# Union assignments

**Mandatory**

1. Refer the code below and comment on size of the given structure considering
   1. Structure as union
   2. Structure as struct
   3. arr
   4. uarr

\_\_\_ Job

{

char name[32];

unsigned short ucount;

float salary;

int workerNo;

char \*orgname;

};

\_\_\_ Job myvar; //could of union or of struct

Struct Job arr[10];

Union Job uarr[10];

A. char->32 bytes

unsigned short->2 bytes

float ->4 bytes

int ->4 bytes

char ->4 bytes

Total :46 bytes

a. structure as union: 32 bytes

b. structure as struct :46 bytes

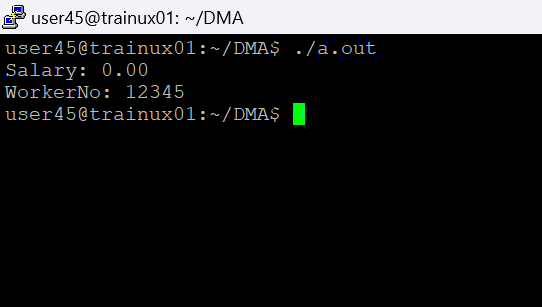
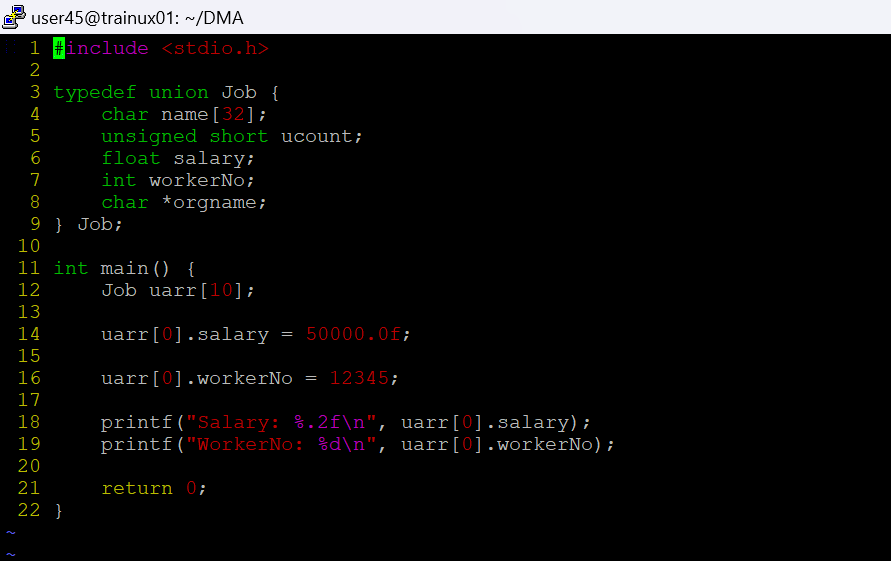
c. arr : 10\*46 bytes=460

d. uarr: 10\*32 bytes=320

2.Refer Job datastructure in Q#1 above. Using uarr, perform below operations.

* 1. Read and store salary
  2. Read and store workerNo

Comment on values of output if salary and workerNo are printed in order. Justify your statement.



1. Refer Job datastructure in Q#1 above. Assume that myvar is a structure variable. If I need to place 2 bytes (i.e 0x0102) as ucount using a char \*ptr then list all possible statements that can be used in \_\_\_\_\_.

[Let solutions include cases such as

* + 1. using base address of ucount
    2. using relative address of ucount w.r.t to base address of myvar]

int main()

{

char \*ptr = &myvar;

\_\_\_\_\_\_\_\_\_\_\_ = 0x01;

\_\_\_\_\_\_\_\_\_\_\_ = 0x02

}

i) \*ptr =0x01;

\*(ptr+1)= 0x02

ii) As ucount is at 32 we directly access that bit

\*(ptr+32)=0x01;

\*(ptr+33)=0x02;