ASSIGNMENT

Lecture 8

C-Programming (Pointers)
ENG/Mostafa Hassan Aboshaker

▪ Q1:

Write a program in C to demonstrate how to handle the pointers in the program.

```c
int main() {
    int num = 10;
    int *ptr=&num;

    printf("Value of 'num': %d\n", num);
    printf("Address of 'num': %p\n", &num);

    printf("\nUsing Pointers:\n");
    printf("Value at the address stored in 'ptr': %d\n", *ptr);
    printf("Address stored in 'ptr': %p\n", ptr);

    *ptr = 20;
    printf("\nAfter Modification:\n");
    printf("Value of 'num': %d\n", num);

    return 0;
}
```

▪ Q2:

Write a program in C to print all the alphabets using a pointer.

```c
#include <stdio.h>

int main() {
    char *ptr;
    char alphabet = 'A';

    ptr = &alphabet;

    printf("Alphabets:\n");

    for (int i = 0; i < 26; ++i) {
        printf("%c ", *ptr);
        (*ptr)++;
    }
    printf("\n");

    return 0;
}
```

■ Q3:

Write a program in C to print a string in reverse using a pointer

```c
#include <stdio.h>
#include <string.h>
int main()
{
char str1[50];
char revstr[50];
char *stptr=str1;
char *rvptr=revstr;
int i=-1;
printf("\n\n Pointer Print a string in reverse order : \n");
printf(" Input a string: ");
scanf("%s", str1);

while(*stptr)
{
stptr++;
i++;
}
while(i>=0)
{
    stptr--;
    *rvptr=*stptr;
    rvptr++;
    --i;
}
*rvptr='\0';
printf("The reverse of the string is: %s \n",revstr);
return 0;
}
```

■ Q4:

Write a program in C to print the elements of an array in reverse order.

```c
#include <stdio.h>

int main() {
    int arr[100], size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; ++i) {
        printf("Element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    printf("Elements in reverse order: ");
    int *ptr = arr + size - 1;

    while (ptr >= arr) {
        printf("%d ", *ptr);
        ptr--;
    }

    printf("\n");

    return 0;
}
```

- Q5:

Write a program in C to show a pointer to an array which contents are pointer to structure.

```c
#include <stdio.h>

struct Person {
    char name[50];
    int age;
};

int main() {

    struct Person people[3];

    struct Person (*ptrToPeople)[3] = &people;

    for (int i = 0; i < 3; ++i) {
        printf("Enter name for person %d: ", i + 1);
        scanf("%s", (*ptrToPeople)[i].name);

        printf("Enter age for person %d: ", i + 1);
        scanf("%d", &(*ptrToPeople)[i].age);
    }

    printf("\nDisplaying Information using Pointer to Array of Pointers to Structures:\n");
    for (int i = 0; i < 3; ++i) {
        printf("Person %d\n", i + 1);
        printf("Name: %s\n", (*ptrToPeople)[i].name);
        printf("Age: %d\n", (*ptrToPeople)[i].age);
        printf("\n");
    }

    return 0;
}
```