**EX-5 (Theory Part)**

**Pulkit Saxena**

Q1: **Pointers in c ?**

* Pointers are variables whose values are memory addresses.
* Int a=5;

Int \*p=&a;

Printf(“%d”,\*p); //gives output as 5

* Pointers enable programs to simulate call-by-reference and to create and manipulate dynamic data structures, i.e., data structures that can grow and shrink at execution time, such as linked lists, queues, stacks and trees.

Q2:  **What is the difference between double pointer and pointer to pointer?**

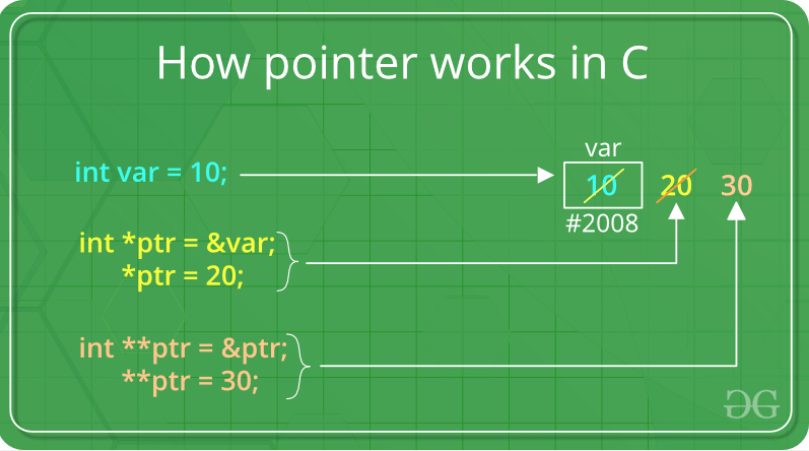
* A pointer is used to store the address of variables. So, when we define a pointer to pointer, the first pointer is used to store the address of the second pointer. Thus it is known as double pointers.
* Int a=10;

Int \*p=&a;

Int \*\*p=&p;

Printf(“%d”,\*p); //gives output 10 and 10

Printf(“%d”,\*\*p);



Q3: **What is a void pointer?**

* Void pointer is a generic pointer that can represent any pointer type.
* All pointer types can be assigned a pointer to void, and a pointer to void can be assigned a pointer of any type.
* In both cases, a cast operation is not required.
* A pointer to void cannot be dereferenced.
* Int a=10;

Char b=’h’;

Void \*p=&a;

Printf(“%d”,\*p); //prints 10

\*p=b;

Printf(“%d”,\*p); // prints h

Q4: **Why does C treat array parameters as pointers?**

* Since pointer contains memory address and arrays are the collection of consecutive memory address.So array and pointers are used interchangeably.
* C treats array parameter as pointers because it is less time consuming and more efficient. Though if we can pass the address of each element of the array to a function as argument but it will be more time consuming

Q5: **What is a function pointer and how to use it?**

* A pointer to a function contains the address of the function in memory.
* we saw that an array name is really the address in memory of the first element of the array. Similarly, a function name is really the starting address in memory of the code that performs the function’s task.
* A function's name can also be used to get functions' address.

void fun(int a)

{

    printf("Value of a is %d\n", a);

}

int main()

{

    void (\*fun\_ptr)(int) = fun;  // & removed

    fun\_ptr(10);  // \* removed

    return 0;

}

Q6**: How to pass a double array to a function? At Least 2 different ways and what is the difference in both of them?**

* Using Single pointer

Int main(void)

{

Int m=3,n=3;

Int arr[m][n]={{1,2,3},{4,5,6},{7,8,9}};

Printfunc(\*arr,m,n);

Return 0;

}

Void printfunc(int \*arr,m,n)

{

Int I,j;

For(int i=0,i<m;i++)

For(int j=0;j<m;j++)

Printf(“%d ”,\*((arr+i\*n)+j); Output 1 2 3 4 5 6 7 8 9

}

* Using Arrays

int main()

{

    int arr[][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

    printfunc(arr);

    return 0;

}

void printfunc(int arr[3][3])

{

    int i, j;

    for (i = 0; i < 3; i++)

      for (j = 0; j < 3; j++)

        printf("%d ", arr[i][j]); Output 1 2 3 4 5 6 7 8 9

}

Q7: **What is a pointer to an array? How to access elements of a double array by pointers?**

Int main(void)

{

Int m=3,n=3;

Int arr[m][n]={{1,2,3},{4,5,6},{7,8,9}};

Printfunc(\*arr,m,n);

Return 0;

}

Void printfunc(int \*arr,m,n)

{

Int I,j;

For(int i=0,i<m;i++)

For(int j=0;j<m;j++)

Printf(“%d ”,\*((arr+i\*n)+j); Output 1 2 3 4 5 6 7 8 9

}

Q8: **How to print the address of a variable?**

Void main()

{

Int a=5;

Int \*P;

P=&a;

printf(“%x”,P); %print address of a variable

}

Q9: **What is the difference between NULL pointer and Void pointer?**

* pointer with the value NULL points to nothing. Initializing a pointer to 0 is equivalent to initializing a pointer to NULL. When 0 is assigned, it’s first converted to a pointer of the appropriate type.
* Void pointer is a generic pointer that can represent any pointer type. All pointer types can be assigned a pointer to void, and a pointer to void can be assigned a pointer of any type.