POINTERS IN C

PART II

FAQs

1. **What is the difference between malloc and calloc?**
   There are two major differences.
   First, is in the number of arguments. Malloc() takes a single argument (memory required in bytes), while calloc() needs two arguments.
   Secondly, malloc() does not initialize the memory allocated, while calloc() initializes the allocated memory to ZERO.
   calloc() allocates a memory area, the length will be the product of its parameters. calloc fills the memory with ZERO's and returns a pointer to first byte. If it fails to locate enough space it returns a NULL pointer.
   Syntax: ptr_var=(cast_type *)calloc(no_of_blocks , size_of_each_block); i.e. ptr_var=(type *)calloc(n,s);
   malloc() allocates a single block of memory of REQUSTED SIZE and returns a pointer to first byte. If it fails to locate requsted amount of memory it returns a null pointer.
   Syntax: ptr_var=(cast_type *)malloc(Size_in_bytes);
   The malloc() function take one argument, which is the number of bytes to allocate, while the calloc() function takes two arguments, one being the number of elements, and the other being the number of bytes to allocate for each of those elements. Also, calloc() initializes the allocated space to zeroes, while malloc() does not.

2. **What is meant by a pointer to a pointer?**
   Since we can have pointers to any data type in C, we can also have pointers to other pointers. For example a declaration of a pointer-to-pointer looks like
   int **ipp;
   where the two asterisks indicate that two levels of pointers are involved.

3. **Explain the use of const with pointers.**
   While we use the **const** keyword in pointers we have to distinguish between making the pointer itself constant and making the value is pointed to constant.
   For example, consider the following declarations;
   const float *ptr; or **float const *ptr;**
   Here ptr points to a constant float value. That is the ptr points to a value , that must remain constant. The value of ptr itself can be changed.
   float *const ptr;
   Here ptr is a const pointer. That is the pointer ptr cannot have its value changed. It must always point to the same address, but the pointed to value can change.

For example,
const float * const ptr;
means both that ptr must always point to the same location and that the value stored at location must not change.

4. **Explain pointer to functions.**
C provides a special feature of pointer to a function. As you know that every function defined in C language have a base address attached to it. This base address acts as an entering point into that function. This address can be stored in a pointer known as function pointer. The pointer to a function is the declaration of pointer that holds the base address of the function. The declaration of a pointer to a function is:
return_type (* pointer_name) (variable1_type variable1_name , variable2_type variable2_name , variable3_type variable3_name .....);

5. **What is the purpose of sizeof operator?**

C provides an operator sizeof to find the required space to store the data object of a particular data type. The format of a sizeof operator is
sizeof(data-type or variable);
For example sizeof(int) returns the size of memory to hold an int data type. The header file stdlib.h should be included at the beginning of the program while using the memory allocation functions.