

→ ARRAYS :-

1. What is an Array?

Array is a datatype which is used to store similar data items in contiguous memory locations under a single variable name.

2. What is subscript or index?

The individual elements are addressed by specifying the index number called 'subscript'.

3. Explain types of Arrays. Write syntax.

1. 1-Dimensional Array

2. Multi-Dimensional array

 ↳ 2-Dimensional array

 ↳ 3-Dimensional array

1. 1-D Array → It is an array that has only one subscript.

datatype arrayname [size];

• Here datatype specifies the type of data say int, float, char, etc.

• Array name is the name of the array

• Size defines how many elements the array will hold.

Eg: int balance[10];

Here balance is a arrayname of type integer that holds 10 elements.

2. 2-D Array → Array which has 2 subscript is called a two dimensional array.

datatype arrayname [row][column]; (Explain the terms)

Eg: int a[2][2];

4. Write the structure of one-dimensional array

• Index / subscript for an array for n elements starts from zero and ends with n-1.

• Array[0] holds the first element, array[1] holds second

element and so on, array[n-1] holds the last elements.

5. Explain the Methods of initializing one dimensional array

1. Initialization in a single statement can be done as below

- int arr[4] = {1, 2, 3, 4} which means that an array of type of integer of size four having four elements in it.
- Number of elements in array should not be greater than the array size
- int arr[] = {1, 2, 3, 4} which is same as above type of initialization except that size of array is not specified which means array can hold any number of elements.

2. Initialization of array elements one by one :-

→ Consider int arr[4], ~~1 2 3 4~~

arr[0] = 1 indicates initialization of first element.

arr[1] = 2 indicates initialization of second element.

last element arr[n-1]

where n is the size of array → arr[3] = 4 - last element

3. Initializing of array (elements) using for or while loop.

let n be the number of elements

int arr[20], n, i; (declaration)

printf("Enter %d elements in", n);

[Initializing using for loop]

for (i=0; i < n; i++)

scanf("%d", &arr[i]);

[Initializing using while loop]

i = 0;

while (i < n)

{ scanf("%d", &arr[i]);

i++;

}

6. What are the applications of one dimensional array?

- Used for searching of an element
- Used for sorting of elements
- Used for storing elements of same type

7. What are the applications of two dimensional array?

- Reading and displaying elements of matrix
- Adding and subtracting the elements of 2 matrix
- Multiplying two matrices
- Searching and sorting elements of a matrix

8. What are the disadvantages of Arrays?

- We must know in advance that how many elements are to be stored in array.
- Array is a static structure. That means array is of fixed size. The memory allocated to array can not be increased or decreased.
- As array is of fixed size, if we allocate more memory than requirement then space will be wasted.
if we allocate less memory than requirement, then it will create problem.
- The elements of array are stored in consecutive memory locations, so insertion and deletion is very time consuming and difficult.

→ STRINGS

1. What is a string?

- A sequence of characters is called as a string
- There is no built-in data type for strings
- String always ends with null character, '\0'.
- String constants are always enclosed within double quotes.

char string-name [string-length];

where char is data type of strings

stringname is name for string variable

string-length : length of string which is to be stored

2. How to initialize a string?

char variable-name [size] = "Value";

char str[6] = { 'H', 'e', 'l', 'l', 'o', '\0' };

3. What are the methods of Reading and displaying strings?

1. using scanf and printf (formatted)
2. using gets and puts (unformatted)
3. using getchar and putchar

4. Explain different Manipulation function:

i) String Compare:

strcmp() - function is used to compare two string data, every character of one string is compared with corresponding position character of another [Compares the ascii value]

strcmp (string1, string2)

ii) String Concatenation:

strcat() - function is used to concatenate (attach) two strings. One string will be attached to the another string. (string2 will be attached to the end of string 1)

strcat (str1, str2)

iii) String Reverse:

strrev() - function is used to reverse a string. The characters from left to right in the original string are placed in reverse order.

strrev (string);

iv) String Copy:

strncpy() - function copy the n characters of one string to another

strncpy (str1, str2, n);

v) String length

strlen() - function is used to find out length of the string means on number of characters in it.

strlen (string_data);

5. What is a storage class?

- Storage classes can be used to define the scope (local variable or global) that is visibility and lifetime - that is duration and linkage of variables and functions.

storage class datatype variable name - syntax

6. How many Primary storage classes are there? Explain.

→ There are four primary storage classes

They are • Auto

- Register
- Static
- Extern

i) Auto:

Auto storage class is used for local variables within blocks or functions. It is a default storage class for local variable if none is specified.

The duration of the block or function in which it is defined. Once the block or function ends, the variable is destroyed.

Default - value will be garbage value

Location - Main Memory

Lifetime - within the block or function

Scope - Block / function

- Processing will be fast but time taken to run is more.

- So we use register

ii) Register:

The register storage class is used for local variable that are frequently accessed, suggesting to the compiler that the variable should be stored in CPU register for faster access.

Default - Garbage view

Location - CPU Register

Lifetime - Block / function

Scope - Block / function

To increase the time of execution we use register

iii) Static :-

The static storage class extends the lifetime of a variable to the entire program. For local variables this means they retain this value between function call. For Global variables and functions static limits their visibility to the function in which they are declared.

Default - Zero

Scope - Block / program

Location - RAM

Lifetime - Till the end of program.

iv) Extern :-

The extern storage class is used to declare a variable of function that is defined in another file.

It is included that the actual storage and definition are in a different file

• Global across all the function in program.

Default - zero

location - RAM

Scope - Global

Lifetime - Duration of entire program.

→ FUNCTIONS :-

1. What is a function?

Function is a group of statements that together perform a specific task.

- The main objective is to avoid the repetition of codes in a program
- It provides modularity and code reusability.
- Saves time and space

2. Explain the types of functions.

→ Functions can be broadly classified in two types

- Built in Functions

- User defined functions

i) Built in functions are the pre-defined functions

→ Also called as Library function

→ The functions available in C are defined under the Header files.

Some built in functions are

- String manipulation functions Eg: `strncpy`, `strlen`, `strcat` etc
- Character manipulation functions Eg:- `islower(ch)`, `isblank(ch)`
↳ `cctype.h` - `#include <cctype.h>` → Header file
- Memory management function
- Mathematical function → Eg: `exp(n)`, `clog(n)`, `sqrt()`
↳ `#include <math.h>` → Header file

ii) User defined Functions : As built in functions provided by C are not sufficient to perform customized functions → C allows the user to write their own functions to specific task.
perform

3. What is a header file?

• A header file is a file containing C declarations and macro definition to be shared between several source files.

• We use header file in program by including it, with the C preprocessing directive '`#include`'.

Eg: `stdio.h` - input / output functions.

`math.h` - Mathematical function.

• All built in functions are available in Header files

4. What is the location of the function in a program?

Location of function refers to the placement of the function definition in relation to main program.

Type 1 : Placing a function first, followed by the entire program

Type 2 : Placing function declaration before main program followed by main program and then followed by function body.

5. Explain the three elements of function

i) Function definition :-

Syntax : returntype function name (type1 parameter1, type2 parameter2, ...)

{

// function body goes here

}

- The first line is known as function header
- The statements within { } are function body
- returntype specifies the type of value returned by the function.
- function name is the name of function
- type 1 is the datatype of parameter 1
- type 2 is the datatype of Parameter 2
- The body consists of 3 parts.
 1. Local variable declaration
 2. Statements to perform the task inside the function
 3. return statement to return the result evaluated by the function.
- The parameters present in function definition are called formal parameters.

ii) Function declaration :-

Syntax : returntype function Name (type1 parameter1, type2 parameter2...);

Parts of function declaration :-

- | | |
|------------------|--------------------------|
| 1) Returntype | 3) Parameter list |
| 2) function name | 4) Terminating semicolon |

iii) Function call :-

Syntax: function Name(argument 1, argument 2...);

When a function is called, the control of the program gets transferred to the function definition.

→ (The arguments in function definition are known as formal parameters). The Arguments in a function call are called as 'actual parameters'] → What are formal and actual parameters?

- The name of the variables while declaring, calling and defining a function can be different.

6. Explain Variable declaration

There are two ways variable declaration

- 1) Local variable declaration
- 2) Global variable declaration

1) Local Variable declaration:

The variables are declared within the function and such variables can not be used in other function without declaring it again.

2) Global Variable declaration:

The variables that are declared before the main function and the scope of such variables is throughout the program.

7. What are the two ways of calling a function?

(For functions with arguments) (Parameter passing

1) Call by Value Mechanism)

When a variable or a value is passed to the function during its invocation, this function invocation is called function as call by value or parameter passing by value

2) When an address of the variable is called / passed to the function during its invocation then function invocation is called as call by reference or parameter passing by reference

8. what is calling and called function?

The function or a main() program which calls another function is called calling function.

Eg: main();

The function used or called by calling function is called as called function

Eg: display();

9. What are the types of user-defined functions based on parameters?

Type 1: With Parameter with returning value

Parameters are passed from calling function to the called function and based on the received parameter values called function performs required action and returns a value.

Type 2: With parameter without returning value

Parameters are passed from calling function to the called function and called function does not return a value. It just performs the specified action.

Type 3: Without Parameter with returning value

No parameter is passed from calling function to the called function, but function returns a value.

Type 4: Without Parameter without returning value

No parameter is passed from calling function to the called function. Called function does not return any value.

10. Parameter less functions — without parameter with returning value

• Without Parameter without returning value

→ (Type 3 and 4)

11. Void Functions • with Parameter with returning value

• With Parameter without (returning) returning value

→ (Type 1 and 2)

12. Note on Passing Arrays in Function

- Array elements or an entire array can be passed to a function.
- If we pass the values stored in an array, single element at a time, then it is called array passing by value or call by value
- If we pass the values stored in an array without any index, entire array at a time - then is called array passing by reference or call by reference.

→ RECURSIVE FUNCTION

What is a Recursive function? Give an example.

- Recursive functions are those functions that call themselves during their execution until certain condition are not satisfied.
- While using recursion, programmers need to be careful to define an exit condition from the function, otherwise it will go in infinite loop.
- Recursive functions are very useful to solve many mathematical problems.

Eg: To find Factorial of a number

Say we want to find factorial of 5, it can be done recursively.
Assume factorial(0) is 1 which acts as a stopping condition.

$$\begin{aligned} \text{factorial}(5) &= 5 * \underline{\text{factorial}(4)} \\ &= 5 * 4 * \underline{\text{factorial}(3)} \\ &= 5 * 4 * 3 * \underline{\text{factorial}(2)} \\ &= 5 * 4 * 3 * 2 * \underline{\text{factorial}(1)} \\ &= 5 * 4 * 3 * 2 * 1 * \underline{\text{factorial}(0)} \\ &= 5 * 4 * 3 * 2 * 1 * 1 \\ &= 120 \end{aligned}$$

At each step factorial function calls itself and the procedure is repeated until it reaches stopping condition.