     getch();
15 }
16 /* Structure Packing */
17 #pragma pack(1)
18 struct abc
19 {
20     char a;
21     int b;
22 };
23 int main()
24 {
25     struct abc s;
26     printf("size of memory allocated for object s of structure abc is %d", sizeof(s));
27     getch();
28 }
29
30

```

"D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNY'S LECTURE_STRUCTURES\22_STR...
size of memory allocated for object s of structure abc is 5

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 23-STRUCTURE PACKING IN C (1) **/
4  /* Structure Padding */
5  struct abc
6  {
7      char a;
8      int b; double c;
9  };
10 int main()
11 {
12     struct abc s;
13     printf("size of memory allocated for object s of structure abc is %d", sizeof(s));
14     getch();
15 }
16 /* Structure Packing */
17 #pragma pack(1)
18 /*struct abc
19 {
20     char a;
21     int b; double c;
22 };
23 int main()
24 {
25     struct abc s;
26     printf("size of memory allocated for object s of structure abc is %d", sizeof(s));
27     getch();
28 }*/
29
```

"D:\1. C C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNY'S LECTURE_STRUCTURES\23-STRUCTURE PACKING IN C (1)...
size of memory allocated for object s of structure abc is 16

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 23-STRUCTURE PACKING IN C (1) **/
4  /* Structure Padding */
5  /*struct abc
6  {
7      char a;
8      int b; double c;
9  };
10 int main()
11 {
12     struct abc s;
13     printf("size of memory allocated for object s of structure abc is %d",sizeof(s));
14     getch();
15 }*/
16 /* Structure Packing */
17 #pragma pack(1)
18 struct abc
19 {
20     char a;
21     int b; double c;
22 };
23 int main()
24 {
25     struct abc s;
26     printf("size of memory allocated for object s of structure abc is %d",sizeof(s));
27     getch();
28 }

```

"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUC

```

size of memory allocated for object s of structure abc is 13_

```