Assignment 1

V. L. Narasimha Reddy - EE18BTECH11046

Download all python codes from

https://github.com/narasimha−123/EE4013/tree/main/Assignment−1/codes

and latex-tikz codes from

https://github.com/narasimha−123/EE4013/tree/main/Assignment−1/figs

1 Problem

Consider the following C program

```
#include <stdio.h>

struct OurNode
{
    char x, y, z;
};

int main()
{
    struct OurNode p = {'0', '1', 'a' + 2};
    struct OurNode *q = &p;
    printf("%c,%c \n", *((char *)q + 1), *((char
        *)q + 2));
}
```

The output of the following program is

2 Solution

The output of the given C program we get is

```
1,c
```

In the code, We are defining a new structure using struct.

```
struct OurNode
{
    char x, y, z;
};
```

A struct is a composite data type (or record) declaration that defines a physically grouped list of variables under one name in a block of memory, allowing the different variables to be accessed via a single pointer or by the struct declared name which returns the same address.

Memory Block created using struct

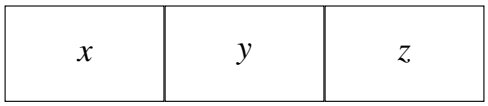

Fig. 0: Memory block created using Struct OurNode

As part of code the data types defined in the struct are same(char). So we can also assume this as a char array of size 3 in a single continuous block of memory.
Initially we created a block of memory using the struct and assigned three chars $'0', '1', 'a' + 2$ as values in memory fields. Then a pointer p is created to store the address value of the first char element of the struct.

```
struct OurNode p = {'0', '1', 'a' + 2};
```

Since the field members of the struct are chars, $'a'+2$ is stored as char $'c'$.
The struct created along with pointer pointing to the first element ia as shown below:

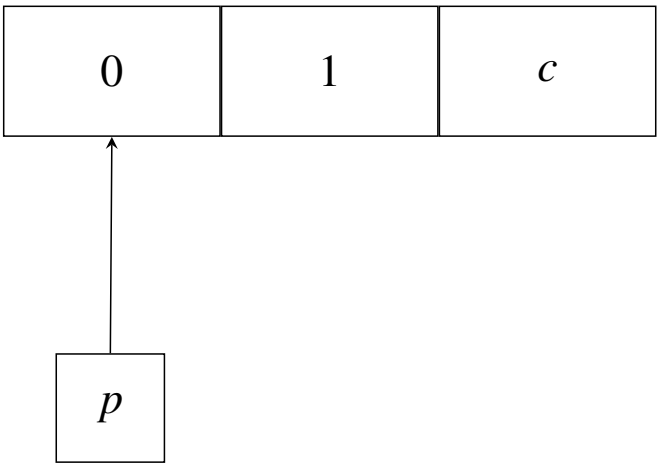

Fig. 0: struct OurNode p

Then we are creating a new pointer q and assigning it the address of previously created struct p. So new pointer q stores the address of the first element in struct p.

```
struct OurNode *q = &p;
```

The new pointer q points to the same memory address as that of pointer p

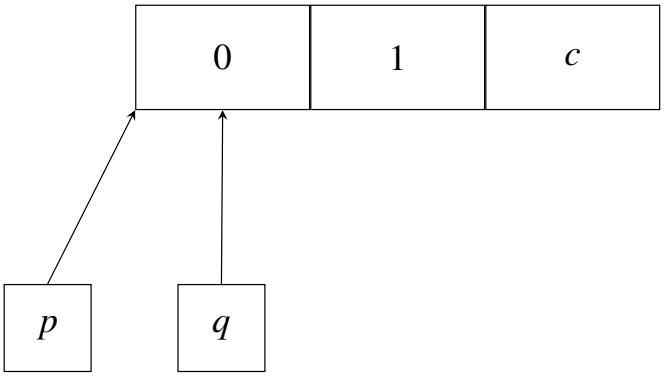

Fig. 0: OurNode p with pointer q

Now $q+1$ points to the address of the next element from the pointer q and similarlly $q+2$ points to the next address from the pointer $q+1$.

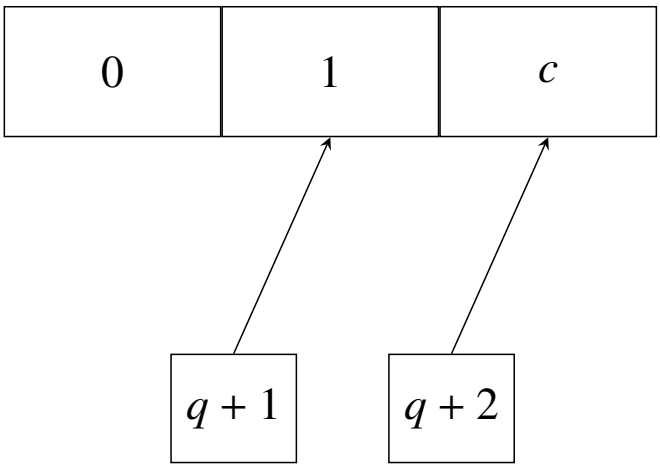

Fig. 0: pointers $q+1$ and $q+2$

```
printf("%c,%c \n", *((char *)q + 1), *((char
    *)q + 2))
```

So, now the above printf line prints the element stored at the address $q+1$ and $q+2$ points in the terminal i.e., **1**, **c**.