ASSIGNMENT_03

Question a:

● **Pointer(s) to a Structure** : in pointer to structure a pointer variable can point to the address of a structure variable. Below is an example of how we can declare a pointer to structure variable.

```
//Pointer(s) to a structure
#include <stdio.h>
struct student
{
        char name[30];
        int age;
        float weight;
};

int main()
{
   struct  student *stdPtr;
   struct student student1;
   stdPtr = &student1;

   printf("*************************************************\n");
   printf("Enter Name: ");
   scanf("%s", student1.name);

   printf("Enter age: ");
   scanf("%d", &stdPtr->age);

   printf("Enter weight: ");
   scanf("%f", &stdPtr->weight);

   printf("\n");
   printf("*************************************************\n");
   printf("Name: %s\n", student1.name);
   printf("Age: %d\n", stdPtr->age);
   printf("weight: %f\n", stdPtr->weight);

   return 0;
}
```

- **Structure containing Pointer(s)** : A pointer as a member of structure is known as structure containing pointers.

```
//Structure Containing Pointer(s)
#include <stdio.h>
struct student
{
   char *name;
   int age;
   char *course;
};
int main()
{
   struct student stu = { "ugyen", 19, "BSc.IT"};
   struct student *ptr_stu = &stu;

   printf("****************************************\n");
   printf("Name: %s\n", ptr_stu->name);
   printf("Age: %d\n", ptr_stu->age);
   printf("Course: %s\n", ptr_stu->course);
   return 0;
}
```

Question b:

Passing structure to Function(s) : we can implement passing structure to function in two ways i.e,
1. passing structure to a function by value
In this the whole structure is passed to another function value. That is the whole structure is passed to
another function with all the members and their value making this structure can be accessed from called
function.

```
//Passing Structure to Fuction
#include <stdio.h>
#include <string.h>
struct student
{
   char name[20];
   int age;
   char course[20];
};
void display(struct student record)
{
        printf("**********************************\n");
   printf(" Name: %s \n", record.name);
   printf(" Age: %d \n", record.age);
   printf(" Course: %s\n", record.course);
}
int main()
{
   struct student record;
   strcpy(record.name, "ugyen");
        record.age = 19;
        strcpy(record.course, "BSc.IT");
   display(record);
   return 0;
}
```

2. Passing structure to a function by address

The whole structure is passed to another function by address. That is, only the address of the structure is
passed to another function. The whole structure is not passed to another function with all members and
their values. So, this structure can be accessed from called function by its address.

```
//Returning a structure from a Function
#include<stdio.h>
struct student{
	char name[50];
	int age;
	char course[20];
};
struct student display(){
	struct student std;
	printf("*****************************************\n");
	printf("Name: ");
	scanf("%s",std.name);

	printf("Age: ");
	scanf("%d",&std.age);

	printf("Course: ");
	scanf("%s", std.course);
	printf("\n");
	return std;
}
int main(){
	struct student record;
	record = display();

	printf("*****************************************\n");
	printf("Name :%s\n",record.name);
	printf("Age :%d\n",record.age);
	printf("Course: %s\n", record.course);
	return 0;
}
```