C Programming Questions

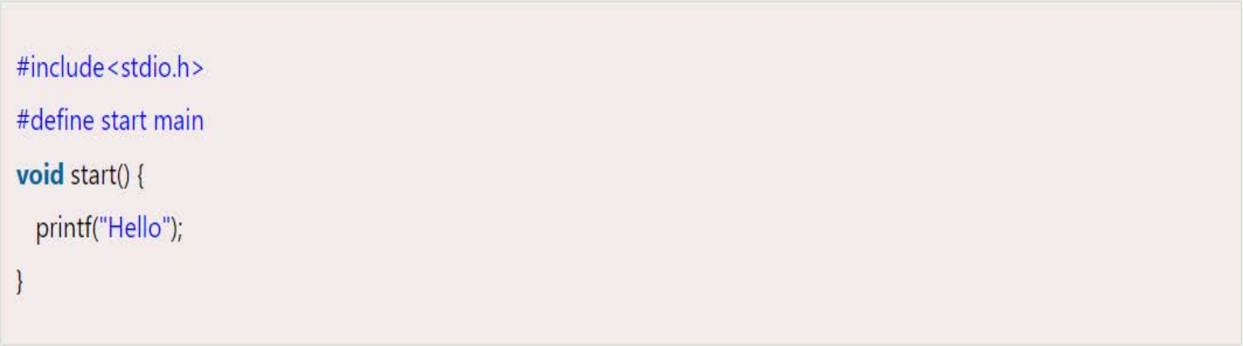
What is C language?

C is a mid-level and procedural programming language. The Procedural programming language is also known as the structured programming language is a technique in which large programs are broken down into smaller modules, and each module uses structured code.

Can we compile a program without main() function?

Yes, we can compile, but it can't be executed.

But, if we use #define, we can compile and run a C program without using the main() function.



```
#include<stdio.h>
#define start main
void start() {
  printf("Hello");
}
```

What is the difference between getch() and getche()?

The getch() function reads a single character from the keyboard. It doesn't use any buffer, so entered data will not be displayed on the output screen.

The getche() function reads a single character from the keyword, but data is displayed on the output screen. Press Alt+f5 to see the entered character.

What is the use of printf() and scanf() functions?

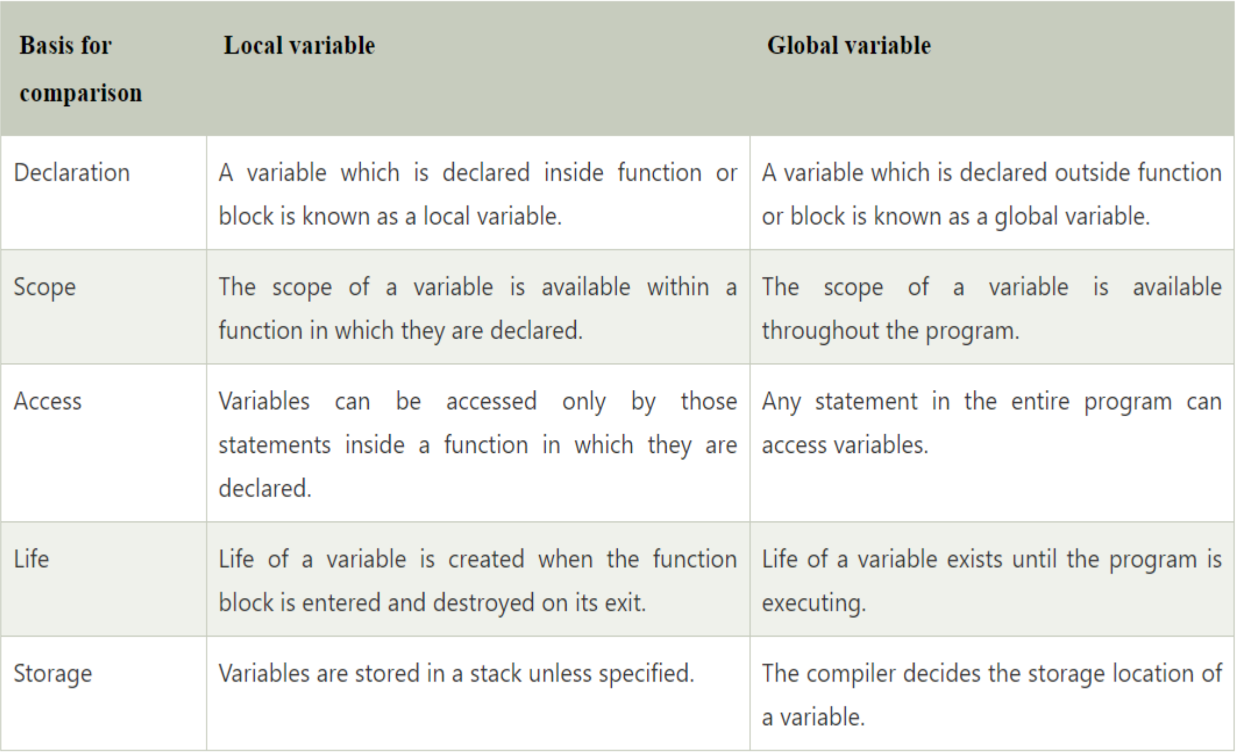
printf(): The printf() function is used to print the integer, character, float and string values on to the screen.

Following are the format specifier:

- %d: It is a format specifier used to print an integer value.
- %s: It is a format specifier used to print a string.
- %c: It is a format specifier used to display a character value.
- %f: It is a format specifier used to display a floating point value.

scanf(): The scanf() function is used to take input from the user.

What is the difference between the local variable and global variable in C?



Basis for comparison	**Local variable**	**Global variable**
Declaration	A variable which is declared inside function or block is known as a local variable.	A variable which is declared outside function or block is known as a global variable.
Scope	The scope of a variable is available within a function in which they are declared.	The scope of a variable is available throughout the program.
Access	Variables can be accessed only by those statements inside a function in which they are declared.	Any statement in the entire program can access variables.
Life	Life of a variable is created when the function block is entered and destroyed on its exit.	Life of a variable exists until the program is executing.
Storage	Variables are stored in a stack unless specified.	The compiler decides the storage location of a variable.

What is the use of a static variable in C?

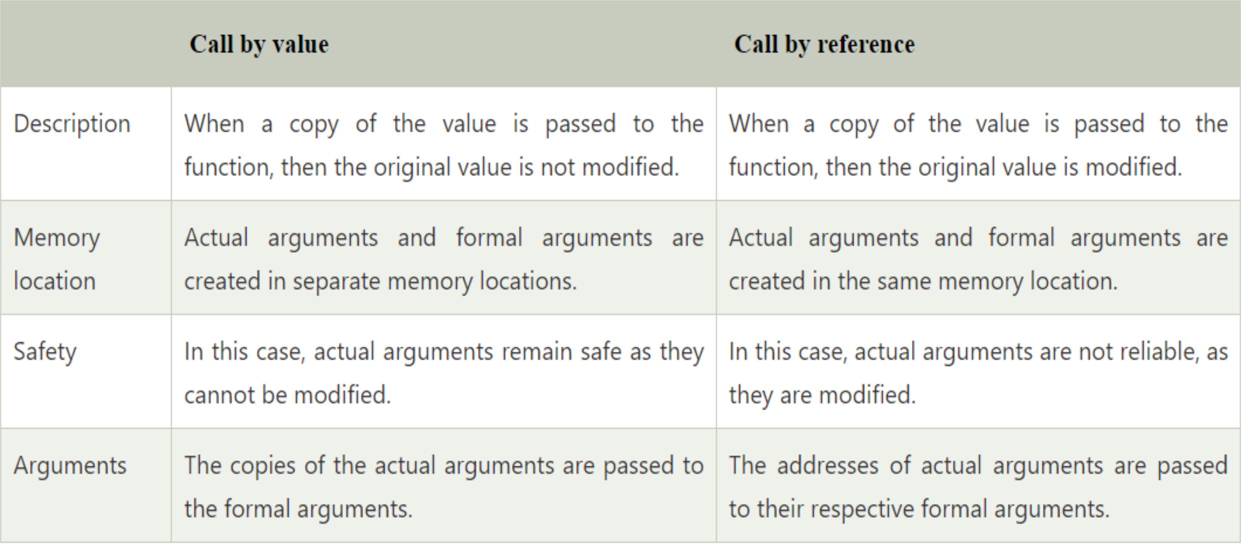
- A variable which is declared as static is known as a static variable. The static variable retains its value between multiple function calls.
- Static variables are used because the scope of the static variable is available in the entire program. So, we can access a static variable anywhere in the program.
- The static variable is initially initialized to zero. If we update the value of a variable, then the updated value is assigned.
- The static variable is used as a common value which is shared by all the methods.
- The static variable is initialized only once in the memory heap to reduce the memory usage.

What is the use of the function in C?

Uses of C function are:

- C functions are used to avoid rewriting the same code again and again in our program.
- C functions can be called any number of times from any place of our program.
- When a program is divided into functions, then any part of our program can easily be tracked.
- C functions provide the reusability concept, i.e., it breaks the big task into smaller tasks so that it makes the C program more understandable.

What is the difference between call by value and call by reference in C?



	Call by value	Call by reference
Description	When a copy of the value is passed to the function, then the original value is not modified.	When a copy of the value is passed to the function, then the original value is modified.
Memory location	Actual arguments and formal arguments are created in separate memory locations.	Actual arguments and formal arguments are created in the same memory location.
Safety	In this case, actual arguments remain safe as they cannot be modified.	In this case, actual arguments are not reliable, as they are modified.
Arguments	The copies of the actual arguments are passed to the formal arguments.	The addresses of actual arguments are passed to their respective formal arguments.

What is recursion in C?

When a function calls itself, this process is known as recursion. The function that is called itself is known as a recursive function.

Recursive function comes in two phases:

- Winding phase
- Unwinding phase

Winding phase: When the recursive function calls itself, and this phase ends when the condition is reached.

Unwinding phase: Unwinding phase starts when the condition is reached, and the control returns to the original call.

What is an array in C?

An Array is a group of similar types of elements. It has a contiguous memory location. It makes the code optimized, easy to traverse and easy to sort. The size and type of arrays cannot be changed after its declaration.

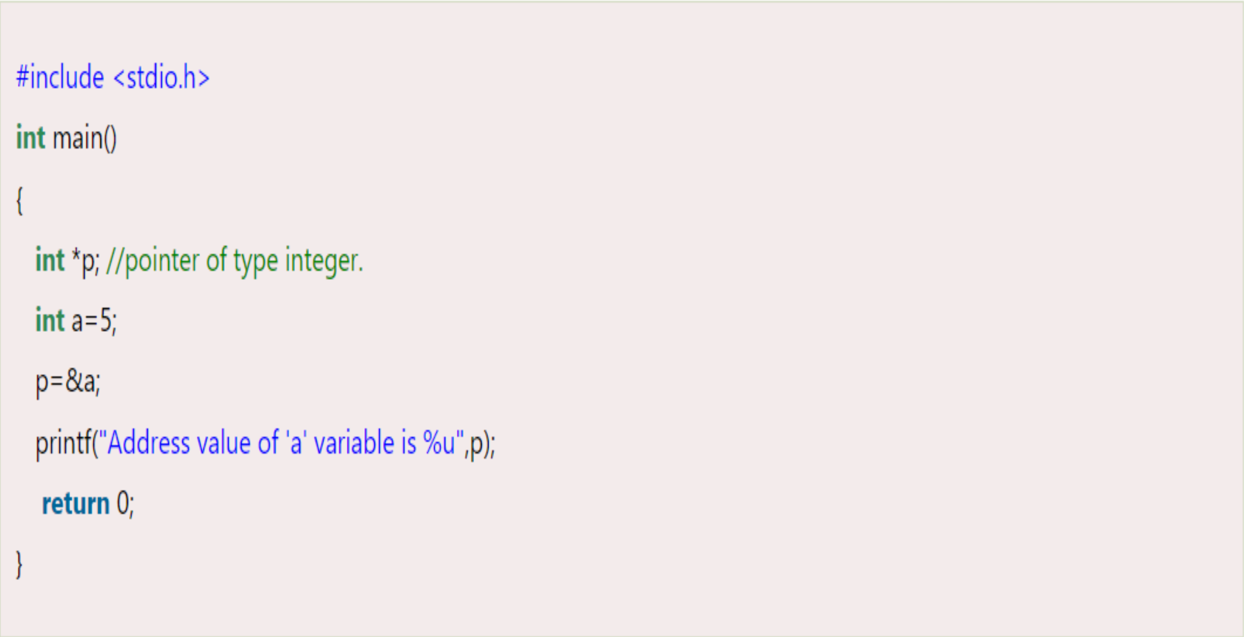
Arrays are of two types:

- **One-dimensional array**: One-dimensional array is an array that stores the elements one after the another.
- **Multidimensional array**: Multidimensional array is an array that contains more than one array.

What is a pointer in C?

A pointer is a variable that refers to the address of a value. It makes the code optimized and makes the performance fast. Whenever a variable is declared inside a program, then the system allocates some memory to a variable. The memory contains some address numbers. The variables that hold this address number is known as the pointer variable.

Example of pointer



```
#include <stdio.h>
int main()
{
  int *p; //pointer of type integer.
  int a=5;
  p=&a;
  printf("Address value of 'a' variable is %u",p);
   return 0;
}
```

What is the usage of the pointer in C?

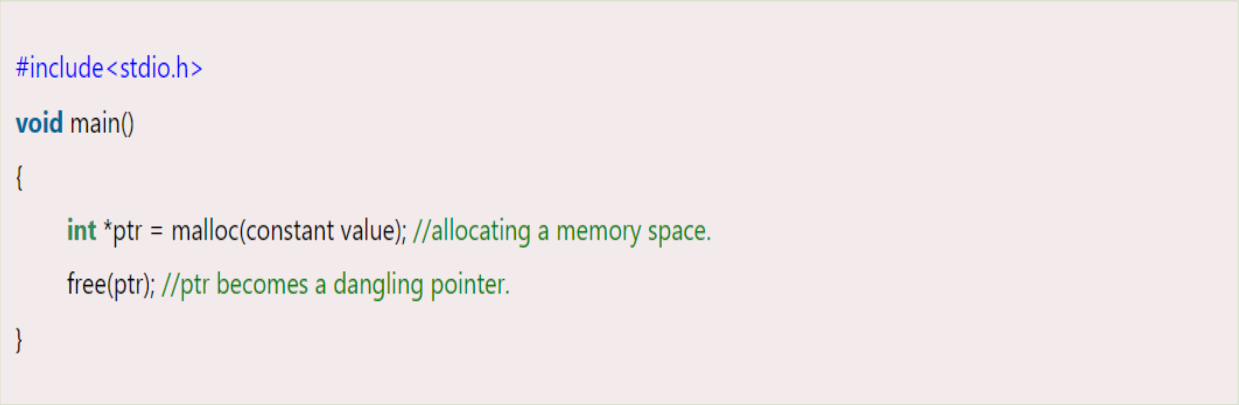
- **Accessing array elements**: Pointers are used in traversing through an array of integers and strings. The string is an array of characters which is terminated by a null character '\0'.
- **Dynamic memory allocation**: Pointers are used in allocation and deallocation of memory during the execution of a program.
- **Call by Reference**: The pointers are used to pass a reference of a variable to other function.
- **Data Structures like a tree, graph, linked list, etc.**: The pointers are used to construct different data structures like tree, graph, linked list, etc.

What is a far pointer in C?

A pointer which can access all the 16 segments (whole residence memory) of RAM is known as a far pointer. A far pointer is a 32-bit pointer that obtains information outside the memory in each section.

What is dangling pointer in C?

- If a pointer is pointing any memory location, but meanwhile another pointer deletes the memory occupied by the first pointer while the first pointer still points to that memory location, the first pointer will be known as a dangling pointer. This problem is known as a dangling pointer problem.
- Dangling pointer arises when an object is deleted without modifying the value of the pointer. The pointer points to the deallocated memory.

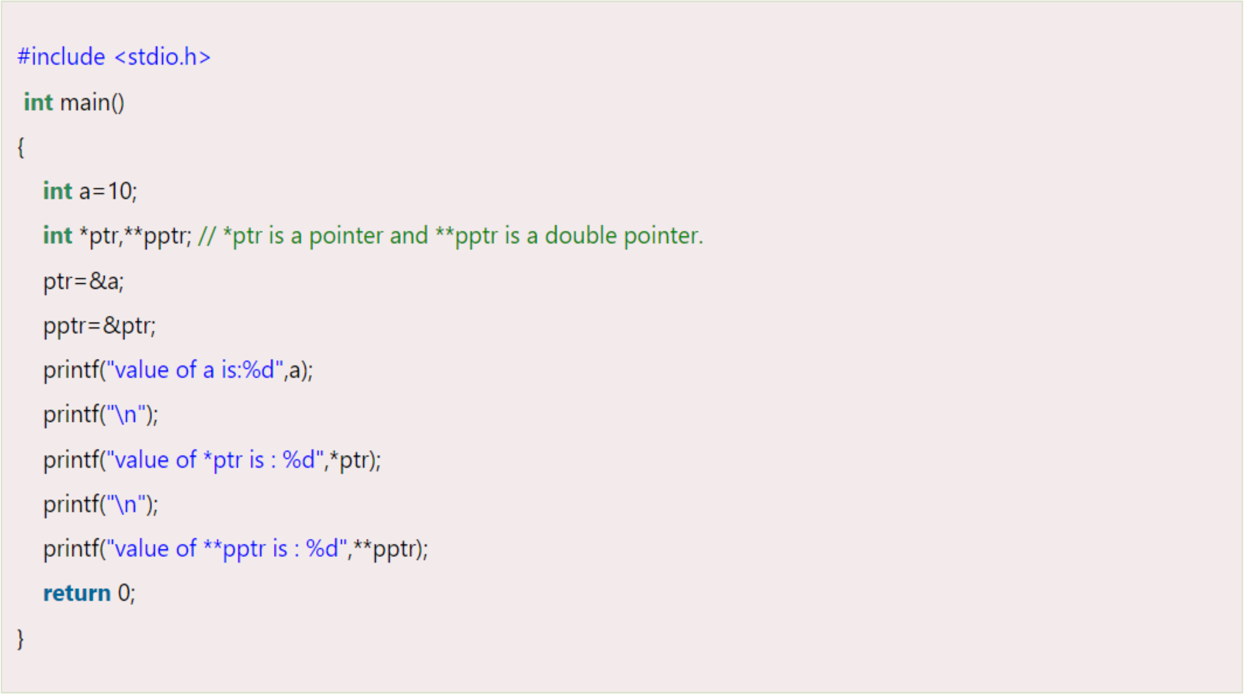


```
#include<stdio.h>
void main()
{
     int *ptr = malloc(constant value); //allocating a memory space.
     free(ptr); //ptr becomes a dangling pointer.
}
```

What is pointer to pointer in C?

In the case of a pointer-to-pointer concept, one pointer refers to the address of another pointer. The pointer to pointer is a chain of pointers. Generally, the pointer contains the address of a variable. The pointer to pointer contains the address of a first pointer.

Example



```
#include <stdio.h>
 int main()
{
   int a=10;
   int *ptr,**pptr; // *ptr is a pointer and **pptr is a double pointer.
   ptr=&a;
   pptr=&ptr;
   printf("value of a is:%d",a);
   printf("\n");
   printf("value of *ptr is : %d",*ptr);
   printf("\n");
   printf("value of **pptr is : %d",**pptr);
   return 0;
}
```

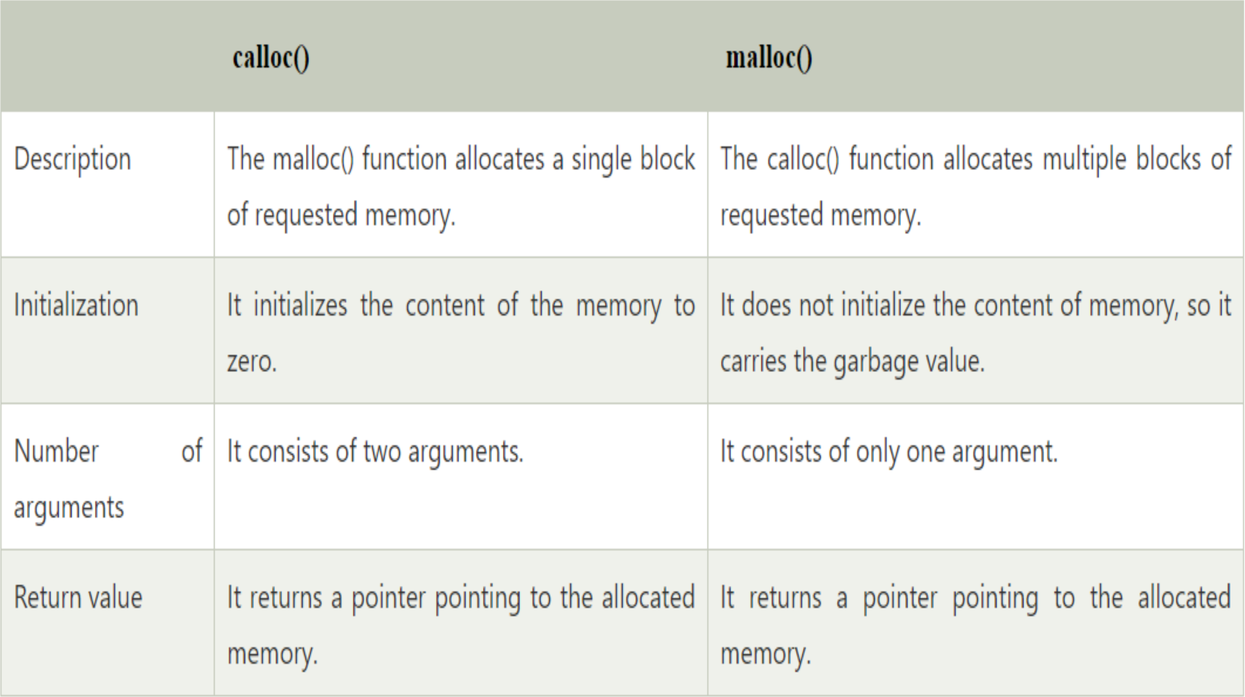
What is static memory allocation?

- In the case of static memory allocation, memory is allocated at compile time, and memory can't be increased while executing the program. It is used in the array.
- The lifetime of a variable in static memory is the lifetime of a program.
- The static memory is allocated using static keyword.
- Static memory is implemented using stacks or heap.
- The pointer is required to access the variable present in the static memory.
- Static memory is faster than dynamic memory.
- In static memory, more memory space is required to store the variable.

What is dynamic memory allocation?

- In the case of dynamic memory allocation, memory is allocated at runtime and memory can be increased while executing the program. It is used in the linked list.
- The malloc() or calloc() function is required to allocate the memory at the runtime.
- An allocation or deallocation of memory is done at the execution time of a program.
- No dynamic pointers are required to access the memory.
- The dynamic memory is implemented using data segments.
- Less memory space is required to store the variable.

What is the difference between malloc() and calloc()?

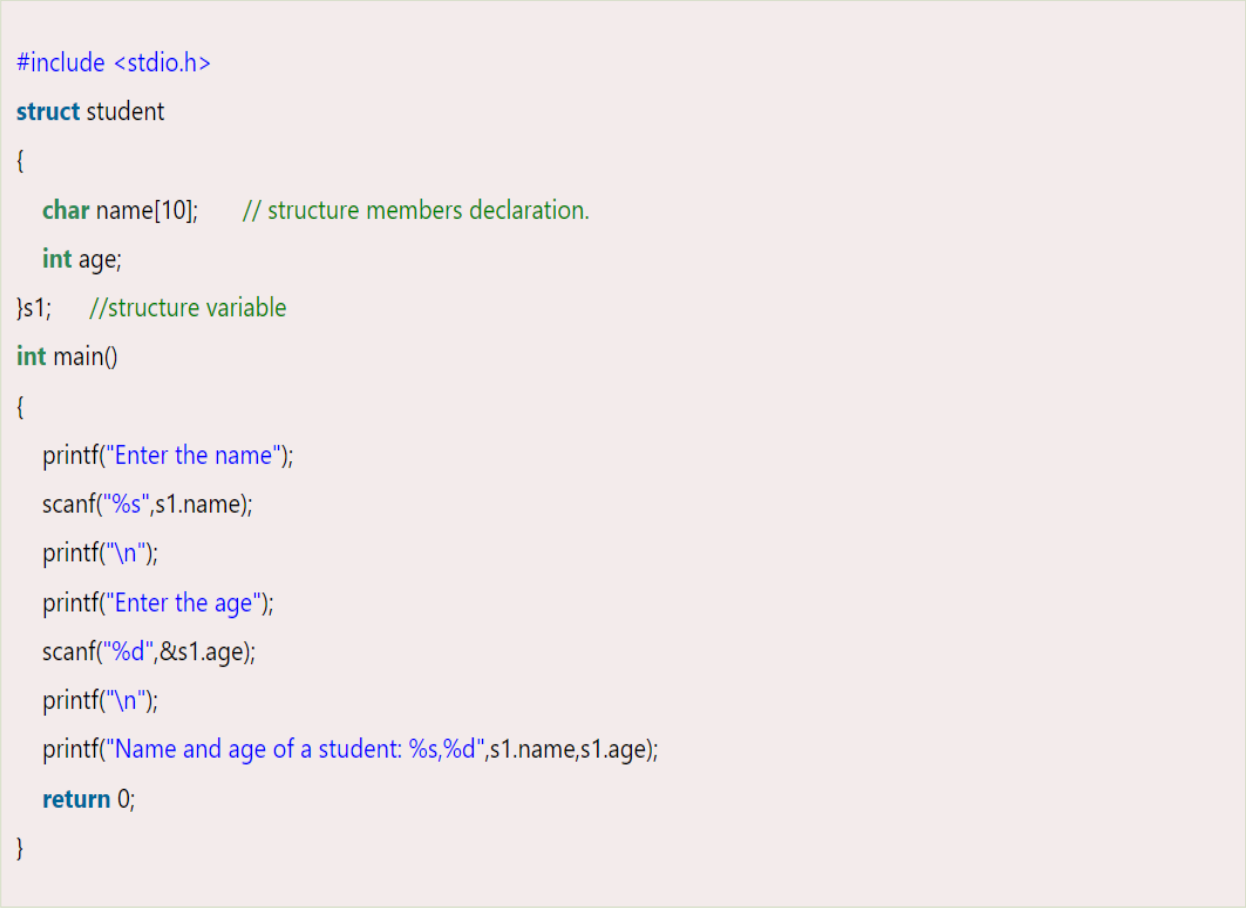


	calloc()	**malloc()**
Description	The malloc() function allocates a single block of requested memory.	The calloc() function allocates multiple blocks of requested memory.
Initialization	It initializes the content of the memory to zero.	It does not initialize the content of memory, so it carries the garbage value.
Number of arguments	It consists of two arguments.	It consists of only one argument.
Return value	It returns a pointer pointing to the allocated memory.	It returns a pointer pointing to the allocated memory.

What is the structure?

- The structure is a user-defined data type that allows storing multiple types of data in a single unit. It occupies the sum of the memory of all members.
- The structure members can be accessed only through structure variables.
- Structure variables accessing the same structure, but the memory allocated for each variable will be different.

Example

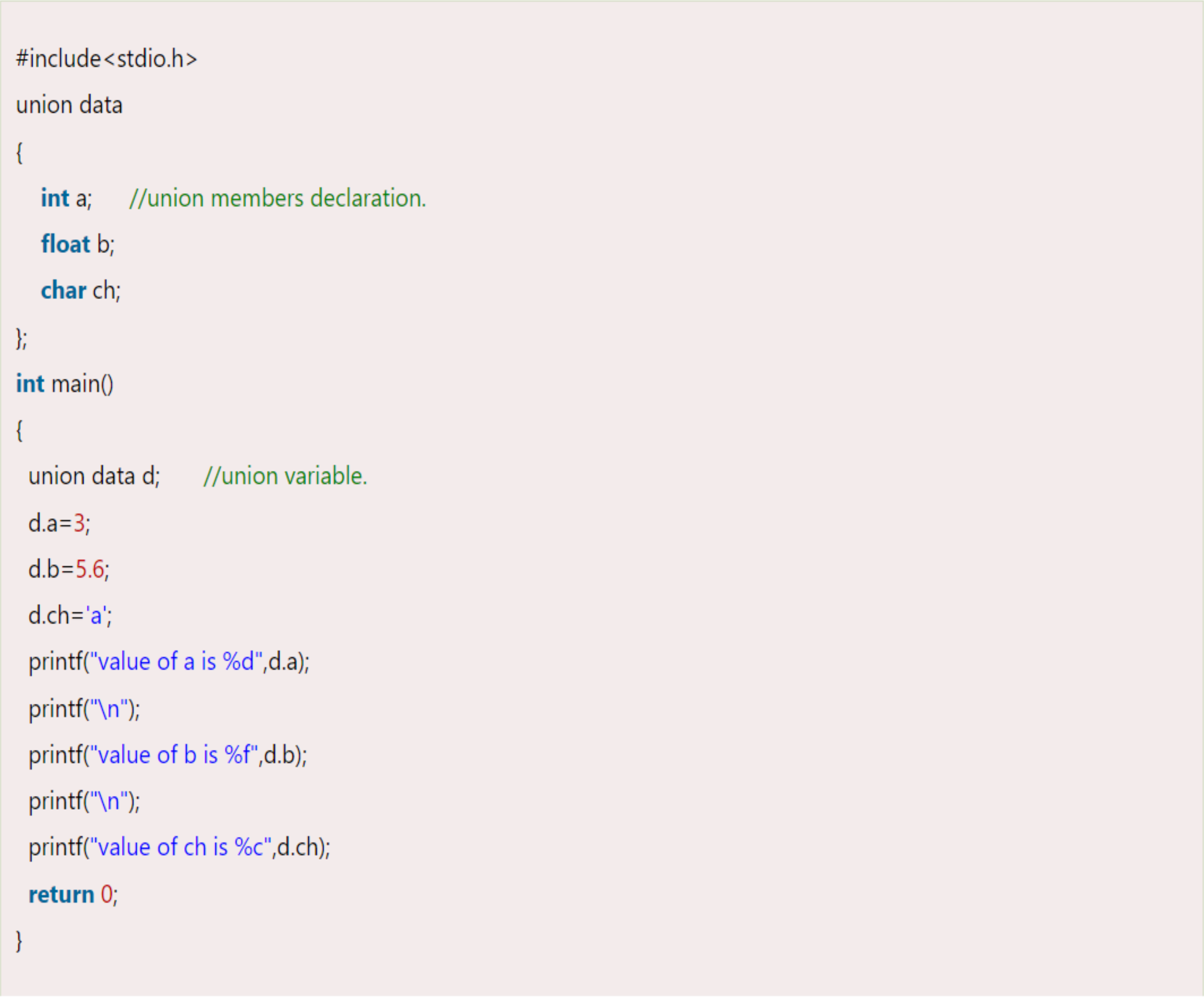


```
#include <stdio.h>
struct student
{
    char name[10];       // structure members declaration.
    int age;
}s1;      //structure variable
int main()
{
    printf("Enter the name");
    scanf("%s",s1.name);
    printf("\n");
    printf("Enter the age");
    scanf("%d",&s1.age);
    printf("\n");
    printf("Name and age of a student: %s,%d",s1.name,s1.age);
    return 0;
}
```

What is a union?

- The union is a user-defined data type that allows storing multiple types of data in a single unit. However, it doesn't occupy the sum of the memory of all members. It holds the memory of the largest member only.
- In union, we can access only one variable at a time as it allocates one common space for all the members of a union.

Example



```
#include<stdio.h>
union data
{
    int a;      //union members declaration.
    float b;
    char ch;
};
int main()
{
  union data d;       //union variable.
  d.a=3;
  d.b=5.6;
  d.ch='a';
  printf("value of a is %d",d.a);
  printf("\n");
  printf("value of b is %f",d.b);
  printf("\n");
  printf("value of ch is %c",d.ch);
  return 0;
}
```

What is the maximum length of an identifier?

It is **32 characters** ideally but implementation specific.

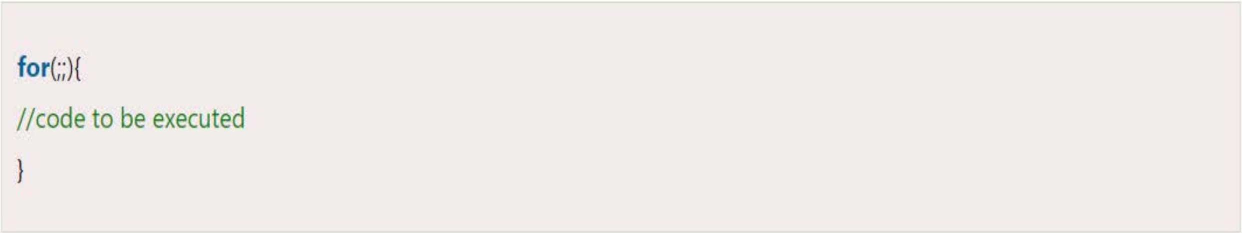
What are the functions to open and close the file in C language?

The ***fopen()* function** is used to open file whereas *fclose()* is used to close file.

What is an infinite loop?

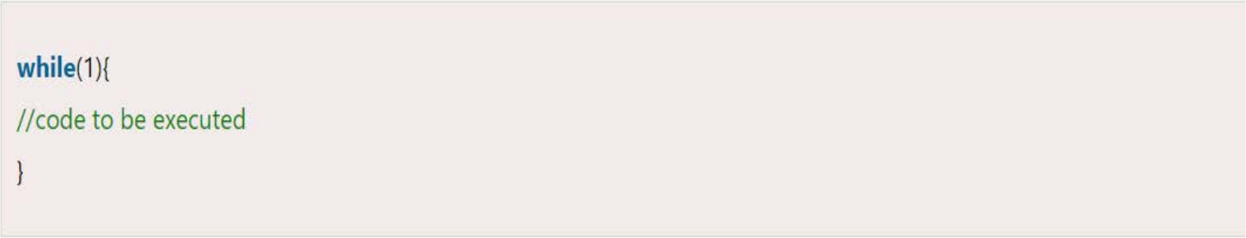
A loop running continuously for an indefinite number of times is called the infinite loop.

Infinite For Loop:



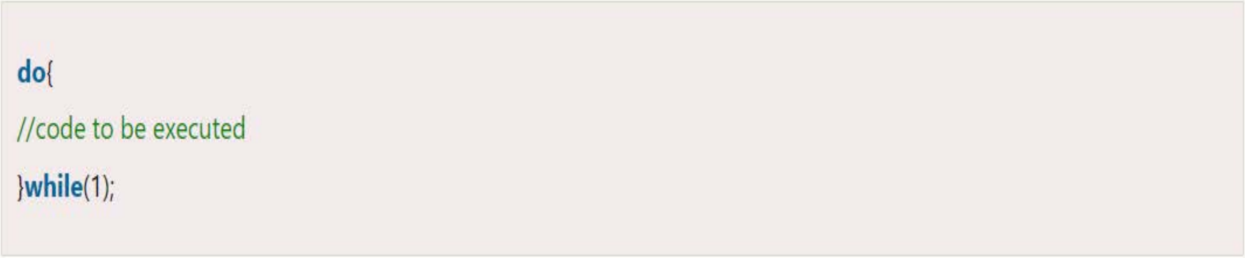
```
for(;;){
//code to be executed
}
```

Infinite While Loop:



```
while(1){
//code to be executed
}
```

Infinite Do-While Loop:



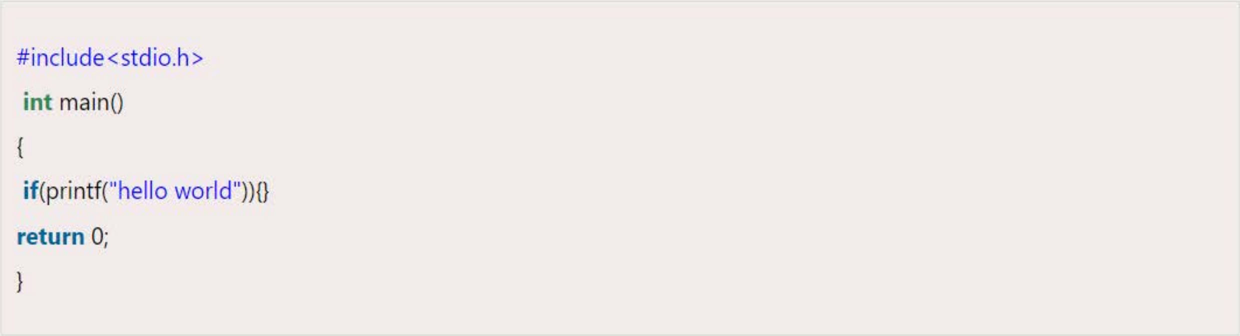
```
do{
//code to be executed
}while(1);
```

C Program to print "hello" without semicolon

We can print "hello" or "hello world" or anything else in C without using semicolons. There are ways to do:

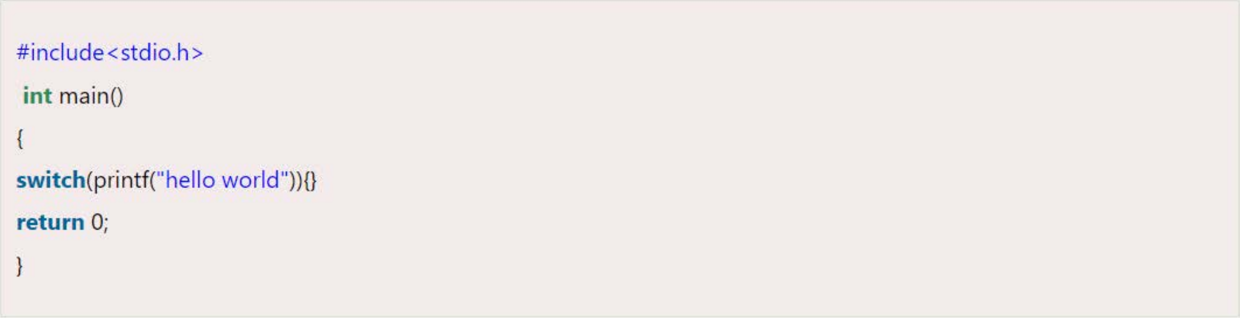
- Using if
- Using switch
- Using loop

Program 1: Using if statement.



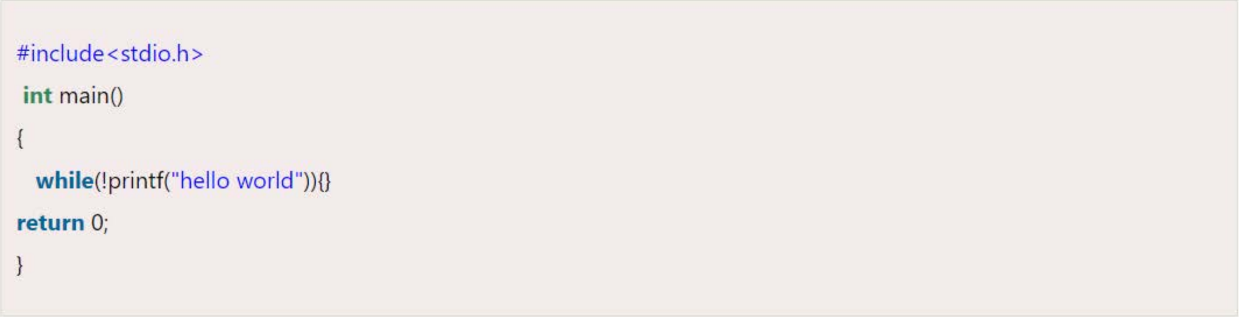
```
#include<stdio.h>
 int main()
{
 if(printf("hello world")){}
return 0;
}
```

Program 2: Using switch statement.



```
#include<stdio.h>
 int main()
{
switch(printf("hello world")){}
return 0;
}
```

Program 3: Using while loop.



```
#include<stdio.h>
 int main()
{
   while(!printf("hello world")){}
return 0;
}
```