Introduction to Pointers Assignments:

Mandatory

1. Refer the code snippet below. int main()

```
{
  char arr=”hello hi “;
  int *ptr = arr;

  printf(“sizeof ptr:%d, arr:%d”, sizeof(ptr), sizeof(arr));
  display(ptr); // display the address in hex and contents using pointer
}
```

Perform the following.

a. Implement the display() function (Use the “0x%x” formatting specifier to print addresses in hexadecimal.)

b. comment on the sizeof(ptr) and sizeof(arr)

```
#include<stdio.h>
void display(int *);
int main()
{
    char arr[]="hello hi";

    int *ptr = arr;

//  printf("sizeof ptr: %ld, arr:%d", sizeof(ptr), sizeof(arr));

    display(ptr);

    return 0;
}
void display(int *ptr)
{
    printf("0x%x\n",ptr);
}
```

OUTPUT:

0x71999bbf

2. Refer the code snippet below. int main()

```
#define MAX 100
#define SUCCESS 0
#define FAILURE 1

int main()
{
  char arr[MAX] = “Learning C“;
  char*ptr = arr;
  char appendstr[3]= “in my org”;

  printf(“Address of ptr:%x”, ptr);
```

```
  int ret = append(ptr, appendstr);// append the string

  printf("Address of ptr:%x", ptr);

  if (ret == SUCCESS)
  {
     display(ptr); // display the address in hex and contents using pointer
  }
}
```

Perform the following.

a. Implement the append() function to append the contents of the appendstr[] to arr using pointer.

   [Note: append() should only use its content and not manipulate it. Contents should be retained even after the call]

```
}
void display(char *ptr)
{

    printf("\nAddress is :0x%x", ptr);
    printf("\n Contents of pointer : %s\n",ptr);
}

int append(char *ptr,char *app)
{
    while(*ptr!='\0')
        ptr++;
    while(*app != '\0')
    {
        *ptr= *app;
        ptr++;
        app++;
    }
*ptr='\0';
 return SUCCESS;

}
```

```
Address of ptr:9b556610
Address of ptr:9b556610
Address is :0x9b556610
 Contents of pointer : Learning Cin my org
```

3. Refer the code in "pointer_prg.c". The functions swap_nums() and swap_pointers() are expected to swap the numbers and pointers respectively. But swap_pointers() is currently not giving the expected results. Analyse and the fix the issue.

```c
//swap pointers to string
void swap_pointers(char **x, char **y)
{
    char *tmp;
    tmp = *x;
    printf("\n swap_pointers: x, y is  %s,%s tmp=x:%d \n", *x, *y, *tmp);
    *x = *y;
   *y = tmp;
    printf("\nswap_pointers :  x, y is %s %s \n", *x, *y);
}

int test_swap_char()
{
    char arr1[MAX]="ABC";
    char arr2[MAX] = "DEFGH";

    char *s1 = arr1;
    char *s2 = arr2;
    printf("\n s1, s2 address bef  is %p %p", s1, s2);

    //swap the pointers
    swap_pointers(&s1,&s2);
    printf("\n s1, s2 address aft is %p %p", s1, s2);

    return 0;
}
```

Here initially there is no change in address because the values are only passed and changed not the address. To swap the address we need to pass by reference for that we need to take a double pointer to catch the address, then we get the required result

```
a is 4
b is 3

 s1, s2 address bef  is 0x7fffabf6a1a0 0x7fffabf6a1f0
 swap_pointers: x, y is  ABC,DEFGH tmp=x:65

swap_pointers :  x, y is DEFGH ABC

 s1, s2 address aft is 0x7fffabf6a1f0 0x7fffabf6a1a0
```