What is Arrray and what kinds of Array?

Answer:- An array is a fixed-size sequenced collection of elements of the same data type.

Types of arrays:- There are three types of arrays:

1. One-dimensional arrays.
2. Two-dimensional arrays.
3. Multidimensional arrays.

Data structures:- C supports creation and manipulation of the following data structures:

1. Linked Lists
2. Stacks
3. Queues
4. Trees
5. Graphs

Initialization of one-dimensional arrays:-

Compile time initialization:- We can initialize the elements of arrays in the same way as the ordinary variables when they are declared. The general form of initialization of arrays is:

type array_name[size] = {list of values};

The values in the list are separated by commas. For example, the statement

int number[3] = {0, 0, 0};

will declare the variable number as an array of size 3 and will assign zero to each element.

If the number of values in the list is less than the number of elements, then only that many elements will be initialized. The remaining elements will be set to zero automatically. Example,

float total[5] = {0.0, 15.75, -10};

will initialize the first three elements to 0.0, 15.75 and -10.0 and the remaining two elements to zero.

In such cases, the compiler allocates enough space for all initialized elements. For example, the statement

```
int counter[ ] = {1,1,1,1};
```

will declare the counter array to contain four elements with initial values 1.

however, If we have more initializers than the declared size, the compiler will produce an error. That is the statement

```
int number[3] = {10,20,30,40};
```

will not work, It is illegal in C.

Run time initialization:-

An array can be explicitly initialized at run time. This approach is usually applied for initializing large arrays. For example, consider the following segment of a C program.

```
for (i=0; i<100; i=i+1)
 {
   if (i<50)
      sum[i]=0.0;
    else
      sum[i] = 1.0;
   }
```

The first 50 elements of the array sum are initialized to zero while the remaining 50 elements are initialized to 1.0 at run time.

We can also use a read function such as scanf to initialize an array. For example, the statements

```
int x[3];
scanf("%d %d %d", &x[0], &x[1], &x[2]);
```

will initialize array elements with the values entered through the keyboard, at runtime.

Initializing two-dimensional arrays:- Two-dimensional arrays may be initialized by following their declaration with a list of initial values enclosed in braces. For example,

```
int table [2] [3] = {0,0,0,1,1,1};
```

initializes the elements of the first row to zero and the second row to one. The initialization is done row by row. The above statement can be equivalently written as

```
int table [2] [3] = {{0,0,0}, {1,1,1}};
```

by surrounding the elements of the each row by braces we can also initialize a two-dimensional array in the form of a matrix as shown below:

```
int table [2] [3] = {
                      {0,0,0},
                      {1,1,1}
                    };
```

When the array is completely initialized with all values, explicitly, we need not specify the size of the first dimension. That is, the statement

```
int table [ ] [3] = {
                      {0,0,0},
                      {1,1,1}
                    };
```

is permitted.

If the values are missing in an initializer, they are automatically set to zero. For instance, the statement

```
int table [2] [3] = {
                      {1,1},
                      {2}
                    };
```

will initialize the first two elements of the first row to one, the first ~~row~~ element of the second row to two, and all other elements to zero.

When all the elements are to be initialized to zero, the following short-cut method may be used.

```
int ... [3] [5] = {{0},{0},{0}};
```

What do you mean by Dynamic arrays:- In C, it is possible to allocate memory to arrays at run time. This feature is known as dynamic memory allocation and the arrays created at run time are called dynamic arrays.

Character Arrays and strings

What is string and what are the common operations perform on character string?

Answer:- A string is a sequence of characters that is treated as a single data item.

The common operations performed on character strings include the following:

(i) Reading and writting strings.

(ii) Combining strings together.

(iii) Copying one string to another.

(iv) Comparing strings for equality.

(v) Extracting a portion of a string.

Declaring and initializing string variables:-

In C, a string variable is any valid C variable name and is always declared as an array of characters. The general form of declaration of a string variable is:

```
char string_name[size];
```

The size determines the number characters in the string_name. Some Example are as follows:

```
char city[10];
char name[30];
```

When the compiler assigns a characters string to a character array, it automatically supplies a null character ('\0') at the end of the string. Therefore,

the size should be equal to the maximum number of characters in the string plus one.

Character arrays may be initialized when they are declared. C permits a character array to be initialized in either of the following forms:-

```
char city[9] = "New York";
char city[9] = {'N','E','W',' ','Y','O','R','K','\0'};
```

The reason that city had to be 9 elements long is that the string NEW YORK contains 8 characters and one element space is provided for the null terminator.

C also permits us to initialize a character array without specifying the number of elements. For example,

```
char string[ ] = {'G','O','O','D','\0'};
```

defines the array string as a five element array. We can also declare the size much larger than

the string size in the initializer. The statement

```
char str[10] = "GOOD";
```

is permitted. In this case, the computer creates a character array of size 10, places the value "GOOD" in it, terminates with the null character, and initializes all other elements to NULL.

Q Why do we need a terminating null character?

Answer:- The string is a variable-~~string~~ length structure and is stored in a fixed-length array. The array size is not always the size of the string and most often it is much larger than the string stored in it. Therefore, the last element of the array need not represent the end of the string. We need some way to determine the end of the string data and the null character serves as the "end of-string" marker.

Does C support string data type?

Answer:- A string is not a data type in C, but it is considered a data structure stored in an array.

Write a program to read a series of words from a terminal using scanf function.

Answer:-

```
#include <stdio.h>
int main()
{
  char word1[40], word2[40], word3[40], word4[40];
  printf("Enter text : \n");
  scanf("%s %s %s %s", word1, word2, word3, word4);
  printf("\n");
  printf("word1 = %s\n word2 = %s\n word3 = %s
        \n word4 = %s \n", word1, word2,
         word3, word4);
  return 0;
}
```

String-handling functions:- The C library supports a large number of string-handling functions that can be used to carry out many of the string manipulations discussed so far. Following are the most commonly used string-handling functions:

Function	Action
strcat()	concatenates two strings
strcmp()	compare two strings
strcpy()	copies one string over another.
strlen()	find the lenght of a string.

String-handling functions:- The C library supports a large number of string-handling functions that can be used to carry out many of the string manipulations discussed so far. Following are the most commonly used string-handling functions:

Function	Action
strcat()	concatenates two strings
strcmp()	compare two strings
strcpy()	copies one string over another.
strlen()	find the lenght of a string.

between modules that do not have calling-called relationship.

⑤ All modules are designed as single-entry, single-exit systems using control structures.

Q4 What is parameter and argument ?

Answer:-

Parameter:- The variables that are defined when the function is declared are known as a parameter.

Argument:- The values that are declared within a function when the function is called are known as an argument.

Points to note :-

① The parameter list must be separated by commas.

② The parameter names do not need to be the same in the prototype declaration and the function definition.

③ The types must match the types of parameters in the function definition, in number and order.

④ Use of parameter names in the declaration is optional.

⑤ If the function has no formal parameters, the list is written as (void).

⑥ The return type is optional, when the function returns int ~~data~~ type data.

⑦ The return type must be void if no value is returned.

⑧ When the declared types do not match with

the types in the function definition, compiler will produce an error.

Parameters Everywhere!

Parameters (also known as arguments) are used in following three places:

① in declaration (prototypes),
② in function call
③ in function definition.

Category of Functions.

A function, depending on whether arguments are present or note and whether a value is returned or not, may belong to one of the following categories:

Category-1: Functions with no arguments and

no return values.

Category 2: Functions with arguments and no return values.

Category 3: Functions with arguments and one return value.

Category 4: Functions with no arguments but return a value.

Category 5: Functions that return multiple values.

What is Recursion?

Answer:- Recursion is a special case, where a function calls itself.

Three rules to pass an Array to a Function.

① The function must be called by passing only the name of the array.

② In the function definition, the formal parameter must be an array type; the size of the array does not need to be specified.

③ The function prototype must show that the argument is an array.

Searching:

Searching is an operation that is used when a look-up needs to be conducted on a set of data to locate a particular element. This search operation may be conducted in a number of ways. Linear search and binary search are two of the most popular

searching ~~technig~~ techniques.

Sorting:

Sorting is an operation that is needed to arrange the data set in a specified order. Bubble sort, merge sort, heap sort, radix sort and quick sort are a few sorting algorithms.

What is the scope, visibility and lifetime of variables?

Answer:

Scope: Scope is defined as the area in which the declared variable is available. There are five scopes in C: program, file, function, block and prototype.

Visibility: Visibility is the "accessibility" of the variable declared.

Lifetime: The lifetime of a variable is the period of time in which the variable is allocated a space. There is three lifetimes in C: static, automatic and dynamic.

Describe the two ways of passing parameters to function. When do you prefer to use each of them.

Answer:- In C programming language, there are two ways of passing parameters from calling function to called function. They are:

(i) Pass by value.

(ii) Pass

Pass by value:- In call by value parameter passing method the copy of actual parameter values are copied to formal parameters and these formal parameters are used in called function. The changes mode on the formal parameter does not effect the values of actual parameters. That means, after the execution control comes back to the calling function. The actual parameters values remain same.

Pass by reference:- In call by reference parameter passing method, the memory location address of the actual parameters is copied to formal parameters. This adress is used to access the memory locations of the actual parameters in called function. In this method

of parameter passing, the formal parameter must be pointer variable.

When I prefer to use each of them:-

Passing by value copies the data, so passing large data structures by value can inhibit performance. Passing by reference passes only a address to the data. For large data structure, this can greatly improve performance. Passing by value copies the data so if the target code modifies that copy, it will not affect the original. Passing by reference passes only the address of the data, so modifications made against that reference will be visible to the calling code.

What is prototyping? Why is it necessary?

Answer:-

A function prototyping is a process that the declaration of a function that specifies function's name, parameters and return type.

Importance of function prototyping:- The function prototypes are used to tell the compiler about the number of arguments and about the required datatypes of a function parameter, it also tells about the return type of the function. By this information, the compiler cross-checks the function signatures before calling it.

Difference between actual and formal parameter.

Answer :-

Actual parameter	Formal parameter
The actual parameter are the variables that are transferred to the function when it is requested.	The formal parameters are the ~~variable~~ values determined by the function that accepts values when the function is declared.
In actual parameters, only the variable is ~~main~~ mentional, not the data types	In formal ~~data typ~~ parameters, the data type is required.
These can be variables, constant and expressions without data type	Formal parameters are variables with the data type
These parameters which are addressed in function call are called actual parameters.	Parameters addressed in the function description are called formal parameters.

Difference between & and * operator?

Answer:-

& operator	* operator
The & is a unary operator in C	The * operator is also unary operator.
It returns the memory address of the passed operand.	It returns the value of object pointed by a pointer variable.
This is also known as adress of operator.	It is known as value of address operator.

Difference between global and local variable?

Answer:-

Local variable	Global variable
Variables are declared inside a function.	Variable are declared outside any function.
Scope within a function, inside which they are declared.	Scope throughout the program.

Uninitialized local variable result in storage of the garbage value.	Uninitialized global variable stores zero by default.
Accessed only by the statements, inside a function in which they are declared.	Accessed by any statement in the entire program.

Distinguish between Automatic and static variables.

Answer:-

Automatic variables	static variables
All local variables are automatic variable. key-word auto can be used to declare an automatic variable.	Static variable declared by using keyword static.
Automatic variable life time is local (limited)	Static variable lifetime is not limited.
Automatic variables exit till the function execution time.	Since it's scope is local but the variable will be live till the program's execution.

Automatic variables a new each time	Static variable create once.

Distinguish between scope and visibility.

Answer:-

Scope is defined as the availability of a variable inside a program, scope is basically the region of code in which a variable is available to use.

Visibility of a variable is defined as if a variable is accessible or not inside a particular region of code or the whole program.