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)

user60@trainux01: ~/Batch17OCT2024_175/Assignments/Day07/Introduction_to_Pointers_Assignments

```c
#include <stdio.h>

void display(int *ptr) {
    printf("Address: 0x%x\n", ptr);
    printf("Content: %s\n", ptr);
}
int main() {
    char arr[] = "hello hi ";
    int *ptr = (int *)arr;
    printf("sizeof ptr: %d, arr: %d\n", sizeof(ptr), sizeof(arr));
    display(ptr);
    return 0;
}
```

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]

user60@trainux01: ~/Batch17OCT2024_175/Assignments/Day07/Introduction_to_Pointers_Assignments

```c
#include <stdio.h>
#include <string.h>
#define MAX 100
#define SUCCESS 0
#define FAILURE 1

int append(char *ptr, char *appendstr) {
    while (*ptr) {
        ptr++;
    }
    while (*appendstr) {
        *ptr = *appendstr;
        ptr++;
        appendstr++;
    }
    *ptr = '\0';
    return SUCCESS;
}
void display(char *ptr) {
    printf("Address: 0x%x\n", ptr);
    printf("Content: %s\n", ptr);
}
int main() {
    char arr[MAX] = "Learning C";
    char *ptr = arr;
    char appendstr[3] = "in my org";
    printf("Address of ptr: 0x%x\n", (unsigned int)ptr);
    int ret = append(ptr, appendstr);
    printf("Address of ptr: 0x%x\n", (unsigned int)ptr);
    if (ret == SUCCESS) {
        display(ptr);
    }
    return 0;
}

~
```

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.

user60@trainux01: ~/Batch17OCT2024_175/Assignments/Day07/Introduction_to_Pointers_Assignments

```c
#include <stdio.h>

void swap_nums(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
    printf("In swap_nums, a: %d, b: %d\n", a, b);
}
void swap_pointers(int **a, int **b) {
    int *temp = *a;
    *a = *b;
    *b = temp;
}
int main() {
    int x = 10, y = 20;
    int *px = &x, *py = &y;
    printf("Before swap_nums: x = %d, y = %d\n", x, y);
    swap_nums(x, y);
    printf("After swap_nums: x = %d, y = %d\n", x, y);
    printf("Before swap_pointers: *px = %d, *py = %d\n", *px, *py);
    swap_pointers(&px, &py);
    printf("After swap_pointers: *px = %d, *py = %d\n", *px, *py);
    return 0;
}

```