POINTERS

SARTHAK SANAY

(1) AIM:-

To write programs in C to demonstrate the usage of pointers (pointer arithmetic, passing pointers to functions).

CODE 1:- (Pointer Arithmetic)

```c
// Program in C to demonstrate Pointer Arithmetic
#include <stdio.h>
int main()
{
    // Pointer Increment and Decrement
    printf("Pointer Increment (char data type):-\n");
    char c= 'A';
    char *cp= &c;
    printf("cp = %d\n", cp);
    cp++;
    printf("cp++ = %d\n", cp);

    // Addition of Integer to Pointer
    printf("\nAddition of Integer to Pointer:-\n");
    int a= 22;
    int *p = &a;
    printf("Pointer before Addition:  %u\n", p);
    p= p+250;
    printf("Pointer after Addition:  %u\n", p);

    // Addition of two pointers
    printf("\nAddition of two pointers:-\n");
    int n1= 100, n2= 200, sumOfPtr;
    int *ptr1= &n1, *ptr2= &n2;
    sumOfPtr= *ptr1 + *ptr2;
    printf("ptr1 = %d\n", ptr1);
    printf("ptr2 = %d\n", ptr2);
```

```
    printf("ptr1 + ptr2 = %d\n", sumOfPtr); // 4 is converted into 1, i.e why
     output given is 300

    // Comparision of two pointers
    printf("\nComparision of two pointers:-\n");
    if(ptr1 == ptr2)
        printf("Pointer 1 and Pointer 2 are equal.");
    else
        printf("Pointer 1 and Pointer 2 are not equal.");

    return 0;
}
```

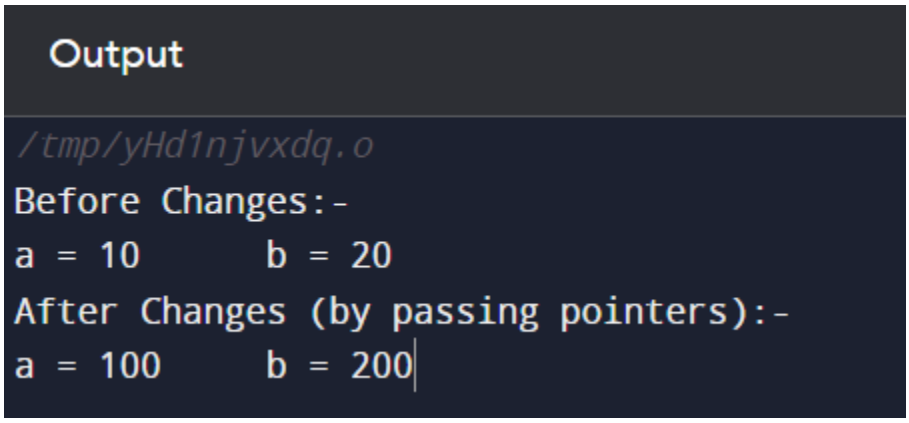
OUTPUT SCREEN 1:-

```
Output

/tmp/yHd1njvxdq.o
Pointer Increment (char data type):-
cp = -1336409269
cp++ = -1336409268

Addition of Integer to Pointer:-
Pointer before Addition:  2958558020
Pointer after Addition:  2958559020

Addition of two pointers:-
ptr1 = -1336409280
ptr2 = -1336409284
ptr1 + ptr2 = 300

Comparision of two pointers:-
Pointer 1 and Pointer 2 are not equal.
```

CODE 2:- (Passing Pointers to Functions)

```c
// Program in C to demonstrate the passing of pointers to functions
#include <stdio.h>
int swap(int *a, int *b)
{
    *a = 100;
    *b = 200;
}
int main()
{
    int a=10, b=20;
    printf("Before Changes:-\na = %d\t\tb = %d\n",a, b);
    swap(&a, &b);
    printf("After Changes (by passing pointers):-\na = %d\t\tb = %d",a, b);
    return 0;
}
```

OUTPUT SCREEN 2:-



(2) AIM:-

To write programs in C by implementing functions to swap values using pointers and without using pointers.

CODE 1:- (Using pointers)

```c
// Program in C to create a function to swap two values by passing pointers
#include <stdio.h>
void swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
int main()
{
    int a=10, b=20;
    printf("Original Values:-\na = %d\t\tb = %d\n",a, b);
    swap(&a, &b);
    printf("After Swapping:-\na = %d\t\tb = %d",a, b);
    return 0;
}
```

OUTPUT SCREEN 1:-

```
Original Values:-
a = 10      b = 20
After Swapping:-
a = 20      b = 10
```

CODE 2:- (Without using pointers)

```c
// Program in C to demonstrate swapping of two variables without passing pointers
#include<stdio.h>
void swap(int a, int b)
{
    int temp;
    temp= a;
    a= b;
    b= temp;
    printf("After Swapping:-\na = %d\t\tb = %d",a, b);
}
int main()
{
    int a=10, b=20;
    printf("Original Values:-\na = %d\t\tb = %d\n",a, b);
    swap(a,b);
    return 0;
}
```

OUTPUT SCREEN 2:-

```
Original Values:-
a = 10      b = 20
After Swapping:-
a = 20      b = 10
```

(3) AIM:-

To write a program in C to find the sum of elements in an array using pointers.

CODE :-

```c
#include <stdio.h>
int main()
{
    int n, sum=0;
    printf("Enter size of array: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter Array elements:- \n");
    for(int i=0; i<n; i++)
    {
        printf("Enter element %d: ", i);
        scanf("%d", &arr[i]);
    }
    int *ptr= arr; // Setting address of first array element to *ptr
    for(int i=0; i<n; i++)
    {
        sum= sum + *ptr;
        ptr++; // Incrementing pointer by one to get next element
    }
    printf("The sum of elements of Array is:  %d", sum);
    return 0;
}
```

OUTPUT SCREEN :-