Assignment 04 pointer

Problem 1:

Write a C program to input elements in an array and print the elements and reverse array's elements using pointers.

```
Enter size of Array: 10
Enter the elements in array: 1 2 3 4 5 6 7 8 9 10
Array elements: 1 2 3 4 5 6 7 8 9 10
The elements of array in reverse order are: 10 9 8 7 6 5 4 3 2 1
```

```c
#include<stdio.h>
int i,size;
int*S=NULL;
int*ptr;
int main(void)
{
S=&size;
printf("Enter size of Array : ",S);
scanf(" %d",S);
int arr[*S];
ptr=arr;
printf("Enter the elements of Array : ");
for (i=0;i<(*S);i++)
{
    scanf(" %d",&ptr[i]);
}
printf("Array Elements : ");
for (i=0;i<(*S);i++)
{
 printf(" %d ",ptr[i]); //or  *(ptr+i)
}
printf("\nThe Elements of Array in reverse order: ");
for (i=(*S)-1;i>-1;i--)
{
 printf(" %d ",*(ptr+i));
}

   return 0;
}
```

```
C:\Users\georg\OneDrive\Desktop\embedde\embedded\yhyy\bin\Debug\yhyy.exe
Enter size of Array : 10
Enter the elements of Array : 1 2 3 4 5 6 7 8 9 10
Array Elements :  1  2  3  4  5  6  7  8  9  10
The Elements of Array in reverse order:  10  9  8  7  6  5  4  3  2  1
Process returned 0 (0x0)   execution time : 8.158 s
```

Problem 2:

Write a C program to find reverse of a given string using pointers.

```
Enter any string: Mohammed

Original string: Mohammed
Reverse string: demmahoM
```

```c
#include<stdio.h>
char*ptr;
int size=10;
int main(void)
{
    char arr[size];
    ptr=arr;
    printf("Enter Any String :");
    scanf("%s", ptr);
    printf("Original String :");
    for(int i = 0; i < size && ptr[i] != '\0'; i++)
    {
        printf("%c", *(ptr+i));
    }

    int length = 0;
    while(ptr[length] != '\0')
    {
        length++;
    }

    printf("\nReverse String :");
    for(int i = length - 1; i >= 0; i--)
    {
        printf("%c", *(ptr+i));
    }

   return 0;
}
```

```
Enter Any String :mohammed
Original String :mohammed
Reverse String :demmahom
Process returned 0 (0x0)   execution time : 4.514 s
Press any key to continue.
```

Problem 3:

Write a C program to calculate the length of a string using a pointer.

```c
#include<stdio.h>
char*ptr;
int size=10;
int main(void)
{
    char arr[size];
    ptr=arr;
    printf("Enter Any String :");
    scanf("%s", ptr);
    printf("Original String :");
    for(int i = 0; i < size && ptr[i] != '\0'; i++)
    {
        printf("%c", *(ptr+i));
    }
    char length = 0;
    while(ptr[length] != '\0')
        length++;
    // size of actual array
    char*Start=&arr[0];
    char*End=&arr[length];
    printf("\nSize of  String :%d",End-Start);
    return 0;
}

```

```
Enter Any String :ahmed
Original String :ahmed
Size of  String :5
Process returned 0 (0x0)   execution time : 5.970 s
Press any key to continue.
```

Problem 4:

Write a C program to Swap elements using call byreference.

```
Input the value of 1st element : 5
Input the value of 2nd element : 6
Input the value of 3rd element : 7

The value before swapping are :
element 1 = 5
element 2 = 6
element 3 = 7

The value after swapping are :
element 1 = 7
element 2 = 5
element 3 = 6
```

```c
#include<stdio.h>
int ele_1=5;
int ele_2=6;
int ele_3=7;
void Swap(int*ele_1,int*ele_2,int*ele_3)
{
 int Swap=*ele_1;
 *ele_1=*ele_3;
 *ele_3=*ele_2;
 *ele_2=Swap;
}
int main(void)
{
printf("\nInput value of 1st element :%d",ele_1);
printf("\nInput value of 2nd element :%d",ele_2);
printf("\nInput value of 3rd element :%d\n",ele_3);

printf("\nThe Value before swapping: \n");

printf("\n  element 1:%d",ele_1);
printf("\n  element 2:%d",ele_2);
printf("\n  element 3:%d\n",ele_3);

printf("\nThe Value after swapping: \n");
Swap(&ele_1,&ele_2,&ele_3);
printf("\n  element 1:%d",ele_1);
printf("\n  element 2:%d",ele_2);
printf("\n  element 3:%d",ele_3);

    return 0;
}

```

```
Input value of 1st element :5
Input value of 2nd element :6
Input value of 3rd element :7

The Value before swapping:

  element 1:5
  element 2:6
  element 3:7

The Value after swapping:

  element 1:7
  element 2:5
  element 3:6
Process returned 0 (0x0)   execution time : 0.528 s
Press any key to continue.
```

Problem 5:

Write a C program *to* access members of a structureusing pointers.

```
Enter name: Mohammed
Enter age: 22
Enter ID: 15
====================
Displaying:
name: Mohammed
Age: 22
ID: 15
```

```c
#include<stdio.h>

typedef struct
{ char Name[15];
  int Age;
  int ID;
}student_t;

student_t *ptr ,Sstudent;

int main(void)
{
 ptr=&Sstudent;
 printf("Enter name: ");
 scanf("%s",&ptr->Name);
 printf("Enter Age: ");
 scanf("%d",&ptr->Age);
 printf("Enter ID: ");
 scanf("%d",&ptr->ID);

printf("\n================\n");

printf("Displaying\n");

printf("Enter name: %s \n",ptr->Name);

 printf("Enter Age: %d\n",ptr->Age );

 printf("Enter ID: %d\n",ptr->ID );


    return 0; }

3
```

```
Enter name: Mohammed
Enter Age: 22
Enter ID: 15

================
Displaying
Enter name: Mohammed
Enter Age: 22
Enter ID: 15

Process returned 0 (0x0)   execution time : 14.305 s
Press any key to continue.
```