

Q programming with c.

Q 1st CT: chapter-10, chapter-11

Question:

1. Rules of pointer operations.
2. Rules for pass by pointer.
3. Differentiate pass by value vs pass by pointer.
4. What is pointer.
5. How to declare the pointer variable.
6. Differentiate between structure and union.
7. What is structure.
8. Differentiate between array vs structure.
9. How to declare a structure variable. or, what do mean by structure variable
10. Rules for initializing Structure variable.
11. What is union.

Chapter - 10

5.5 Structures and unions

1. What is structure:

→ Structure is a user defined datatype in C language which allows us to combine data of different types together. Structured helps to construct a complex data type which is more meaningful.

2. How to declare a structure variable or, what do mean by structure variable:

→ A structure variable declaration is similar to the declaration of variables of any other data types. It includes the following elements:

1. The Keyword "struct"
2. The Structure tag name.
3. List of variables names separated by commas.
4. A Terminating semicolon.

8.

3. Rules for initializing structure variable:

→ There are a few rules to keep in mind which are as follows:-

1. We cannot initialize individual members inside the structure template.
2. The order of values enclosed in brackets must match the order of members in the structure definition.
3. We can initialize only the first few members and leave the remaining blank.

7. The uninitialized members will be assigned default values as follows:-
- Zero for integer and floating point numbers.
 - '0' for characters and strings.

Q4. Difference between array and structure:

⇒ Array:

1. An array is collection of items stored at contiguous memory location.
2. Array uses square bracket "[]" for element access.
3. Array is pointer.
4. ~~initial~~ Instantiation of array objects is not possible.
5. Array size is fixed.
6. Bit filed is not possible in an array.
7. Array is a non-primitive datatype.
8. Array traversal and Searching is easy and fast.
9. Array elements are accessed by their index number.

⇒ Structure:

1. A structure is a user defined datatype in c language.
2. Structure uses ". " (dot operator) for element access.
3. Structure is not a pointer.
4. Instantiation of structure objects is possible.
5. Structure size is not fixed.
6. Bit filed is possible in a structure.
7. Structure is a structure defined datatype.
8. Structure traversal and Searching is complex and slow.
9. Structure elements are accessed by their names.

Q6. Difference between structure and union:

⇒ Structure:

1. We use the `struct` statement to define a structure.
2. Every member is assigned a unique memory location.
3. In the structure, change in the value of one data member does not affect other data members.
4. We can initialize multiple members at a time.
5. A structure can store multiple values of the different members.
6. Users can access any member at a time.

⇒ Union:

1. We use the "union" keyword to define a union.
2. All the data members share a memory location.
3. In the union, change in the value of one data member affects the value of other data members.
4. We can initialize only the first member at once.
5. A union can store one value at a time.
6. Users can access only one member at a time.

what is union?

⇒ A union is a user-defined data type. It is just like the structure except that all of its members start at the same location in memory.

= Rules of pointer operations:—

1. A pointer variable can be assigned the address of another variable.
2. Can be assigned the values of another pointer variables.
3. Can be initialized with NULL or zero value.
4. Can be pre-fixed or post-fixed with increment or decrement operations
5. An integer value may be added or subtracted from a pointer variable.
6. A pointer variable can not be multiplied by a constant.
7. Two pointer variable cannot be added.

4 Rules for pass by pointer:

1. The types of the actual and formal arguments must be same.
2. Actual arguments must be the address of variables that are local to the calling function.
3. Formal arguments in the function header must be prefixed by the indirection operator.
4. In the prototype, the arguments must be prefixed by the symbol.
5. To access the value of an actual arguments in the called function, we must use the corresponding formal argument prefixed with the indirection operator.

5. pass by value versus pass by pointer:-

pass data one function to another is known as parameter passing. parameter passing can be done in following two ways:-

1. pass by value
2. pass by pointer.

In pass by value, values of actual parameters are copied to the variables in the parameter list of the called function. The called function works on the copy and not on the actual values of the actual parameter.

In pass by pointers, the memory addresses of the variables rather than the copies of values are sent to the called function. In this case, the called function directly works on the data in the calling function.

chapter-12

Q The important file handling functions that are "already" available in the c library:-

- fopen() → Creates a new file for use.
- fclose() → Closes a file which has been opened for use.
- getc() → Reads a character from a file.
- putc() → Writes a character to a file.
- fprintf() → Writes a set of data values to a file.
- fscanf() → Reads a set of data values from a file.
- gets() → Reads an integer from a file.
- puts() → Writes an integer to a file.
- fseek() → Sets the position to a desired point in the file.
- ftell() → Gives the current position in the file.
- rewind() → Sets the position to the beginning of the file.

Q what are the fprintf and fscanf function. write down the general form of these function.

→ The fprintf() function is used to writes a set of data values to a file. The general form of fprintf() is:-

fprintf(fp, "control string", list);

Where, fp is the file pointer associated with a file. The "control string" contains output specifications for the item in the list.

→ The fscanf() function is used to reads a set of data values from a file.

The general form of fscanf() function

is :—

fscanf(fp, "control string", List);

where, fp is the file pointer. The "control string" contains input specifications for the item in the list.

For example: fscanf(f2,"%s %d",item,s quantity);

Q How to store data in a file in the Secondary memory?

→ If we want to store data in a file in the secondary memory, we must specify certain things about the file to the operating system. They include the following:—

1. filename
2. data structure
3. purpose.

→ filename: filename is a string of characters that make up a valid filename for the operating system.

→ data structure: data structure of a file is defined as FILE in the library of standard function definitions. Therefore, all file should be declared as type FILE before they are used.

For example: FILE *fp;
fp=fopen("filename", "mode");

→ purpose: the second statement also specifies the purpose of opening this file. The mode does this job. Mode com-

one of the following:-

r → open the file for reading only.

w → open the file for writing only.

a → open the file for appending.

Q Explain handling errors in input output operations.

→ For input output operations, there are several important error-handling phrases and clauses. These are as follows:-

1. AT END phrase

2. INVALID KEY phrase

3. NO DATA phrase

4. File status clause.

An important characteristics of error handling is the issuing of a runtime message when an error occurs during the processing of an input output statement if there is no AT END INVALID KEY phrase in the input output statement, FILE STATUS clause in the select statement for the file.

Q Explain mode:-

There are some mode including e-programming. They are as follows:-

r → Open the file for reading only

w → Open the file for writing only

a → Open the file for appending.

Chapter-1

2.1 Overview of c

Ques Important of c language: — /advantage .

1. c programming can directly interact with hardware .
2. c programming language are highly portable .
3. c is faster than dynamically typed languages .
4. c is an easy and simple language .
5. c is structured programming language .
6. There are 32 never reserved keyword in c , which mean they cannot be used for other purposes .
7. The c language is considered
8. The execution and compilation of codes is faster compare to any other programming language .
9. c is used in system programming mostly .

Ques #include directive: #include preprocessor directive is used to paste code of given file into current file .
By the use of #include directive , we provide information to the #include preprocessor where to look for the header files .

for example: #include <filename> .

The main function:-

There are Six main function have in ~~every~~ C language. They are -

• main(), • void main(),

• main(), • int main(), • void main (void),

• main(void), • int main(void), • void main (void),

The empty pair of parentheses indicates that the function has no arguments. The Keyword "void" means that the function does not return any information. and int means that the function returns an integer value + the operating system. When it is specified the last statement in the program must be return zero.

#define directive: A #define is a preprocessor command

#define directive and not a statement. Therefore, #define lines should not end with a semi colon;

<math.h> → math.h is the file name containing the required function.

<stdio.h> → stdio.h refers to the standard header file containing standard input and output functions.

#include <filename>

55 Basic structure of c. programs:—

A c program may contain one or more Sections.
They are:—

Documentation Section

Link section

Definition Section

Global declaration Section.

main() function Section

{ declaration part

~~execution~~
executable part

Sub program Section

function-1

function-2

—

function-n

} (user-defined function)

- Documentation Section : This section contains consists of a set of comment lines giving the name of the program.
- Link section provides instructions to the compiler to link function from the system library.
- Definition section defines all symbolic constants.
- Global declaration section declares all global variables.
- The main() section contains two parts, declaration part and executable part. Declaration part declares of all variables to use executable part.
- Subprogram section contains all the user-defined functions that are called in the main function.

Q Describe the process of executing a program.

During execution, the program may request for some data to be entered through the keyboard. Sometimes the program does not produce the desired results. Perhaps, something is wrong with the program logic or data. Then it would be necessary to comment the source program or data. The entire process of compiling, linking and executing the program should be repeated.

Q Why do we need to use comments in program?

Comments is used in every program to easily describle the purpose of the code. A user should be able to utilize a previously written program.

Chapter-2

Constants, Variables and Datatypes

C ANSI-C Keyword:

Auto

break

case

char

const

continue

default.

do

double

else

enum

extern

float

for

goto

if

int

long

register

return

short

signed

sizeof

static

Struct

switch

typedef

union

unsigned

void

volatile

while

Rules for Identifiers: —

1. First character must be an alphabet (or underscore)

2. must consist of only letters, digits or underscore.

3. Only first 31 characters are significant.

4. Cannot use a Keyword.

5. must not contain white space.

Chapter 1

~~Ques.~~ Describe the four basic data types. How could we extend the range of values they represent?

1. Integer type: Integer are whole numbers with a range

of values supported by a particular machine. An integer occupies 2 bytes memory space. Range (-32768 to 3276)

2. Character type: A single character can be defined

as a character type data characters are usually stored in 8 bits of internal storage. Range (-128 to +127)

3. Floating point type: Floating point number are stored in 32 bits, with 6 digits of precision. Floating point numbers are denoted by the keyword float. Range (~~-3.4E-38 to 3.4E+38~~)
Range (3.4E-38 to 3.4E+38)

4. Void Type: The void type has no values. This is usually used to specify the type of function. The type of a function said to be void when it does not return any value.

what is
value?

what is variable and what is meant by the "value" of a variable?

A variable is a name of the memory location. It is used to store data.

the value of the variable, which is the value the variable name represents.

Q How do variables and symbolic names differ?

⇒ An instance of an object is created when a variable is declared. Declaration of symbolic name just defines a name that can be used in a program.

⇒ Symbolic names are called constant identifier. On the other hand variable value can be changed within the program.

Q Overflow and Underflow of data:-

when the value of a variables becomes too large or too small for its type for the computer running the application. When this occurs, it's called overflow and underflow.

Bitwise operator: Bitwise operators are used to perform bit level operation on the operands. Lists the bitwise operators:—

1. & → Bitwise AND.
2. | → Bitwise OR.
3. ^ → Bitwise exclusive OR.
4. << → shift left.
5. >> → shift right.

Assignment operator: Assignment operators are used to assigning value to a variable. The left side operand of the assignment operator is a variable and right side operand of the assignment operator is a value.

Others/ Special operators: —

1. Sizeof operator: Sizeof operator can be used to compute the size of its operand.
2. comma operator: The comma operator is a binary operator that evaluates its first operand and discards the result.

chapter-8 (Operations and expressions)

(6)

④ **Operator:** An operator is a symbol that tells the computer to perform certain mathematical or logical manipulations.

e has many built-in operators and can be classified into 6 types. They are:—

1. Arithmetic operators
2. Relational operators
3. Logical operators
4. Bitwise operators
5. Assignment operators
6. Other operators

④ **Arithmetic operators:** These operators are used to perform mathematical operations. Examples: (+, -, *, /, %, ++, --).

④ **Relational operators:** These operators are used to compare of the values of two operands. Examples: (==, >, <=)

④ **Logical operators:** Logical operators are used to combine two or more conditions to complement the evaluation of the original condition in consideration. e has following three logical operators

1. && → meaning Logical AND
2. || → meaning logical OR
3. ! → meaning Logical NOT.

chapter-11 (pointers)

Rules

1. what is pointer?

A pointer is a variable whose value is the address of another value variable.

2. How to declare the pointer variable?

The declaration of a pointer variable takes the following form:-

data-type *pt-name;

1. The asterisk (*) tells that the variable pt-name is a pointer variable.

2. pt-name needs a memory location.

3. pt-name points to a variable of type dataType.

For example:-

`int *p; /*integer pointer*/`

Declares the variable p as a pointer variable that points to an integer data type.

`float *x; /* float pointer */`

Declares x as a pointer to a floating point variable.