All Deaf Educational Technolog

Power of Knowledge

Computer

PROGRAMMING

ABU SALEH MUSA MIAH(ABID)

A Guidebook of Computer Programming with C

Engr. Abu Saleh Musa Miah (Abid)
Lecturer, Rajshahi Engineering Science and Technology College(RESTC)
B.Sc.Engg. in Computer Science and Engineering(**First Class First**)
M.Sc.Engg.(CSE-Pursuing),University of Rajshahi.
Email:abusalehcse.ru@gmail.com
Cell:+88-01734264899

Computer Programming with C

Writer	:	**Engr. Abu Saleh Musa Miah (Abid).** B.Sc.Engg. in Computer Science and Engineering(**First Class First**) **M.Sc.Engg.**(CSE-Pursuing),University of Rajshahi. **Cell:+88-01734264899**
Email	:	**abusalehcse.ru@gmail.com**
Publisher	:	**Abu Syed Md Mominul Karim Masum.** **Chemical Engr.** Chief Executive Officer (CEO) Swarm Fashion, Bangladesh. Mohakhali,Dhaka.
Email	:	**swarm.fashion@gmail.com**
Cover page Design	:	**Md. Kaiyum Nuri** Chief Executive Officer (CEO) Jia Shah Rich Group(Textile) **MBA,Uttara University**
First Publication	:	**Octobor-2016**
Copyright	:	**Writer**
Computer Compose	:	**Writer**
Print	:	**Royal Engineering Press & Publications.** Meherchandi, Padma Residential Area, Boalia, Rajshahi.
Reviewer Team	:	**Engr. Syed Mir Talha Zobaed.** **B.Sc. Engg. (First Class First)** **M.Sc. Engineering (CSE)** **University of Rajshahi** **Omar Faruk Khan (Sabbir)** **M. Engineering (CSE) University of Rajshahi**
PRICE	:	**400.00 TAKA (Fixed Price).** US 05 Dollars (**Fixed Price**).

OOP CPP	:	**Engr. Abu Saleh Musa Miah (Abid).**
Published by	:	**Rajshahi Engineering Science and Technology College.** **Mirjapur,Binodpur,Rajshahi.**

Dedicated

To

My Family and

My honorable Teahers of CSE RU.

.

[This Page is Left Blank Intentionally]

লেখকের কথা

Computer Programming with C কোর্সটি বাংলাদেশের প্রায় সকল বিশ্ববিদ্যালয়ের সিলেবাসে স্থান লাভ করেছে,বাজারে টেক্সটবই থাকলেও গভীর ভাবে অনুধাবন ও পরীক্ষায় ভাল মার্কস উঠানোর মত সাজানো টেক্সটবই সহজলভ্য নয় । পরীক্ষার প্রস্তুতির স্বার্থে তাই, এই গাইডবইটি আপনাকে সহযোগিতা করবে বলে আশা করা যায় ।

এই বইয়ের পাঠক হিসেবে, আপনিই হচ্ছেন সবচেয়ে গুরুত্বপূর্ণ সমালোচক বা মন্তব্যকারী । আর আপনাদের মন্তব্য আমার কাছে মূল্যবান, কারন আপনিই বলতে পারবেন আপনার উপযোগী করে বইটি লেখা হলো কিনা অর্থাৎ বইটি কিভাবে প্রকাশিত হলে আরও ভাল হতো । সামগ্রিক ব্যাপারে আপনাদের যে কোন পরামর্শ আমাকে উৎসাহিত করবে ।

Engr. Abu Saleh Musa Miah (Abid)
Writer

ACKNOWLEDGEMENTS

I wish to express my profound gratitude to all those who helped in making this book a reality; especially to my families, the classmates, and authority of Royal Engineering Publications for their constant motivation and selfless support. Much needed moral support and encouragement was provided on numerous occasions by my families and relatives. I will always be grateful, to the numerous web resources, anonymous tutorials hand-outs and books I used for the resources and concepts, which helped me build the foundation.

I would also like to express my profound gratitude to Engr. Syed Mir Talha Zobaed for his outstanding contribution in inspiring, editing and proofreading of this guidebook. I am also grateful to Omar Faruque Khan (Sabbir) for his relentless support and guideline in making this book a reality.

I am thankful to the following readers those have invested their valuable times to read this book carefully and have given suggestion to improve this book:
Abu Hurayra (Principle,RESTC)
Engr. Mainuddun Maruf (B.Sc. Engg in Electrical and Electronics Engineering. IUT) .
Md.Moyeed Hossain (B.Sc.Engr. Computer Science and Engineering ,RUET, Lecturere, RESTC).
Md.Shamim Akhtar (B.A Honours,English,M.A. Lecturere, RESTC).
MD.Sharafat Hossain(B.Sc. Engg in Electrical and Electronics Engineering. RUET).
……………………………... And numerous anonymous readers.
I am also thankful to the different hand notes from where I have used lots of solutions, such as Dynamic Memory Allocation,Operator overloading etc

I wish to express my profound gratitude to the following writers whose books I have used in my Guidebooks:

1.	**Kernighan and Ritchie**	:	The C Programming Language, Prentice Hall
2.	**Gotfreid**	:	Programming with C, Schaum's Outline Series, Tata McGraw Hill
3.	**D.E. Knuth**	:	**The Art of Computer Programming**, *Addison-Wesley Professional*
4.	**E. Balagurusamy**	:	**Programming with ANSI C**, *Tata McGraw Hill*
5.	**H. Schildt**	:	**Teach yourself C**, *McGraw-Hill Publishers*

…………………….. and Numerous anonymous Power Point Slides and PDF chapters from different North American Universities.

Computer Programming with C Syllabus

CSE1121: Computer Programming with C
75 Marks [70% Exam, 20% Quizzes/Class Tests, 10% Attendance]
3 Credits, 33 Contact hours, Exam. Time: 4 hours

Introduction: Programming languages, basic concepts of compiler, interpreter, algorithm and flowchart.
Simple C: Program structure in C, Program creating, compiling, debugging and running, Basic I/O functions, Identifiers and keywords, Simple data types, variables, constants, operators, Bitwise operators, comments, Decision making statements with if and switch, Looping structures with for, while, do-while.
More Data Types: Array, Structures, Union, Pointes, Strings, Dynamic allocation, Static, global, external and registrar, User defined data types
Functions: C Functions and user defined function, Function types, parameters, prototypes, Recursive function.
File Handling: Concepts, Character and File I/O, Basics of simple File I/O, ANSI Standard Libraries.
Others: Pre-processor with define, include, macro, ifdef, Uses of graphics functions.
Books Recommended:

1.	**Kernighan and Ritchie**	:	The C Programming Language, Prentice Hall
2.	**Gotfreid**	:	Programming with C, Schaum's Outline Series, Tata McGraw Hill
3.	**D.E. Knuth**	:	**The Art of Computer Programming**, *Addison-Wesley Professional*
4.	**E. Balagurusamy**	:	**Programming with ANSI C**, *Tata McGraw Hill*
5.	**H. Schildt**	:	**Teach yourself C**, *McGraw-Hill Publishers*

Index

Total=75 Marks
Class Test/MCQ=15
Attendence=7.50
Exam=26.25+26.25=52.50

Part-A Marks-26.25

4 set*8.75Marks=26.25

Part-B Marks-26.25

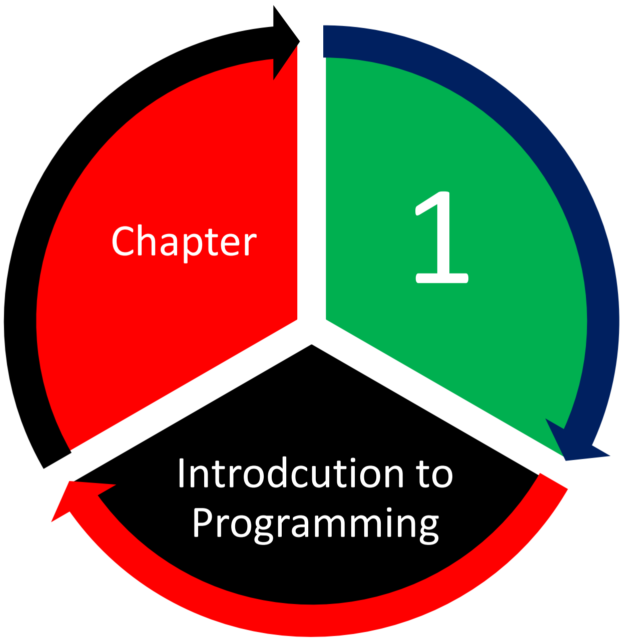
4 set*8.75Marks=26.25

Total=15+7.50+52.50

1.

Part-A

Marks-26.5



Importent Question

1. What is computer programming? **Exam-2012**
2. What is meant by a computer program? What , in general, happens when a computer program is executed? **Exam-2013**
3. Name some commonly used high-level language. What are the advantages of using high-level languages? **Exam-2013**
4. Mention the advantages of high level programming language. Differentiate between compiler and interpreter.**Exam-2014**
5. What is meant by compilation? What is meant by interpretation? What difference between tow processes differ?**Exam-2013**
6. Differentiate between pseudo-code and algorithm with example. **Exam-2014**
7. What is an escape sequence? What is its purpose? **Exam-2013**
8. Define escape sequence. List any 5 escape sequence used in c programming. **Exam-2014**
9. Explain the purpose of keyword " void " and "return" statement. **Exam-2014**
10. What is run time error? Give an example. **Exam-2015**

Q.What does computer-programming mean? Exam:Acce-2014
Q.What is mean by computer program and programming language? Exam:Acce-2013
Q.What is Programming Language?
Ans:
Computer program:
To process of particular set of data the computer must be given an appropriate set of instructions called a program . these instruction are entered into the computer and then stored in a portion of computers memory.
Computer programming is a process that leads from an original formulation of a computing problem to executable **computer** programs.
Programming language:A **programming language** is a special **language programmers** use to develop software programs, scripts, or other sets of instructions for computers to execute. The following is an index of the different **programming** and scripting **languages** currently listed at mine.

Q.Discuss different kind of Programming language? Exam:Acce-2013

Language types
1.From

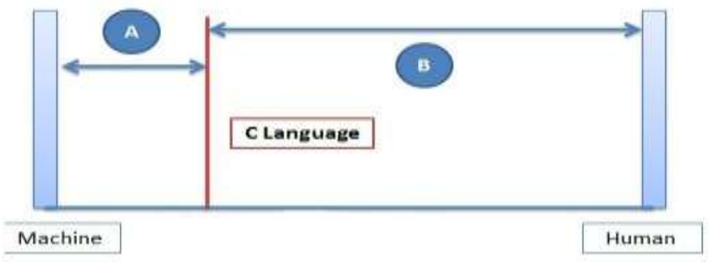
- ❖ Machine and assembly languages
- ❖ Algorithmic languages
  - FORTRAN
  - ALGOL
  - LISP
  - C
- ❖ Business-oriented languages
  - COBOL

2. Above the level

- **High level** - Ada , Modula-2 , Pascal, COBOL, FORTRAN, BASIC, Java, C++.
- **Middle level** - C, FORTH, Macro-assemble.
- **Low level** - Assembler.

Q.why c is called middle level language? Exam:Acce-2013
Middle Level Language:
C Programming bridges gap between Machine Understandable Machine Level language and more conventional High level languages.
Why C is Middle Level Language:
1. C programming language support the low level language i.e. Assembly Language..
2. C language also gives the facility to access memory through pointer.
3. Its combines the elements of high-level languages with the functionalism of assembly language.
4. Using inline assembly language feature in C we can directly access system registers.
5. C Programming also Supports high Level Language Features.

So, C language neither a High Level nor a Low level language but a **Middle Level Language**.



Q.Why compiler Need?

Ans:
The term compiler refers to the way in which a program is executed. in theory any programming language can be compiled or interpreted.For example BASIC is usually interpreted and c is usually compiled. The way a program is executed is not defined by the language in which it is written compilers are simply sophisticated programs that operate on our program source code.

Q.Why to use C?
Ans:
C was initially used for system development work, in particular the programs that make up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as code written in assembly language. Some examples of the use of C might be:

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities

Q.What is Compiler? Exam:ACCE-2012,ICE-CSE-
Ans:
A **compiler** is a computer program that transforms source code written in a programming language (the source language) into another computer language (the target language)
or
with the latter often having a binary form known as object code.

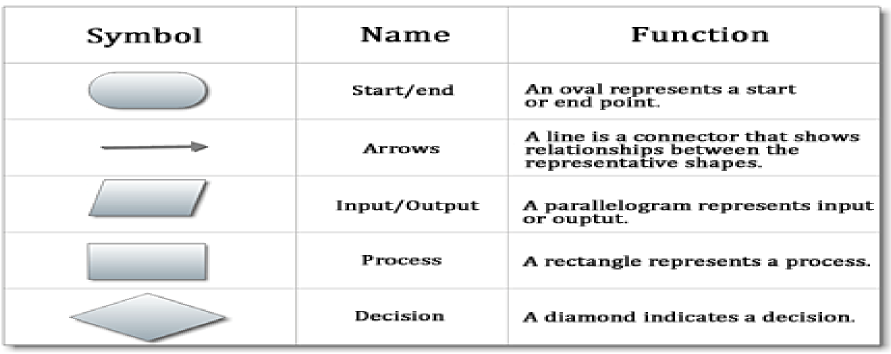
Q.What is Algorithm?Exam:ACCE-2012,ICE-CSE- MSE,APPE-,CSE-2011
Ans:
Algorithm is a finite set of steps to solve problem by computer.
Or An algorithm (pronounced AL-go-rith-um) is a procedure or formula for solving a problem.
OrAn algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time
Example: If we want largest number from 2 number .that time our algorithm is

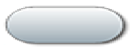
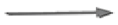
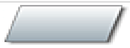
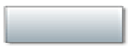
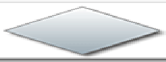
- Input 2 number
- Choose largest number
- Then print largest number

Question: Differentiate between pseudo-code and algorithm with example Exam-2016
Answer:

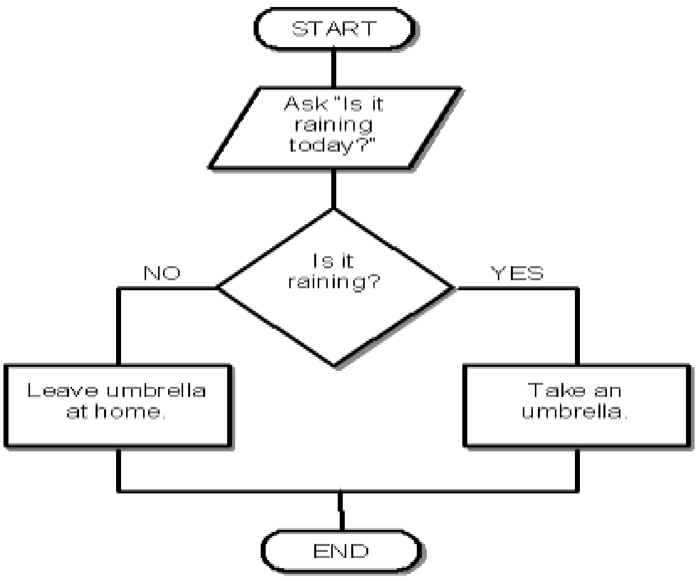
Algorithm	**Pseudocode**
Algorithms can be described in various ways, from pure mathematical formulas to complex graphs, more times than not, without pseudo code.	Pseudo code describes how you would implement an algorithm without getting into syntactical details

Q.What is flowchart?
Ans:
A **flowchart** is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows.
or
Flowchart is a graphical representation use to solve a particular problem.
Symbole of flowchart:



Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

Example:



Q. Discuss the similarities and dissimilarities of algorithm and flowchart. Exam-2011
Ans:
Similarities:

- ➢ Flow chart is very important tool for developing algorithm and program . It is pictorial representation of step by step solution of a problem.
- ➢ Programmer often uses it as a program planning tool for visually organising step necessary to solve a problem. It uses boxes of different shapes that denotes different type of instruction.
- ➢ While making a flow chart a programmer need not to pay attention on the elements of the programming language ,he has to pay attention to the logic of solution to the problem

- ❖ The term algorithm refers to the logic.it is step by step description how to arrive at the solution to the problem.algorithm is define as sequence of instruction that when executed in the specified sequence the desired results are obtained .
- ❖ The set of rules that define how a particular problem can be solved in finite number of steps is known as algorithm.

- ❖ A good algorithm help us to create a good program.

Dissimilarity:

- An algorithm is a description of how to carry out a process. An algorithm lists the steps that must be followed to complete the process. Algorithms can be described in English but such descriptions are often ambiguous and open to misunderstanding. Therefore various formal methods of describing algorithms have been developed. The simplest of these is the flowchart.
- A flowchart consists of a sequence of instructions linked together by arrows to show the order in which the instructions must be carried out. Each instruction is put into a box. The boxes are different shapes depending upon what the instruction is.

Q.What is a static variable? APPE-CSE-
Ans:
A static local variables retains its value between the function call and the default value is 0. The following function will print 1 2 3 if called thrice.

```
void f() {
  static int i;
  ++i;
  printf("%d ",i);
}
```
If a global variable is static then its visibility is limited to the same source code.

Q.Define static variable. CSE-2011
Ans:
Akindoflocal variableincertainprogramming languagesthatretainsitsvalueevenwhenprogramexecutionmovesoutsideitsscope. Staticvariableshaveafixedlocationinthedatasectionoftheprogram'saddress spacewhereasautomaticvariablesaretypicallyallocatedonthestack.

Q. What are the differences between compiler and interpreter?
OR compare between compiler and interpreter? Exam: CSE-Acce-2014,13,12
Ans:
Difference between Compiler and Interpreter
Comparison of compiler and interpreter are as follows:

Compiler	**Interpreter**
Scans the entire program and translates it as a whole into machine code	Translates program one statement at a time.
Intermediate Object Code is **Generated**	**No** Intermediate Object Code is **Generated**
Conditional Control Statements are Executes **faster**	Conditional Control Statements are Executes **slower**
Memory Requirement : **More** (Since Object Code is Generated)	**Memory Requirement** is **Less**
Program need not be **compiled** every time	Every time higher level program is converted into lower level program
6**Errors** are displayed after **entire program** is checked	Every **Errors** are displayed for **every instruction** interpreted (if any)
7**Example** : C ,C++Compiler	**Example** : BASIC, Python, Ruby

Q. Describe Different type of Header, Keyword and its tasks?Exam:ICE-2013,CSE,APPE-
Ans:

There are some header file in C language.Example:

- ❖ **#include <stdio.h>**
  - Stdio abbreviation of Standard input output.
  - Preprocessor directive
  - Tells computer to load contents of a certain file

  #include<stdio.h>
  #include<conio.h>
- ❖ **"printf( )" is used -**
  - (a) 1.To show the text display
  - (b) 2.To show the display the content of the variable
- ❖ **"clrscr( )"** is used to clear the screen.
- ❖ ***Getch(**) is used to get the output & to comeback to the program
- ❖ **"scanf( )"** is used to get the input from the user.
  - ❖ it is stored in header file like dictionary i.e. <stdio.h>
- ❖ **int main()**
  - C++ programs contain one or more functions, exactly one of which must be **main**
  - Parenthesis used to indicate a function
  - **int** means that **main** "returns" an integer value
  - Braces (**{** and **}**) indicate a block
  - The bodies of all functions must be contained in braces
- ❖ **printf("Welcome to C!\n");**
  - Instructs computer to perform an action
  - Specifically, prints the string of characters within quotes (" ")
  - Entire line called a statement
  - All statements must end with a semicolon (;)
  - **\n** is the newline character
- ❖ **scanf("%d", &integer1);**
  - Obtains a value from the user
  - **scanf** uses standard input (usually keyboard)
  - This **scanf** statement has two arguments
  - **%d** - indicates data should be a decimal integer
  - **&integer1** - location in memory to store variable
- ❖ **return 0;**
  - A way to exit a function
  - **return 0**, in this case, means that the program terminated normally

Question:Explain the purpose of keyword " void " and "return" statement. Exam-2014
Answer:

<u>Purpose "void" and "return" keyword:</u>

<u>Return keyword:</u>

A return statement causes execution to leave the current subroutine.
and resume at the point in the code immediately after where the subroutine was called, known as its return address. The return address is saved, usually on the process's call stack, as part of the operation of making the subroutine call. Return statements in many languages allow a function to specify a return value to be passed back to the code that called the function.

If no return statement appears in a function definition, control automatically returns to the calling **function** after the last **statement** of the called **function** is executed. In this case, the **return** value of the called **function** is undefined.

<u>Void keyword:</u>

We use void to indicate that a function doesn't return a value or that it has no parameters or both

Q.How a negative integer is stored.?
Ans:
Get the two's compliment of the same positive integer. Eg: 1101 (-5)
Step-1 – One's compliment of 5 : 1010
Step-2 – Add 1 to above, giving 1011, which is -5

Q.What is the purpose of extern storage specifier? CSE-2011
Ans:
Used to resolve the scope of global symbol.

```
Ex:
main(){
externint i;
Printf("%d",i);
}
int i =20;
```

Q.Explain the purpose of the function sprintf().
Ans:
Prints the formatted output onto the character array.

Q.What is the purpose of the keyword typedef? Exam:ICE-2013
Ans:
It is used to alias the existing type. Also used to simplify the complex declaration of the type.

Q.Can a program be compiled without main() function?
Ans:
Yes, it can be but cannot be executed, as the execution requires main() function definition.

Q.How can you print a \ (backslash) using any of the printf() family of functions.
Ans:
Escape it using \ (backslash).

Q.What is typecasting?
Ans:
Typecasting is a way to convert a variable/constant from one type to another type.

Q.What is the default value of local and global variables?
Ans:
Local variables get garbage value and global variables get a value 0 by default.

Q.Explain the use of comma operator (,).
Ans:
Comma operator can be used to separate two or more expressions.
Eg: printf("hi") , printf("Hello");
Q.How many operators are there under the category of ternary operators?
Ans:
There is only one operator and is conditional operator (? :)
What is reminder for 5.0 % 2?
Ans:
Error, It is invalid that either of the operands for the modulus operator (%) is a real number.

Q.Where the address of operator (&) cannot be used?
Ans:
It cannot be used on constants.

It cannot be used on variable which are declared using register storage class.

Question:Define escape sequence. List any 5 escape sequence used in c programming. Exam-2014
Question: What is an escape sequence? What is its purpose? Exam-2013
Answer:
Escape Sequence:
An escape sequence refers to a combination of characters beginning with a back slash (\) followed by letters or digits.
Escape sequences represent non-printable and special characters in character and literal strings. As such, they allow users to communicate with a display device or printer by sending non-graphical control characters to specify actions like question marks and carriage returns.

Purpose of Escape Sequence:
An escape sequence is used when writing sections of code, like preprocessor definitions, to specify continuation characters, so that multiple lines of code are considered as a single line by the compiler. Regular expressions that help perform sophisticated string search operations use escape sequences to locate substrings within a large string. By enabling quoted strings, escape sequences may be used to create output files containing text template tags and files

Escape sequences always begin with a backslash

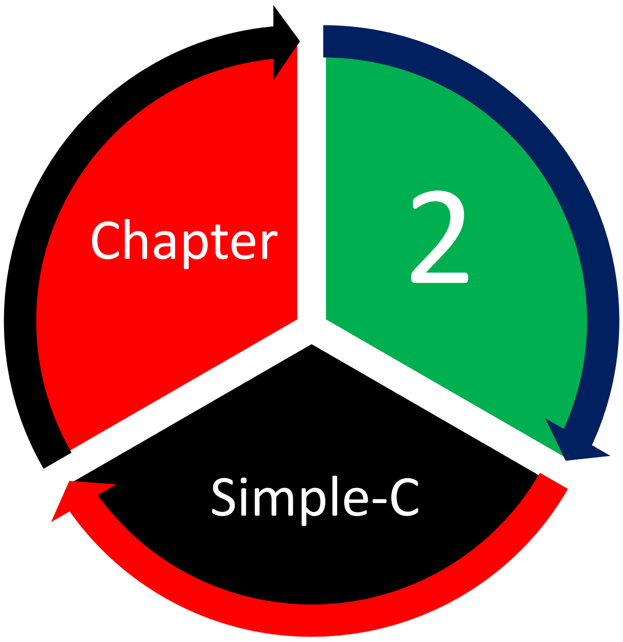
Example:
C interprets **\n inside** a literal as a newline character, whatever that may be on the target system:

```
#include<stdio.h>
intmain(){
printf("Hello,\nworld!");
}
```

In this code, the escape sequence \n does not stand for a backslash followed by the letter n, because the backslash causes an "escape" from the normal way characters are interpreted by the compiler. After seeing the backslash, the compiler expects another character to complete the escape sequence, and then translates the escape sequence into the bytes it is intended to represent. Thus, "Hello,\nworld!" represents a string with an embedded newline, regardless of whether it is used inside printf or anywhere els
List of escap sequence:

Escape sequence	Hex value in ASCII	Character represented
\a	07	Alert (Beep, Bell) (added in C89)[1]
\b	08	Backspace
\f	0C	Formfeed
\n	0A	Newline (Line Feed); see notes below
\r	0D	Carriage Return
\t	09	Horizontal Tab
\v	0B	Vertical Tab

Question:What is run time error? Give an example. Exam-2015
Answer:
Run Time Error:
An error that occurs during the execution of a program. In contrast, compile-time errors occur while a program is being compiled. Runtime errors indicate bugs in the program or problems that the designers had anticipated but could do nothing about.
For example: running out of memory will often cause a runtime error.



2.1 Introduction to C Important Question

1. What are the general characteristics in C ? **Exam-2012**
2. Correct the following C programs so that you can compile them successfully.
   a.c

```
/* Date. 5.3.2015
Main(){
    int i,
    for(i=0,i<5,i++){
            scanf("%d", x)
            printf("%d\n",x)
    }
```

b.c

```
#include<stdio.h>
main(){
    int x,sum;
    float y;
    scanf("%d", x);
    sum=summation(x,y);
}
```

Exam-2015

Q.What do you mean by structured programming ? write down the general structure of a c program?
Q.Discuss structure of C Programming? Exam: ACCE-2013
Ans:
Structured Programming:
Structured programming is a **programming** paradigm aimed at improving the clarity, quality, and development time of a computer **program** by making extensive use of subroutines, block structures, for and while loops—in contrast to using simple tests and jumps such as the goto statement which could lead to "spaghetti code" .

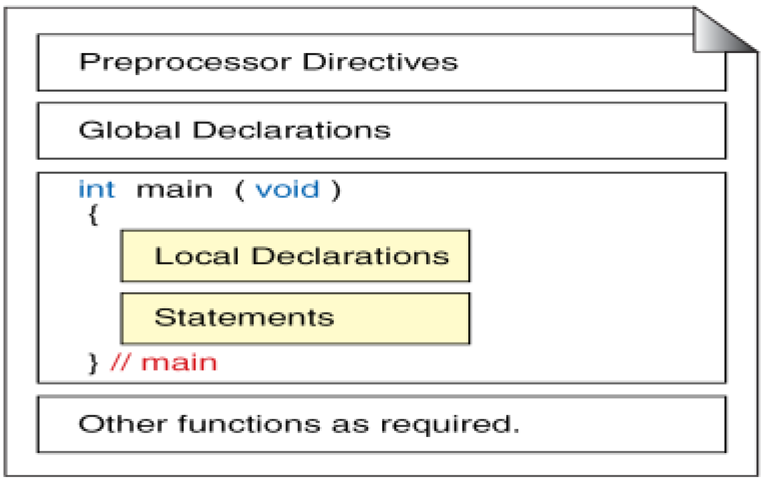


Fig: General Structure of C program.Example:

```
 #include<stdio.h>
int main()
   {
  printf("Hello World");
  return 0;
   }
```

Q.Briefly discuss about a simple c Program?
Ans:

A Simple C Program

Every C program must have one special function main (). This is the point where execution begins when the program is running. The group of statements defining the main () enclosed in a pair of braces ({}) are executed sequentially. Each expression statement must end with a semicolon. The main function can be located anywhere in the program but the general practice is to place it as the first function.
Here is an elementary C program.

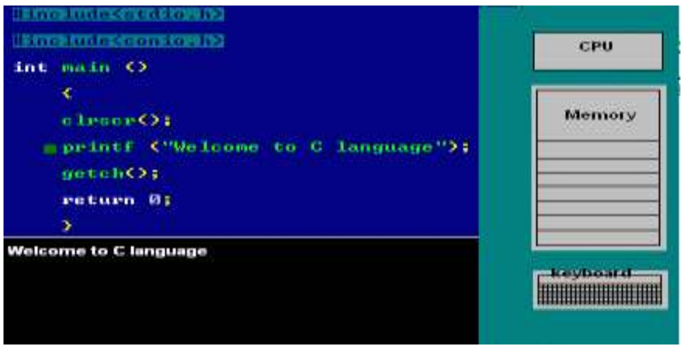
```
main ()
  {
  }
```

There is no way to simplify this program. main ( ) function should return zero or o.
main( ) should be declared as

```
int main( )
 {
  ……………..
  ……………..
 return 0 ;
}
```

.For a much more interesting program, load the program

```
int main ()
   {
    printf ("Welcome to C language");
    return 0;
   }
```



```
#include<stdio.h>
#include<conio.h>
int main ()
    {
    clrscr();
    printf ("Welcome to C language");
    getch();
    return 0;
    }
```

```
Welcome to C language
```

and display it on your monitor. It is same as the previous program except that it has one executable statement between the braces.
Question:What are the general characteristics in C ? Exam-2012

Answer:

General Characteristics In C:

1. Simplicity:- C has richest collection of inbuilt functions, keywords and data types. It also resembles general English language. Also it follows a structured approach so it is to learn also.
2. Clarity:- The keywords and library functions available in C resembles common English words thus it helps to improve the clarity of the program,
3. Portability: – By the term portable we mean computer independent. The program made in C can be made to run on different machines thus increasing the efficiency of the program.
4. Modularity: – The programs made in C can be easily divided into small modules with the use of functions. This feature helps to increase the understandability of the program.
5. Easy Availability: – The C compilers are easily available and they require very less disk space for their storage. It is very easy to load Turbo C complier in your comput

Q.Different the structured and non structured language?
Ans:

Structured Language	**Non structured language**
Code compartmentalization can be done	Code compartmentalization cannot be done
Loops can be created	Loops cannot be created
These are new languages	These are old languages
Do not require a strict field concept.	Strict field concept is used mostly.
Examples: C, C++, JAVA, ADA, PASCAL, MODULA -2	Examples: BASIC, COBOL, FORTRAN

Q.what are the difference between structured programming and object oriented programming?
Acce-2014.CSE,APPE,MSE
Ans:

Structured Programming	**Object Oriented Programming**
Structured Programming is designed which focuses on **process**/ logical structure and then data required for that process.	Object Oriented Programming is designed which focuses on **data**.
Structured programming follows **top-down approach**.	Object oriented programming follows **bottom-up approach**.
Structured Programming is also known as **Modular Programming** and a subset of **procedural programming language**.	Object Oriented Programming supports **inheritance, encapsulation, abstraction, polymorphism**, etc.
In Structured Programming, Programs are divided into small self contained **functions**.	In Object Oriented Programming, Programs are divided into small entities called **objects**.
Structured Programming is **less** secure as there is no way of **data hiding**.	Object Oriented Programming is more secure as having data hiding feature.
Structured Programming can solve **moderately** complex programs.	Object Oriented Programming can solve any **complex** programs.
Structured Programming provides **lessreusability**, more function dependency.	Object Oriented Programming provides more reusability, less function **dependency**.
Less abstraction and less flexibility.	More abstraction and more **flexibility**.

Question: Discuss programming input and output with example?
Ans:
ANSI standard has defined many library functions for input and output in C language. Functions printf() and scanf() are the most commonly used to display out and take input respectively.
#include<stdio.h>//This is needed to run printf() function.

```
int main()
{
   printf("C Programming");//displays the content inside quotation
return0;
}
```

Output

```
C Programming
```

Explanation of How this program works

1. Every program starts from main() function.
2. printf() is a library function to display output which only works if #include<stdio.h>is included at the beginning.
3. Here, stdio.h is a header file (standard input output header file) and #include is command to paste the code from the header file when necessary. When compiler encounters printf() function and doesn't find stdio.h header file, compiler shows error.
4. Code return 0; indicates the end of program. You can ignore this statement but, it is good programming practice to use return 0;.

Q.What is a token?

Ans:

A C program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.

Q.Discuss about Token.

Ans:

In a passage of text individual words and punctuation marks are called tokens.

• In a C program the smallest individual units known as C tokens.

• C has 6 types of tokens namely:

1) Keywords
2) identifiers
3)constants
4)Strings
5)special symbols
6)operator

Question:Correct the following C programs so that you can compile them successfully.

a.c

```
/* Date. 5.3.2015
Main(){
    int i,
    for(i=0,i<5,i++){
            scanf("%d", x)
            printf("%d\n",x)
    }
```

Answer:

Error and Correction:

(1). int i this statement need semiclone(;)

(2). scanf and printf at the end of this two statement semiclone needed.

(3) & need before x in scanf statement, is not compile error this is logical error

b.c

```
#include<stdio.h>
main(){
    int x,sum;
```

```
            float y;
            scanf("%d", x);
            sum=summation(x,y);
        }
```

Exam-2015
Error and Correction:

(1). Value of variable y is not initialized.
Q. What is a NULL statement?
Ans:
A null statement is no executable statements such as ; (semicolon).
Eg:int count =0;
while(++count<=10);Above does nothing 10 times.

2.2 Data Types

1. What is meant by constant and variable? Write down the name of different constant types. **Exam-2014**
2. Name and describe four basic data types in C. **Exam-2012**
3. What is a character constant? How do character constants differ from numeric –type constants? **Exam-2012**
4. Mention and describe the four basic data types in c. **Exam-2013**
5. Do character constants represent numerical values ? **Exam-2012**
6. What is a variable? How can variables be characterized? **Exam-2012**
7. How the value of an expression can be converted to a different data type? Consider the expression x=(y+z) %4 ; if the value of y and z are 5.5 respectively, what will be the value of x? Explain the reason of output. **Exam-2014**
8. Write a function named 'int floatinteger(float n)' to decide whether a number, n, is a floating point or pure integer. Your function will return '1', if n is a floating point number otherwise '0'. **Exam-2015**
9. What will be the output of the following statements? **Exam-2015**

```
printf("%5d\n",123);
printf("%-5d\n",123);
printf("%05d\n",15);
printf("%3.2f\n",3.14159);
printf("%x\n",255);
printf("%o\n",255);
```

10. What will be simplified form of the following code ?

    (Objective of the question: To check the formatting knowledge)

```
 #include<stdio.h>
 #include<conio.h>
main()
{
    printf("%7d\n",123);
    printf("%-4d\n",123);
    printf("%07d\n",15);
    printf("%4.3f\n",3.14159);
    printf("%x\n",127);
    printf("%o\n",127);
    getch()
 }
```

Exam-2016

11. What will be simplified form of the following code ?

    (Objective of the question: To check the formatting knowledge)

```
 #include<stdio.h>
 #include<conio.h>
main()
{
    printf("%7d\n",123);
    printf("%-4d\n",123);
    printf("%07d\n",15);
    printf("%4.3f\n",3.14159);
    printf("%x\n",127);
    printf("%o\n",127);
    getch()
}
```

Exam-21016

Q. Define key words and Identifier ? can you use a key word as an identifier? Exam-2011,ICE,CSE
Ans;
<u>Key Word :</u>

Keywords are the reserved words used in programming. Each keywords has fixed meaning and that cannot be changed by user.
For example:

int *money*;
Here, int is a keyword that indicates, *'money'* is of type integer.
As, C programming is case sensitive, all keywords must be written in lowercase

Points to remember

1. Keywords can be used only for their intended purpose.
2. Keywords can't be used as programmer defined identifier.
3. The keywords can't be used as names for variables.

Identifier:

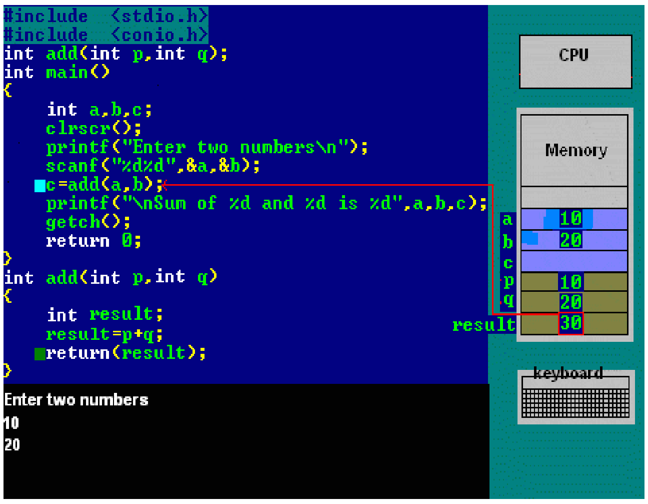
Identifier is a name used to identify a variable, function, or any other user-defined item. An identifier starts with a letter A to Z or a to z or an underscore _ followed by zero or more letters, underscores, and digits (0 to 9).

For example:
int *money*;

int *mango_tree*;

Here, *money* is a identifier which denotes a variable of type integer. Similarly, *mango_tree* is another identifier, which denotes another variable of type integer.

We can not use a key word as an identifier



```
#include  <stdio.h>
#include  <conio.h>
int add(int p,int q);
int main()
{
    int a,b,c;
    clrscr();
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b);
    c=add(a,b);
    printf("\nSum of %d and %d is %d",a,b,c);
    getch();
    return 0;
}
int add(int p,int q)
{
    int result;
    result=p+q;
    return(result);
}
```

```
Enter two numbers
10
20
```

Q.Write the rules of naming variable? Exam –ACCE-2014,ICE-2013,CSE-2011
RULES FOR IDENTIFIERS/variables:
Ans:

(a) First character must be an alphabet.
(b) Must consist of only letters, digits or underscore.
(c) Only first 31 characters are significant.
(d) Cannot use keyword. Must not contain white space.
(e) C does not allow punctuation characters such as @, $, and % within identifiers.
(f) C is a case sensitive programming language.

Here are some examples of acceptable identifiers:
mohd, zara, abc, move,_name, a_123.
myname50 _temp, j, a23b9, retVal.
No we can not use keyword as identifier.

Q. What are the keywords in C ? what restriction apply to their use.
Ans:
Every C word is classified as either a keyword. The list of all keywords of C are listed bellow:

auto	double	int	struct	break	else	long	switch	case	enum
register	typedef	char	extern	return	union	const	float	short	unsigned

continue for signed void default goto sizeof volatile do if static while

All keywords have fixed meaning and these meaning can not be changed . the keyword are all lowercase.

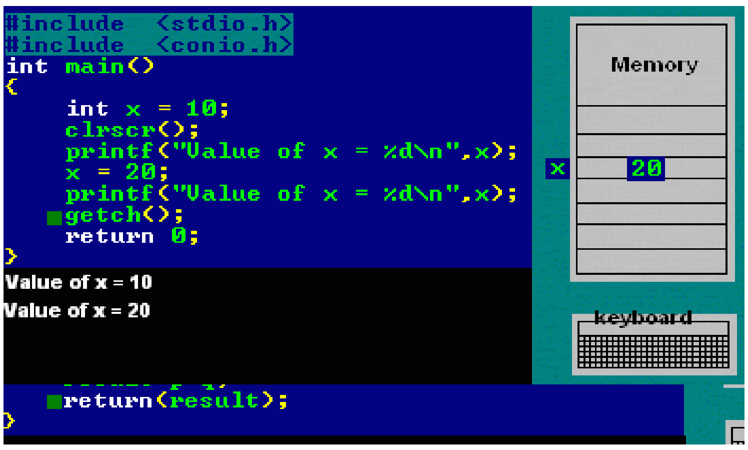
Q. What do you mean by Constant Comments? How constant use in Program?
Ans:
A constant comment is a note that we put into our source code. All comments are ignored by the compiler. Comments are used primarily to document the meaning

1.single line comment start with
//this is comment line
2.Multiple line comment
They start with /* and terminates with the
Characters and end with */ as shown below:
.

We can use a comment to temporarily rmove a line of code.

Qeustion:What is meant by constant and variable? Write down the name of different constant types. Exam-2014,2013 or
Question:Define variable and explain briefly wit example? ICE-2013,2015,APPE-2012
Or
Question:What is a variable? How can variables be characterized? Exam-2012

Answer:
Definition:- Variables are named memory locations that have a type, such as integer or character, which is inherited from their type.
.
Example:
int *num*;
Here, *num* is a variable of integer type.



```
#include  <stdio.h>
#include  <conio.h>
int main()
{
    int x = 10;
    clrscr();
    printf("Value of x = %d\n",x);
    x = 20;
    printf("Value of x = %d\n",x);
    getch();
    return 0;
}
Value of x = 10
Value of x = 20
    return(result);
}
```

Any combination of letters, numbers, and underscore (_)
Case matters

- "sum" is different than "Sum"
- Cannot begin with a number
- usually, variables beginning with underscore
- are used only in special library routines
- Only first 31 characters are used

Legal
i
wordsPerSecond
_ green
Illegal
10sdigit
done?
Double
There are two type of variable:
(a) **Local Variables**
(b) **Global Variables**
Local Variables: Variables that are declared inside a function or block are local variables.

only accessible in a particular region

Global Variables: Global variables are defined outside of all the functions, usually on top of the program.A global variable can be accessed by any function or anywhere

Q.What is constants?

Ans:

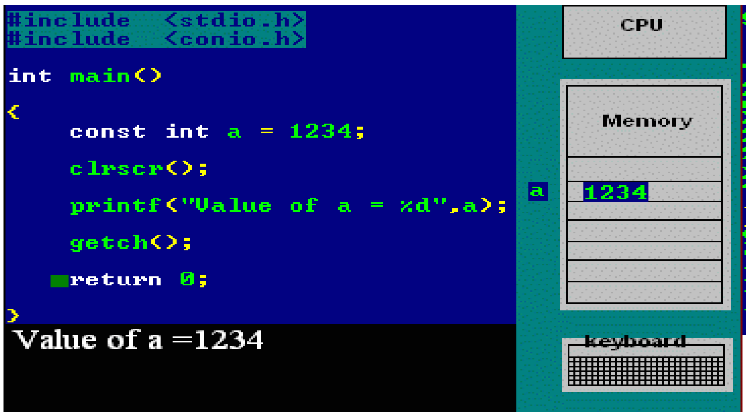
Constants in C refers to fixed value that do no t change during the execution of a program. the constant definitions come before the variable declarations.

-constant –identifier –constant.

There are 4 type of constant in C . they are

(a) Integer constant
(b) Floating constant
(c) Character constant
(d) String constant

Constants are data values that cannot be changed during the execution of a program.



```
#include   <stdio.h>
#include   <conio.h>
int main()
{
    const int a = 1234;
    clrscr();
    printf("Value of a = %d",a);
    getch();
    return 0;
}
Value of a =1234
```

Q.Discuss defferent technique of define constants? Exam-2011

Ans:

There are two technique in C to define constants:

1. **Using #define preprocessor.**
2. **Using const keyword**

1.The Define

#define identifier value

Here The #define is a Preprocessor Following is the form to use #define preprocessor used to define a constant:

Example:

Here LENGTH, WIDTH are identifier name or vaiable name. we can these variable as a constant by defining with #define keyword.

```
#include <stdio.h>
#define LENGTH 10
#define WIDTH 5
#define NEWLINE '\n'
int main()
{
int area;
area = LENGTH * WIDTH;
 printf("value of area : %d", area);
 printf("%c", NEWLINE);
 return 0;
 }
```

2.The const Keyword

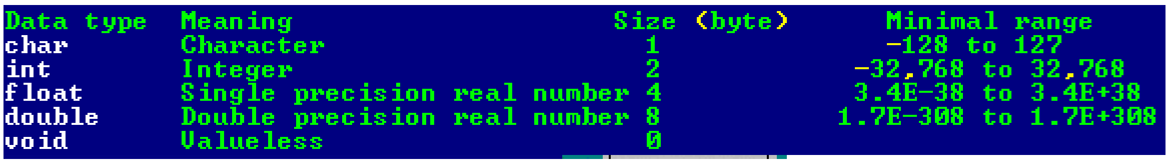
const type variable = value

The const is a Preprocessor Following is the form to use Const preprocessor used to define a constant:
Example:
;
Here LENGTH, WIDTH are identifier name or vaiable name. we can these variable as a constant by defining with const keyword

```
#include <stdio.h>
const int LENGTH = 10;
const int WIDTH = 5;
const char NEWLINE = '\n'
int main()
{
;
int area;
area = LENGTH * WIDTH;
printf("value of area : %d", area);
printf("%c", NEWLINE);
return 0;
}
```

Question:What is a character constant? How do character constants differ from numeric –type constants? Exam-2012
Answer:
Question:Do character constants represent numerical values ? Exam-2012
Question:Name and describe four basic data types in C. Exam-2012
Q.Define Datatype? CSE-
Ans:
All c compilers support five fundamental data typees these are as follows.
Those are:-

(a) Integer
(b) Floating point
(c) Character
(d) Double
(e) Void (valueless)



Data type	Meaning	Size (byte)	Minimal range
char	Character	1	-128 to 127
int	Integer	2	-32,768 to 32,768
float	Single precision real number	4	3.4E-38 to 3.4E+38
double	Double precision real number	8	1.7E-308 to 1.7E+308
void	Valueless	0	

* In C language '%d' is used for Integer.
* '%f' is used for floating point
* '%c' is used for character type data.

Integer Type: The variable of integer type are used to represent the integer type data . Integer type require two bytes of 16 bits of internal storage. Its range is -32768 ro 32767.

Character type: Character set are the set of alphabets, letters and some special characters that are valid in C language. character can be defined as a character are usually stored in 8 bytes of internal storage. Its range is -128 to 127.
Example:
Alphabets:
Uppercase: A B C ………………………… X Y Z
Lowercase: a b c ………………………… x y z
Digits:
0 1 2 3 4 5 6 8 9

Special Characters:

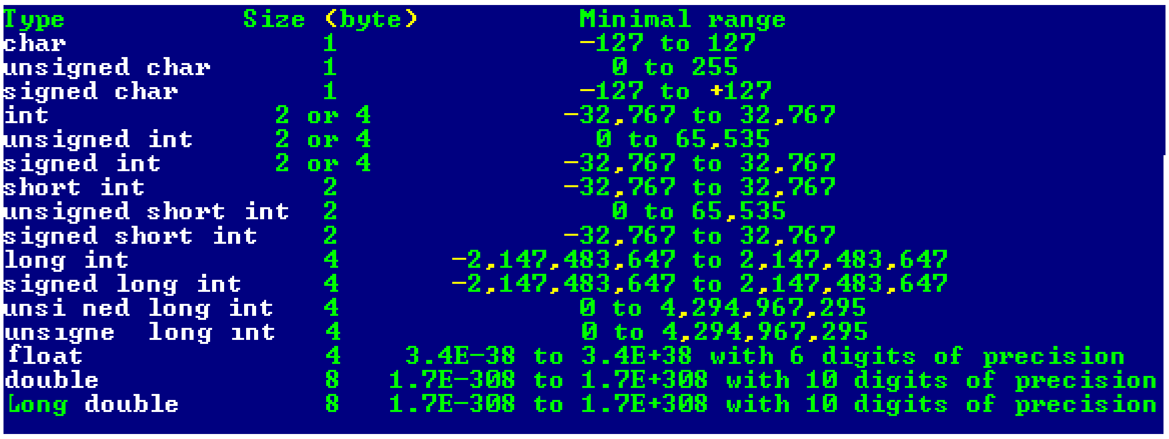
Special Characters in C language														
,	<	>	.	_	(	)	;	$	:	%	[	]	#	?
'	&	{	}	"	^	!	*	/	|	-	\	~	+	

White space Characters:

blank space, new line, horizontal tab, carriage return and form feed

Floating Type: The variable of Floating type are used to represent the floating point type numbers.Integer type require 4 bytes of 32 bits of internal storage. Its range is -3.4 E+38 to 3.4 E-38.
Double Type: The variable of double types are also used to represent the real numbers. But its requirement is double to the floating type that is 8 bytes or 64bits its range is 1.7 E-308 to 1.7 E+308
Void type: The void types has no values . this is usually used to specify the type of functions. The type of a function is said to be void when it does not return any value to calling functions.

Another figure:



```
Type               Size (byte)          Minimal range
char                    1               -127 to 127
unsigned char           1                 0 to 255
signed char             1               -127 to +127
int                  2 or 4            -32,767 to 32,767
unsigned int         2 or 4              0 to 65,535
signed int           2 or 4            -32,767 to 32,767
short int               2              -32,767 to 32,767
unsigned short int      2                 0 to 65,535
signed short int        2              -32,767 to 32,767
long int                4       -2,147,483,647 to 2,147,483,647
signed long int         4       -2,147,483,647 to 2,147,483,647
unsi ned long int       4               0 to 4,294,967,295
unsıgne  long int       4               0 to 4,294,967,295
float                   4     3.4E-38 to 3.4E+38 with 6 digits of precision
double                  8   1.7E-308 to 1.7E+308 with 10 digits of precision
Long double             8   1.7E-308 to 1.7E+308 with 10 digits of precision
```

Example- # include<stdio.h>

```
                    # include< conio.h>
                    Void main ( )
                    {
                    clrscr ( );
                    int i;            //integer type size of I is 2 byte
                    char c;          //character type  size of c is 1 byte
                    float d;         //Floating type  size of d is 8 byte

                    printf ("Enter your number :');
                    scanf ("%d", & i) ;
                    printf ("Enter our character :");
                    scanf ("%c",&c);
               printf ("Enter your floating number :");
                    Scanf ("%f"; & d);
               printf ("you Entered %d %c %f",I, c,d);
                    getch ( );
                    }
```

If anyone wants a floating value just like 4.333 then he has to define it in the

program.
Example:- #include <stdio.h>

```
                                   #include <conio.h>
                                          int main ( )
                                          {
                                          int a, b ;
                                          float c ;
                                          C= (a+b)/4 ;
                                          Print("Enter your 1st number:");
                                          Scanf("%d", & a);
                                          Printf("Enter your 2nd number:");
                                          Scanf("%d", & b);
                                          Printf("in the result is %.2f",c);
                                          Gethc ( );
```

Q.What is different between Float and double variable?
Ans:

Difference between float and double:

Generally the size of float(Single precision float data type) is 4 bytes and that of double(Double precision float data type) is 8 bytes. Floating point variables has a precision of 6 digits whereas the the precision of double is 14 digits.

Q.What is Sizeof operator ? and use of its give with an example?
Ans:

The sizeof operator It is a unary operator which is used in finding the size of data type, constant, arrays, structure etc. For example:

```
#include<stdio.h>
int main(){
int a;
float b;
double c;
char d;
  printf("Size of int=%d bytes\n",sizeof(a));
  printf("Size of float=%d bytes\n",sizeof(b));
  printf("Size of double=%d bytes\n",sizeof(c));
  printf("Size of char=%d byte\n",sizeof(d));
return0;
}
```

Output

```
Size of int=4 bytes
Size of float=4 bytes
Size of double=8 bytes
Size of char=1 byte
```

Q.Discuss I/O integers in C?

Ans:

```
#include<stdio.h>
int main()
{
int c=5;
   printf("Number=%d",c);
return0;
}
```

Output

```
Number=5
```

Inside quotation of printf() there, is a conversion format string "%d" (for integer). If this conversion format string matches with remaining argument,i.e, *c* in this case, value of *c* is displayed.

```
#include<stdio.h>
int main()
{
int c;
   printf("Enter a number\n");
   scanf("%d",&c);
   printf("Number=%d",c);
return0;
}
```

Output

```
Enter a number
4
Number=4
```

I/O of characters and ASCII code

```
#include <stdio.h>
int main(){
   char var1;
   printf("Enter character: ");
   scanf("%c",&var1);
   printf("You entered %c.",var1);
   return 0;
}
```

Output

```
Enter character: g
You entered g.
```

ASCII code

When character is typed in the above program, the character itself is not recorded a numeric value(ASCII value) is stored. And when we displayed that value by using "%c", that character is displayed.

```
#include<stdio.h>
int main(){
char var1;
   printf("Enter character: ");
   scanf("%c",&var1);
   printf("You entered %c.\n",var1);
   printf("ASCII value of %d",var1);
return0;
}
```

Output

```
Enter character:
g
103
```

```
#include<stdio.h>
int main(){
int var1=69;
   printf("Character of ASCII value 69: %c",var1);
return0;
}
```

Output

```
Character of ASCII value 69: E
```

Q.What is ASCII ?
Ans:
abbreviated from **American Standard Code for Information Interchange**,is a character-encoding scheme. ASCII codes represent text in computers, communications equipment, and other devices that use text.
Text information is made of lots of individual units of information called *ASCII characters*
An ASCII character is often just a key stroke on the computer keyboard.
Examples of ASCII Characters :
An ASCII character can be:

- A lower case letter of the English alphabet.
- An upper case letter of the English alphabet.
- A single digit from zero to nine.
- A single space.
- A punctuation mark such as **.** or **,** or **:** or **;** etc
- A math symbol such as **+** or **-** or **/** or **(** or **)** or **=** etc
- A special character such as **#** or **$** or **^** or**|** etc

Q; write a program which reads your name from the keyboard and outputs a listh of ASCII code, which represent your name. Exam:ACCE-2012,11,CSE,APPE
Ans:

```
#include<stdio.h>
int main()
{
int i;
char name[50];
printf("Enter your name: ");
gets(name);
printf("nCharactertASCII Code");
for(i=0;name[i]!=";i++)
printf("n%ctt%d",name[i],name[i]);
return 0;
}
```

Q.What is overflow and underflow?
Ans:
Overflow: *Overflow* is said to occur when the number that one gets as a result of some arithmetic operation on two ***n***-bit binary numbers(signed or unsigned) is larger in **magnitude** than the largest number one can represent using *n*-bits. When adding and subtracting integers, using n-bit 2's complement representation, overflow may occur in 4 different situations, identify these.

or

When stack is full and yet another operation tries to PUSH data onto the stack then a 'stack overflow'

occurs

Underflow:

Talking about underflow makes sense for floating-point numbers. For example using a 4-bit 2's complement representation for an exponent of a floating-point number, would enable us to represent numbers of the order of 10^{-16} to $10^{15.}$ If the number is less than 10^{-16} there would be no way to represent it, and hence it would lead to an "underflow". Underflow in floating-point arithmetic may also be thought of as overflow of the exponent.

or

If the stack is empty and yet a POP operation is attempted, this is called an 'stack underflow'**.**

. **Question:How the value of an expression can be converted to a different data type? Consider the expression x=(y+z) %4 ; if the value of y and z are 5.5 respectively, what will be the value of x? Explain the reason of output. Exam-2014**

Answer:

<u>Value Of An Expression Can Be Converted To A Different Data Type:</u>

Operands of different types can be combined in one operation. For example, the following expressions are permissible:

```
double dVar = 2.5;   // Define dVar as a variable of type double.
  dVar *= 3;         // Multiply dVar by an integer constant.
  if ( dVar < 10L )   // Compare dVar with a long-integer constant.
    { /* ... */ }
```

When the operands have different types, the compiler tries to convert them to a uniform type before performing the operation. In certain cases, furthermore, you must insert type conversion instructions in your program

<u>Example:</u>

We can also convert values from one type to another explicitly using the cast operator

(type_name) expression

In the following example, the cast operator causes the division of one integer variable by another to be performed as a floating-point operation:

```
  int sum = 22, count = 5;
  double mean = (double)sum / count;
```

Because the cast operator has precedence over division, **the value of sum in this example is first converted to type double. The compiler must then implicitly convert the divisor, the value of count, to the same type before performing the division.**

<u>Expression x=(y+z) %4:</u>

If the value of y and z are 5.5 respectively, will be the value of x is
5.5+5.5=11
X=11%4=3

Because % means remainder, here reminder is 3;

Question:Write a function named 'int floatinteger(float n)' to decide whether a number, n, is a floating point or pure integer. Your function will return '1', if n is a floating point number otherwise '0'. Exam-2015

Answer:

```
int floatinteger(float n)
 {
```

```
if (n-(int)n == 0)
     printf("1\n");
else printf("0\n");

}

int main()
  {

float f = 4.5886;
          floatinteger(f);
}
```

Question: What will be the output of the following statements? Exam-2015

```
#include<stdio.h>
 #include<conio.h>
main()
{
printf("%5d\n",123);
printf("%-5d\n",123);
printf("%05d\n",15);
printf("%3.2f\n",3.14159);
printf("%x\n",255);
printf("%o\n",255);
getch();
}
```

Output-1:

		1	2	3

Output-2:

1	2	3

Output-3:

0	0	0	1	5

Output-4:

3	.	1	4

Output-5: integer-255= hexadecimal-ff
Output-6: Integer -255 =Octal-377
Question:What will be simplified form of the following code ?

(Objective of the question: To check the formatting knowledge)

```
#include<stdio.h>
#include<conio.h>
main()
{
   printf("%7d\n",123);
   printf("%-4d\n",123);
   printf("%07d\n",15);
   printf("%4.3f\n",3.14159);
   printf("%x\n",127);
   printf("%o\n",127);
   getch();
}
```

Exam-2016

Answer:
Output-1:

				1	2	3

Output-2:

1	2	3

Output-3:

0	0	0	0	0	1	5

Output-4:

3	.	1	4	2

Output-5: integer-127= hexadecimal-7F
Output-6: Integer -127 =Octal-177

2.3 Operator

1. What is an operator? Describe several different types of operators that are included within the C language. **Exam-2012**
2. Distinguish between unary and binary operator with example. **Exam-2014**
3. Explain bitwise AND and OR operation with example. If x=5; y=11; what is the value of (x&y|2) and (x|y==3)? **Exam-2014**
4. What is the difference between 'i++' and '++i'? Explain with proper example. **Exam-2012**
5. Write a C program that will take integer as input with the following condition. **Exam-2012**
   i. First two input there are no checks.
   ii. From third input, it will check following conditions
   a.Program will terminate, if the input is greater than any of the preceding two taken input
   b.Program will terminate if the input is less than the difference between last two proceeding input.
6. If int i = 7 , float f=5.5, char c=a, What will the output of (a) 'I + c' and (b) 'I + f' **Exam-2016**
7. If int result, i = 7, f =8.5, What will the output of 'result = (i + f) %4'. **Exam-2016.**
8. If float num = 10.5, What will the output of 'num % @2' and '((int)num)% 2' **Exam-2016.**
9. What will be simplified form of (a) !(a<b) , (b) !(c<=d), (c) !(x=>y) ? **Exam-21016**
10. Find and explane the output of the following program: Exam-2014

```
void main()  {
        int a=5 , b=15, r, s;
        r=a<8;
        s=(a<10)&&(b=12);
       printf("r=%d, s=%d", r, s);  }
```

11. Find the output of the following code**. Exam-2014**

```
void main()  {
      int i=10, j=20 ;
      float a, b, c ;
      a = i / j ;
      b = 1.0  * i / j ;
      c = i / j * 1.0 ;
      printf("%f %f %f " , a , b , c);  }
```

12. Explain the output of the following block of C code**: Exam-2014**

```
void main()  {
      int  i=4, j ;
      j = ++i * i++ ;
      i *= j ;
      printf("%d %d ", i,j);  }
```

Question:What is an operator? Describe several different types of operators that are included within the C language. Exam-2012

Q.Describe the different types of operator with suitable example?
Ans:
There are eight types of operator.Example :
1.Arithmatic operator
2.Relational operator/Comparison operator
3. Logical operator
4. Assignment operator
5.Increment operator and Decrement operator.
6.conditional operator.
7.Bitwies operators.
8.Special operator.

(a) **Arithmetic operator :-** There are some arithmeticaloperators which are given below :
i) "+" for addition
ii) "-" For subtraction
iii)"*" for multiplication
iv) "/" for division
v) "%" for remain or modulus division
vi)"--" for decrement
vii) "++" for increment

(b) **Relational operator :-** The following relational operators are in C language :
i) ">" grater than
ii) "<" less than
iii) "<=" less than or equal
iv) ">=" grater than or equal
v) "==" equal
vi) "!=" not equal
Relational operators are also called on comparison operator.

(c) **Logical operator** :- This operator is used for formal logic.There are some logical operators are in C language.Those are:
i)"&&" AND
ii) "||" OR
ii) "! " NOT

(d)Assignment Operator:
or
Q.Replace +,-,/,* and = with equivalent statement using only ++,--,and +=,*=,/=%. Exam:ACCE-2012
(i) a=a-3; Ans:a-=2;
(ii) a=a*2 Ans: a*=2;
(iii) a=a/4 Ans: a/=4;
(iv) a=a%2 Ans:a%=2;
(v) b=b+(c+2) Ans: b+=c+2;
(vi) d=d*(n-5) Ans: d*=n-5;

(E)Bitwise Operator:

&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left

Q.Determine the output of the program.

```
Int amount,count;
Count=3;
Amount=2*++count;
Printf("count=%d\n",count,amount);
```
Ans: count=4;

(F)Conditional Operator:
or
Q. What is conditional operator. Give with an example?
Ans:
Conditional operators (?:)
Conditional operators are used in decision making in C programming, i.e, executes different statements according to test condition whether it is either true or false.
Syntax of conditional operators

conditional_expression?expression1:expression2

If the test condition is true, expression1 is returned and if false expression2 is returned.
Example of conditional operator

```
#include<stdio.h>
int main(){
char feb;
int days;
  printf("Enter l if the year is leap year otherwise enter 0: ");
  scanf("%c",&feb);
  days=(feb=='l')?29:28;
/*If test condition (feb=='l') is true, days will be equal to 29. */
/*If test condition (feb=='l') is false, days will be equal to 28. */
  printf("Number of days in February = %d",days);
return0;
}
```

Output

```
Enter l if the year is leap year otherwise enter n: l
Number of days in February = 29
```

Qeustion:Distinguish between unary and binary operator with example. Exam-2014
Answer:

Unary Operator	**Binary Operator**
The operator which operate one single operand known as unary operator.	The operator which operate on two operands known as binary operator.

Some of the unary operator :	**Some of the binary operator :**
++ increment operator -- decrement operator & address of operator - unary minus operator ~ negation operator ! logical not	+ binary plus operator - binary minus operator == equal to operator < less than operator

Question:Explain bitwise AND and OR operation with example. If x=5; y=11; what is the value of (x&y|2) and (x|y==3)? Exam-2014
Answer:

If x=5; y=11;
The value of (x&y|2) =3
And
The value of (x|y==3)=5

Q.Compare between pre increment and post increment operator with an example?
Or What is the difference between ++X and X++ ? give example Exam:ACCE-2011,CSE-ICE,APPE
Question: What is the difference between 'i++' and '++i'? Explain with proper example. Exam-2012

++X	X++
1.if the increment operator is fixed in a variabl. X a prefix (++x) on the right hand side of an assignment statement.	1.If the increment operator is fixed in a variabbe. X as postfix (++x) on the right hand side of an assignment statement.
2.the compiler at first add one with the value of X and the result of adding in used in the statement.	2. the compiler at first uses the value of x in th statement and add one with the valu in the nex execution.

X=5
Y=++X
In theis case value of Y and X are both 6
	X=5
	Y=X++;
	In this case the value of Y is 5 and X is 6.

Question: Write a C program that will take integer as input with the following condition. Exam-2012

i. First two input there are no checks.
ii. From third input, it will check following conditions
iii. Program will terminate, if the input is greater than any of the preceding two taken input
Iv. Program will terminate if the input is less than the difference between last two proceeding input.

Answer:
(i).

```
#include<stdio.h>
#include<conio.h>
int main()
 {
```

```
int a,b
printf("Enter two number:");
scanf("%d %d",&a, &b);
Printf("a=%d,b=%d",a,b);
getch();
}
```

(ii).

```
#include<stdio.h>
#include<conio.h>
int main()
 {
 int a,b,c;
 printf("Enter two number:");
 scanf("%d %d",&a, &b);
 Printf("a=%d,b=%d",a,b);
 scanf("%d",&c);
 if(c>a||c>b)
  {
  printf("Terminate for first Condition:");
   return 0;
  }
 if(c<abs(a-b))
  {
  printf("Terminate for Second Condition:");
  return 0;
  }
 getch();
 }
```

Question: If int i = 7 , float f=5.5, char c=a, What will the output of (a) 'I + c' and (b) 'I + f' Exam-2016

```
#include<stdio.h>
#include<conio.h>
int main()
 {
 int i=7;
 float f=5.5;
 char c='a';
 printf("Output=%c\n",i+c);
 printf("Output=%c\n",i+f);
 getch();
 }
```

Answer:

(a) i+c where c=a, a=65(from ascii code)
7+65=72
=h
(b) Null

Question: If int result, i = 7, f =8.5, What will the output of 'result = (i + f) %4'. Exam-

2016.

```
#include<stdio.h>
#include<conio.h>
int main()
 {
 int result,i=7;
  float f=8.5;
  result=(i+f)%4;

  printf("%d",result);
  getch();
 }
```

Answer:
invalid operands of types `float' and `int' to binary `operator%'

Question: If float num = 10.5, What will the output of 'num % 2' and '((int)num)% 2' Exam-2016.
Answer:

- **num%2=invalid operands of types `float' and `float' to binary `operator%'**
- **((int)num)% 2=0;**

Question: What will be simplified form of (a) !(a<b) , (b) !(c<=d), (c) !(x=>y) ? Exam-21016
Answer:
Simplified of :

(a) !(a<b	a>=b
(b) !(c<=d)	c>d
(c) !(x=>y)	x<y

Question: Find and explain the output of the following program: Exam-2014

```
void main()  {
        int a=5 , b=15, r, s;
        r=a<8;
        s=(a<10)&&(b=12);
       printf("r=%d, s=%d", r, s);  }
```

Answer:
r=1, s=1;

Question: Find the output of the following code. Exam-2014

```
    void main()  {
          int i=10, j=20 ;
          float a, b, c ;
          a = i / j ;
          b = 1.0  * i / j ;
          c = i / j * 1.0 ;
          printf("%f %f %f " , a , b , c);   }
```

Answer:
a=0.00000, b=0.0000, c=0.0000

Question: Explain the output of the following block of C code: Exam-2014

```
void main()  {
        int  i=4, j ;
        j = ++i * i++ ;
        i *= j ;
        printf("%d %d ", i,j);  }
```

Answer;

```
;i=150
;j=25
```

Q.Are there any problems with performing mathematical operations on different variable types?

Ans:

C has three categories of built-in data types:
pointer types, integer types, and floating-point types.

Pointer types are the most restrictive in terms of the operations that can be performed on them. They are limited to - subtraction of two pointers, valid only when both pointers point to elements in the same array. The result is the same as subtracting the integer subscripts corresponding to the two pointers. + addition of a pointer and an integral type. The result is a pointer that points to the element which would be selected by that integer. Floating-point types consist of the built-in types float, double, and long double. integer types consist of char, unsigned char, short, unsigned short, int, unsigned int, long, and unsigned long. All of these types can have the following arithmetic operations performed on them:
+ Addition
- Subtraction
* Multiplication
/ Division

Integer types also can have those four operations performed on them, as well as the following operations:
% Modulo or remainder of division
<<>> Shift right
& Bitwise AND operation
Bitwise OR operation
^ Bitwise exclusive OR operation
! Logical negative operation
~ Bitwise “one’s complement” operation

Q.Identify errors in the following program. After correction what output would you expect when you execute it?

Ans:

```
#include<stdio.h>
#define PI 3.1416
int main()
  {
  int R,C;
  float perimeter;
  float area;
  C=PI;
  R=5;
  Perimeter=2.0*C*R;
  Area=C*R*R;
  printf("f","%d",&perimeter, &area);
  return 0;
  }
```

Ans:

There are many error in this program.

- We assigned Perimeter variable, but we declare perimeter. So Perimeter undeclared
- Area but we declared area. Area undeclared.
- In the printf function f error causes to print write %f and must under " " and area is floating type but print as a integer type.
- At output & perimeter &area these are logical error, print address of these variable.

After correction error's of this program :

```
#include<stdio.h>
#define PI 3.1416
int main()
  {
  int R=0,C=0;
  float perimeter=0;
  float area=0;
  C=PI;
  R=5;
  perimeter=2.0*C*R;
  area=C*R*R;
  printf("%f %f", perimeter, area);
  return 0;
  }
output is: Perimeter=30 and area=75
```

Q. if a=15 and b=10 what will be the value of c in the following expression?

Ans:

C=++a-b **Ans**:6

C=b+++a; **Ans**:25

Q.Evaluate each of the following expression (The expression are to be evaluated independent of one naother another)

```
int i=3,j=4,k=2;
i. i--;j++
Ans:3,4
(ii).k++*--i;
Ans:4
(iii) j+1/i+1
Ans:2
(iv) k+=k+i;j++
 k=5;j=4;
```

Q.Write a program to read the radius of a circle and compute its area and circumference. Exam:ACCE 2011,APPE,MSE,CSE;

Ans:

```
#include<stdio.h>
int main()
{
int r;
float cir, area;
const int PI=3.1427;
printf("\n ENTER THE RADIUS OF THE CIRCLE: ");
scanf("%d", &r);
cir = 2*PI*r;
area=PI*r*r;
printf("\n THE CIRCUMFERENCE OF CIRCLE IS= %f", cir);
printf("\n THE AREA OF CIRCLE IS= %f", area);
return 0;
}
OUTPUT:
ENTER THE RADIUS OF THE CIRCLE: 5
THE CIRCUMFERENCE OF CIRCLE IS= 31.427
THE AREA OF CIRCLE IS= 78.5675
```

Q. The following is a segment of a program:

```
int main()
 {  int x=1,y=1,n=0;
 if(n>0)
      x=x+1;
      y=y+1;
    printf("%d %d",x,y);
 }
```

What will be the values of x and yif n assumes a vlalue of (a) 1 and (b) 0.

Ans:when n=1 ,x=2,y=2 when n=0, x=1,y=2;

Q.Assuming x=10, state whether the following logical expressions are true or false.

(i) x==10&&x>10&&!x Fasle
(ii) x==10||x>10||!x   True
(iii) x==10&&x>10||!x False
(iv) x==10||x>10||!x   False

Q. Analyze each of the following segmensts that follow and determine that follow and determine how many times the body of the each loop will execute.

```
(i).int main()
{
 int x=5,y=5;
 while(x<=y)
  {
   x=u/x;
  }
  return 0;
}
```

Ans: Error; Because U is undeclared

(ii). for(i=0;i<=5;i=i+2/3)
  infiniete loop,value of i is always 0;

Q.Identify errors, if any , in each of the following array declaration statements, assuming theat ROW and COLUMN are declared as symbolic constant:

(iii)int score(100);

Ans: becasue score is not built in function , here absence declared score function. or its not an array;

(iv)float value[10,1.5] ;

Ans: if value as an array ,] must expect before , token operato. we cauld not assigned value of array in this procedure.

(v) float average[ROW],[COLUMN];

Ans: if its a two dimensional array " , " token is not accepts between two dimensional array.

(vi)int number[]={0,0,0,0,0},

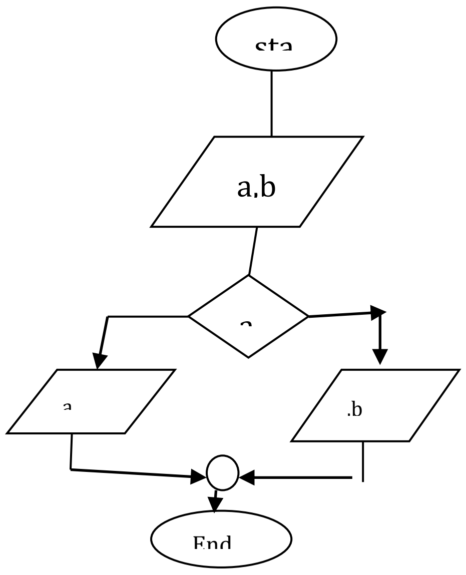
Ans: its right procedur but , is not accept here.expect ; here.

(vii)float item[3][2]={0,1,2,3,4,5};

Ans: here no any physical erro but here logical error. that is item[][] is a two dimensional array , we assign here one dimensional value.

(viii) float result[10]=0;

Ans: invalid initializer valid is float result[10]={0};



Q.Write a flowchart for find largest number from 2 number?

Q.Write a program to find gretest number from two number?

Ans:

```
#include <stdio.h>
main()
{
int i,j,big;
scanf("%d%d",&i,&j);
);
if(i > j)
  {
  printf("biggest of two numbers is %d \n",i);
  }
 else
  {
    printf("biggest of two numbers is %d \n",j);
  }
}
```

Q.Describe increment and decrement operator with example?Marks:1.25 **Exam:ACCE-2014**

Increment operators (++): Increment operators are increase the value of subsequent. value may be increase according to theprogrammer.

- Increment operator are two types as follows :
  1. Post increment
  2. Pre increment

Decrement operators (--)decrement operators decrease the value to one, two and so on.

- As like Increment operators, decrement operators are also two type as :
  1. Post decrement

2. Pre decrement

Before we more discuss about increment and decrement operator, let understand some expression; that's are using in all operators.

Q .S++ or S = S+1, which can be recommended to increment the value by 1 and why?

S++, as it is single machine instruction (INC) internally.

2.4 DECISION MAKING

Important Question

1. What do you know about lvalue and rvalue? Explain with example? **Exam-2013**
2. How the following statements are interpreted? **Exam-2014**
   ```
     if e1 if e2 s1
    else s2
   ```
   Which logical expression is associated with else clause?
3. What is the purpose of the switch statement ? How does this statement differ from the other statements ?**Exam-21016**
4. Draw the flowchart that reads five numbers as input from the user and prints whether the numbers are odd or even (Do not use the modulus operator for this problem). **Exam-2012**
5. Draw the flow chart to find out the 'biggest number' from given 10 integers . **Exam-21016**

Q.What do yu mean by control instruction? what are the types of control instructions? Marks:3 Exam-ACCE-2013
Q. What do you mean by control statement? Exam:ICE-2015,APPE
Answer:
Control Statement:
"Decision making and branching" is one of the most important concepts of computer programming.
Programs should be able to make logical (true/false) decisions based on the condition provided.
C language posseses such decision making capabilities by supporting the following statements :-

(a) if statement
(b) Switch statement
(c) Conditional statement
(d) Go to statement

These statements are popularly known as decision making statements . since these statements control control statements.

Q.**What do you mean by if statement?**
Ans:
The if statement is a powerful decision making statement and is used to control the flow of execution of statements . it is basically a two-way decision statemenet and is used in conjunction with an expression.

The general form of a simple if statement is ,

```
                    If(test expression)
{ statement –blockexecuted if test expression is true;

                 }
                Statement_x;
```

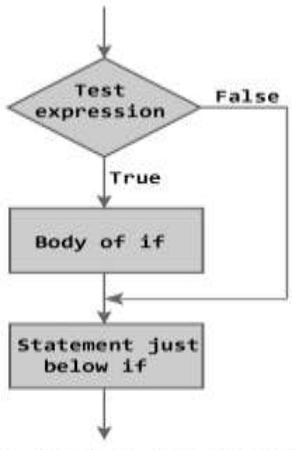


Figure: Flowchart of if Statement

The statement block may be a single statement or a group of statements. If the test expression is true. The statement-block will be executed; otherwise the statement-blockwill be skipped and the execution will jump to the statement_x . Remember , when the condition is true both the statement-block and the statement_x are executed in sequence.
For an example, consider the following program-

```
Int main()
  {
    Int a,b;
Printf("Enter the value of a: ");
Scanf("%d",&a);
Printf("Enter the value of b: ");
Scanf("%d",&b);
If(a>b)printf("The bif number is =%d",a);
}
```

Q. What do you mean by else-If statement? Exam:ICE-
Ans:
The if-else statement is extension of the simple if statement . the general form is-

```
 If(test expression)
{ True-block statement(s)
}
Else {
  False-block statement (s)
}
Statement_x
```

If the test expression is true then the true-block statement(s) immediately following the if statements are executed. In either case, either true-block or false-block will be executed not both

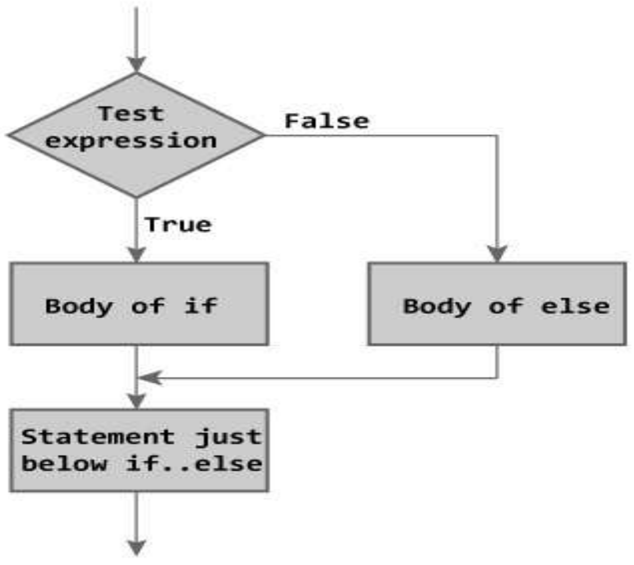


Figure: Flowchart of if...else Statement

Let us consider an example is –

```
#include <stdio.h>
 int main()
 {
   printf("Enter mark: ");
scanf("%d",&mark);
if(mark>=33)
printf("Pass");
else
 printf("Fail");
return 0
}
```

Q. Describe Nested if...else statement ?
Ans:
if...elseif....else Statement:
The nested if...else statement is used when program requires more than one test expression.
Syntax of nested if...else statement.

```
if (test expression1){
    statement to be executed if test expression1 is true;
    }
    else if(test expression2) {
        statement to be executed if test expression1 is false and 2 is
true;
    }
    else if (test expression 3) {
       statement to be executed if text expression1 and 2 are false and
3 is true;
    }
  .
      .
      .
    else {
         statements to be executed if all test expressions are false;
     }
```

Q.How nested if...else works?
Ans:
The nested if...else statement has more than one test expression. If the first test expression is true, it executes the code inside the braces{ } just below it. But if the first test expression is false, it

checks the second test expression. If the second test expression is true, it executes the statement inside the braces{ } just below it. This process continues. If all the test expression are false, code/s inside else is executed and the control of program jumps below the nested if...else
The ANSI standard specifies that 15 levels of nesting may be continued.

Write a C program to relate two integers entered by user using = or > or < sign.

```
#include <stdio.h>
int main(){
    int numb1, numb2;
    printf("Enter two integers to check\n");
    scanf("%d %d",&numb1,&numb2);
    if(numb1==numb2) //checking whether two integers are
equal.
        printf("Result: %d = %d",numb1,numb2);
    else
      if(numb1>numb2) //checking whether numb1 is greater
than numb2.
        printf("Result: %d > %d",numb1,numb2);
      else
        printf("Result: %d > %d",numb2,numb1);
return 0;
}
```

Output 1

```
Enter two integers to check.
5
3
Result: 5 > 3
```

Output 2

```
Enter two integers to check.
-4
-4
Result: -4 = -4
```

Question:What do you know about lvalue and rvalue? Explain with example? Exam-2013

Lvalue:

An lvalue can appear on the left side of an assignment operator.

Rvalue:

An rvalue can appear on the right side.

Example:

As an example:

```
int a;
a = 3;
```

In the second line, "a" is the lvalue, and "3" is the rvalue.
And in this example:

```
int a, b;
a = 4;
b = a;
```

In the third line of that example, "b" is the lvalue, and "a" is the rvalue, whereas it was the lvalue in line 2. This illustrates an important point: An lvalue can also be an rvalue, but an rvalue can never be an lvalue.

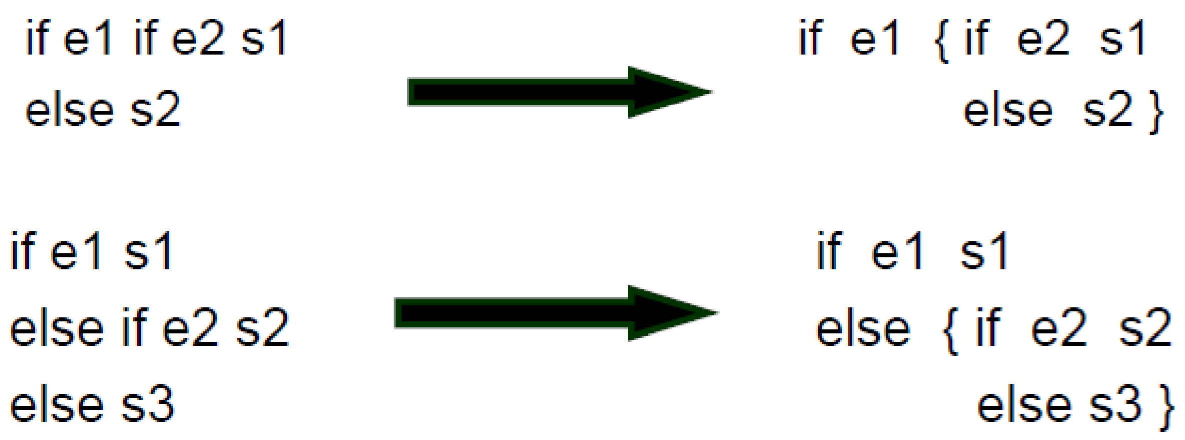
Another definition of lvalue is "a place where a value can be stored." This means certain pointer expressions are also valid lvalues:

```
int *p, *q;
```

```
p = 65000;     /* valid lvalue assignment */
p + 4 = 18;    /* invalid - "p + 4" is not an lvalue */
q = p + 4;     /* valid - "p + 4" is an rvalue */
*(p + 4) = 18;  /* valid - dereferencing pointer expression gives an lvalue */
```

Remembering the mnemonic, that lvalues can appear on the left of an assignment operator while rvalues can appear on the right, will help you keep it straight.

Question:How the following statements are interpreted? Exam-2014
if e1 if e2 s1
else s2
Which logical expression is associated with else clause?



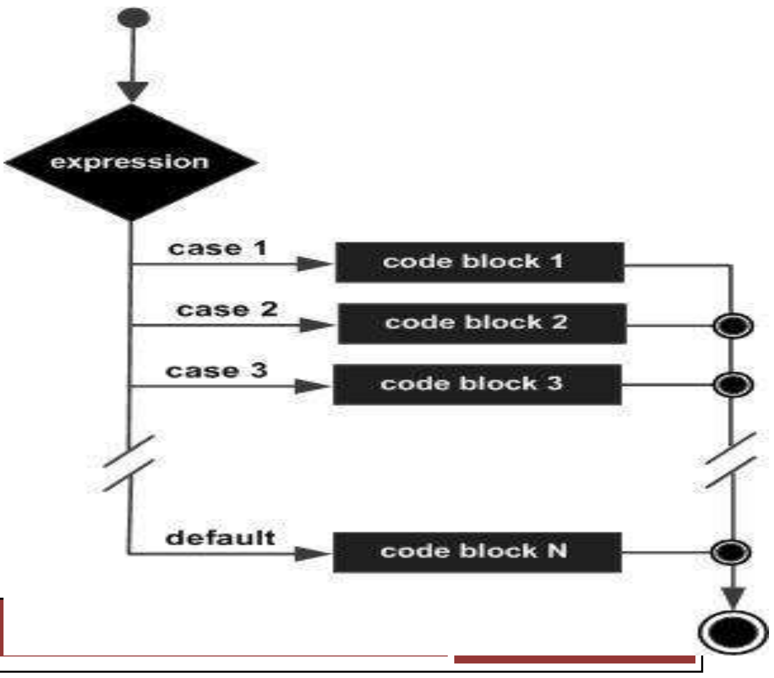
```
if e1 if e2 s1          if e1 { if e2 s1
else s2                        else s2 }

if e1 s1                if e1 s1
else if e2 s2           else { if e2 s2
else s3                        else s3 }
```

Q.Explain the switch case statement with example? Marks:2.5 Exam-MSE,ACCE-2014,13
Q.What is switch statement?
Q.Define Switch Statement?
Ans:

In computer programming languages, a **switch statement** is a type of selection control mechanism used to allow the value of a variable or expression to change the control flow of program execution via a multiway branch.

.

Flow Diagram:



```
public class SwitchDemo {
  public static void main(String[] args) {

    int month = 8;
    String monthString;
    switch (month) {
      case 1:  monthString = "January";
               break;
      case 2:  monthString = "February";
               break;
      case 3:  monthString = "March";
               break;
      case 4:  monthString = "April";
               break;
      case 5:  monthString = "May";
               break;
```

```
            case 6:  monthString = "June";
                     break;
            case 7:  monthString = "July";
                     break;
            case 8:  monthString = "August";
                     break;
            case 9:  monthString = "September";
                     break;
            case 10: monthString = "October";
                     break;
            case 11: monthString = "November";
                     break;
            case 12: monthString = "December";
                     break;
            default: monthString = "Invalid month";
                     break;
        }
        System.out.println(monthString);
    }
}
```

Q. Write the advantage of Switch statement? ICE-

Ans:

<u>Advantage of Switch Statement</u>:

- ➢ A switch statement is a type of control statement that exists in most modern imperative programming languages (e.g., Pascal, C, C++, C#, and Java).
- ➢ Its purpose is to allow the value of a variable or expression to control the flow of program execution.
- ➢ In most languages, a switch statement is defined across many individual statements.

Question:What is the purpose of the switch statement ? How does this statement differ from the other statements ? Exam-21016

<u>Purpose of Switch Statement:</u>

- 🕮 A switch statement is a type of selection control mechanism used to allow the value of a variable or expression to change the control flow of program execution via a multiway branch.
- 🕮 Switch statements are used when you clearly know what are possible values for the condition variable.
- 🕮 Each value in that list of possible value is called case.
- 🕮 When the value given in input matches the value in the case statement, the block of code below case gets executed until it reaches the break statement.

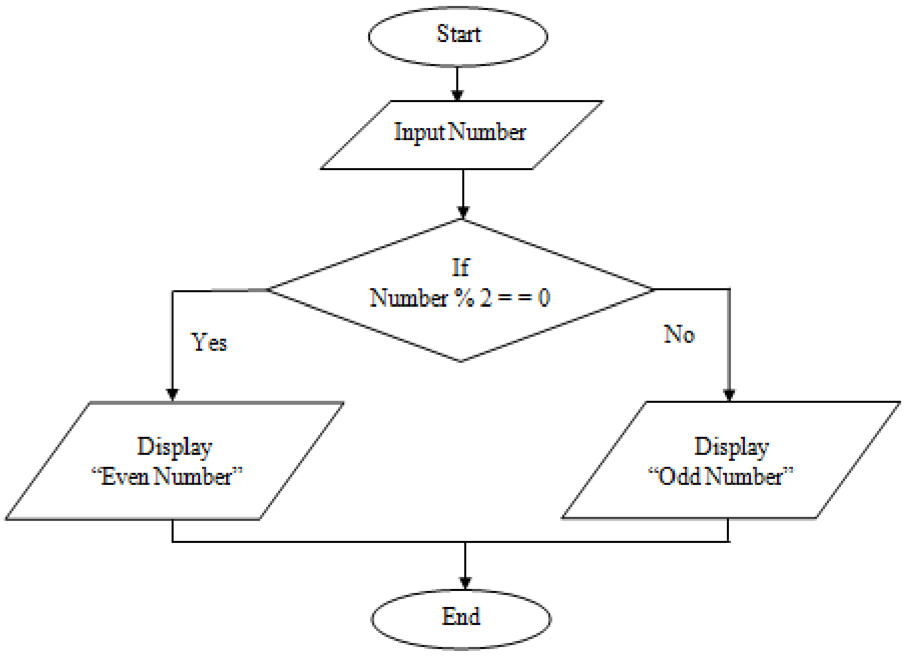
Break is optional. If break statement is not given, the next case statement (if any) will also get executed.

<u>Different between Switch case and if else statement:</u>

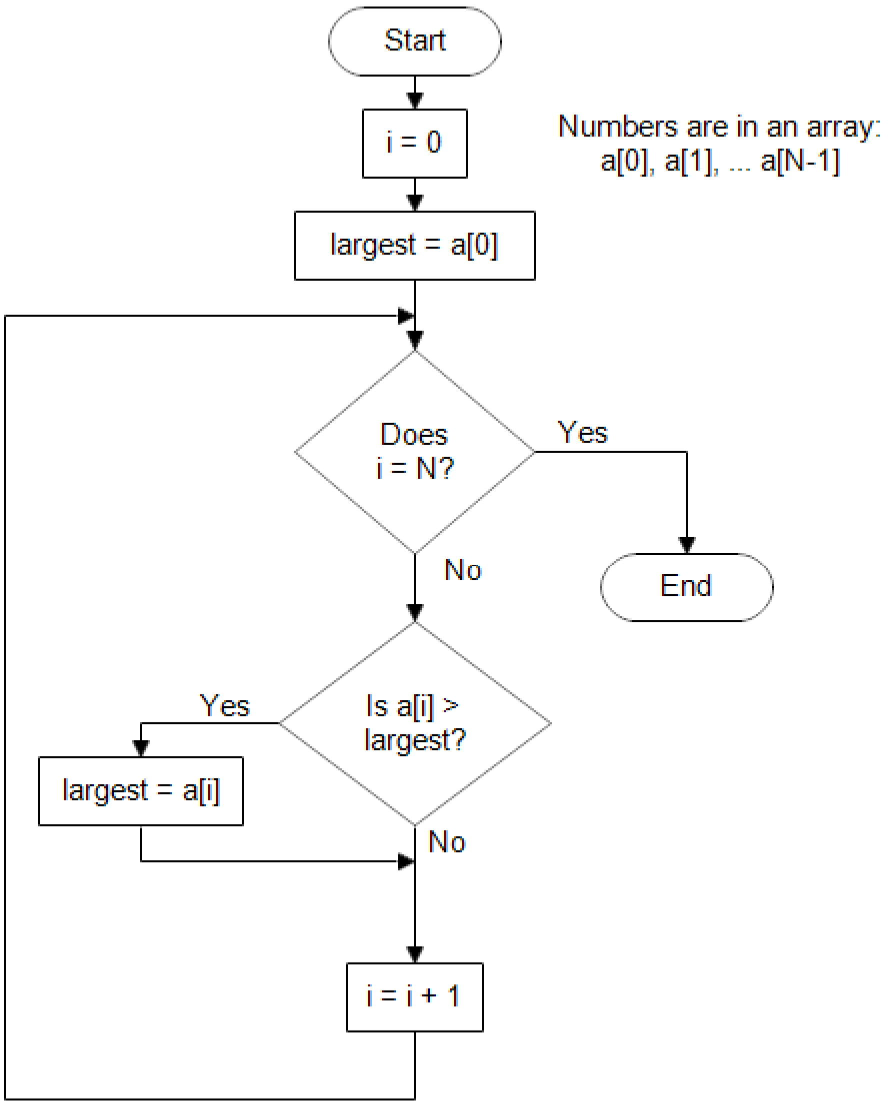
Subject	If Else Statement	Switch Statement
Basic	Which statement will be executed depend upon the output of the expression inside if statement.	Which statement will be executed is decided by user.

Subject	If Else Statement	Switch Statement
Expression	if-else statement uses multiple statement for multiple choices.	switch statement uses single expression for multiple choices.
Testing	if-else statement test for equality as well as for logical expression.	switch statement test only for equality.
Evaluation	if statement evaluates integer, character, pointer or floating-point type or boolean type.	switch statement evaluates only character or integer value.
Sequence of Execution	Either if statement will be executed or else statement is executed.	switch statement execute one case after another till a break statement is appeared or the end of switch statement is reached.
Default Execution	If the condition inside if statements is false, then by default the else statement is executed if created.	If the condition inside switch statements does not match with any of cases, for that instance the default statements is executed if created.
Editing	It is difficult to edit the if-else statement, if the nested if-else statement is used.	It is easy to edit switch cases as, they are recognized easily.

Question:Draw the flowchart that reads five numbers as input from the user and prints whether the numbers are odd or even (Do not use the modulus operator for this problem). Exam-2012



Question:Draw the flow chart to find out the 'biggest number' from given 10 integers . Exam-21016


Start
i = 0
Numbers are in an array:
a[0], a[1], ... a[N-1]
largest = a[0]
Does
i = N?
Yes
End
No
Is a[i] >
largest?
Yes
largest = a[i]
No
i = i + 1

Q.Write a C program to find largest number from 3 number using if...else statement?
Ans:

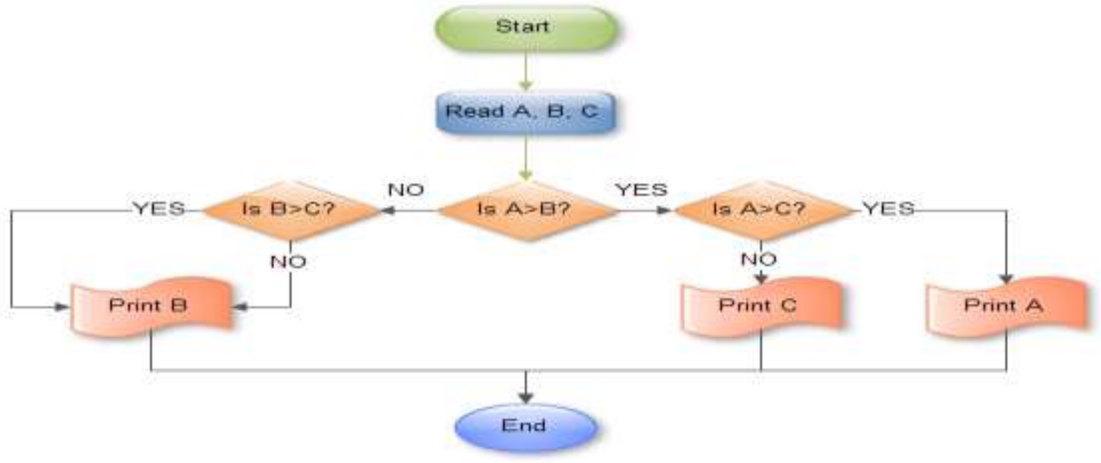
```
#include<stdio.h>
int main(){
    int  a, b, c;
     printf("Enter three numbers: ");

  scanf("%d %d %d",&a,&b,&c);
if(a>=b)
{
if(a>=c)
          printf("Largest number = %d ",a);
else
          printf("Largest number = %d ",c);
}
else
{
if(b>=c)
          printf("Largest number = %d ",b);
else
          printf("Largest number = %d ",c);
}
return0;
}
```

Q. Write a c Program that reads marks of a students and computes and displays grade using if else ,Exam:ACCE
Ans:

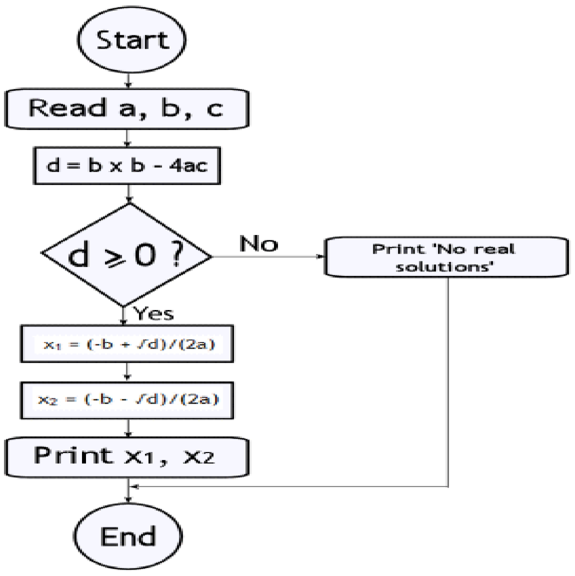
```
#include<stdio.h>
int main ()
    {
    int  a ;
    printf("Please Enter the course result marks: ");
    scanf("%d", &a);
if (a>=80)
    {
    printf("The GPA is: A+");
    }
    else if (a>=75 && a<=79){
    printf("The GPA is: A");}
    else if (a>=70 && a<=74){
    printf("The GPA is: A-");
    }
    else if (a>=65 && a<=69){
    printf("The GPA is: B+");}
    else if (a>=60 && a<=64){
    printf("The GPA is: B-");}
    else if (a>=55 && a<=59){
    printf("The GPA is: C");}
    else if (a>=50 && a<=54){
    printf("The GPA is: C-");}
    else if (a>=40 && a<=49){
    printf("The GPA is: D");
    }
```

```
else
{ printf("Fail);
return 0;}
```

Q.Draw a flow chart to pick the largest of three numbers? Exam-2011,ICE,CSE



Q.Draw a flow chart for computing the roots of a quadriatic equation? Ans:



Q.Write a C program to Find Roots of Quadratic Equation , Exam:ACCE-2013

```
#include<math.h>/* This is needed to use sqrt() function.*/
int main()
{
float a, b, c, determinant, r1,r2, real, imag;
  printf("Enter coefficients a, b and c: ");
  scanf("%f%f%f",&a,&b,&c);
  determinant=b*b-4*a*c;
if(determinant>0)
{
    r1=(-b+sqrt(determinant))/(2*a);
    r2=(-b-sqrt(determinant))/(2*a);
    printf("Roots are: %.2f and %.2f",r1 , r2);
}
elseif(determinant==0)
{
   r1 = r2 =-b/(2*a);
   printf("Roots are: %.2f and %.2f", r1, r2);
}
else
{
   real=-b/(2*a);
   imag = sqrt(-determinant)/(2*a);
   printf("Roots are: %.2f+%.2fi and %.2f-%.2fi", real,imag, real, imag);
}
return0;
}
```

Output 1

Q.What would be the output of the following programs? Exam:ACCE-2013
Ans:

```
#include<stdio.h>
int main()
 {
  int x=4,y,z;
  y=--x;
  z=x--;
  printf("\n%d%d%d",x,y,z);
  return 0;
 }
```

Ans: 2,3,3

```
#include<stdio.h>
int main()
 {
   int j;
  while(j<=10)
   {
      printf("\n%d",j);
      j=j+1;
   }
  return 0;
 }
```

Ans: There is no value of j;because j is not initialized.

Q. Consider the following program Exam:ACCE-2013
Ans:

```
#include<stdio.h>
int main()
{
int i=10,n=0;
while(i<1)
{
if(i&1==1)
{
i+i<<2;
i=i|1;
}
else i>>=1;
n++;

}
printf("%d",n);
    return 0;
 }
```
Ans: 0

Q.**What do you mean by "Nesting of if else statement s" Exam;ICE-2013**
Ans:
when a seres of decisions are involved, we may have use more than one if-else statement in nested form as shown below:

```
  If(test-condition_1)
   {
 If(test_condition_2)
{statement_1
}
Else
{
 Statement_2
}
     }
    Else
      {
         Statement_3
       }
```

If the condition_` is false the statement_3 will be execute. Otherwise it continues to perform the second test . it the condition_2 is true the statement_1 well be evaluated and then the control is transferred to the statement_x
Let a program to explain it which plays to contain same number or none.

```
#include<stdio.h>
Int main()
        {
    Int a,b,c,max;
Printf("a=");
Scanf("%d",&a);
Printf("b=");
Scanf("%d",&b);
Printf("c=");
Scanf("%d",&c);
If(a>b)
 {
   If(a>c)
     Max=a;
Else
  Max=b;
}
Else
  {
If(b>c)
   Max=b;
  Else
Max=c
}
Printf("Maximum =%d",max);
   }
```

Q.Compare different loop statements used in C programming with flowchart and example?Exam:APEE,ACCE-2013
Ans: for loop, while loop, do while loop ,switch and goto statement briefly describe.

Q.**Write the syntax of switch statement**?
Ans:
The syntax of switch statement is as shown below :

```
switch(expression)
  {
case value_1:
block-1;
break;
  case value_2:
     block-2;
     break;
………………………………………………………………………………………………
………………………………………………………………………………………………
default:
default-block;
break:
}
statement-x;
  }
```

Q**. Write the advantage and Disadvantage of if statement?**
Ans:
Advantage of IF:
It can be used whenever you want a condition to get satisfied and then execute a particular set of codes.
Disadvantage of IF:
In IF we wont mention what to do if the condition does not get satisfied,to overcome this we use IF ELSE.
Q**. write the advantage and Disadvantage of Loop?**
Ans:
Advantage of loop:
It can be used whenever a single or a set of statements have to be repeated again and again. It reduces the typing effort.
Disadvantage of loops:
If the condition is not properly specified then there is a chance that the program may go into infinite loop.

2.5 LOOP

Important Question:

1. Define the structure of a 'for' loop. **Exam-2012**
2. How many times will "Bangladesh" be printed on screen?

```
(i)for(i=0;i<=6;i++)printf("Bangladesh\n");
(ii)for(i=0;i<6;i++)printf("Bangladesh\n");
(iii)for(i=2;i<=9;i++)printf("Bangladesh\n");
(iv)for(i=0;i<=9;i--)printf("Bangladesh\n");
```

   Exam-2015

3. Explain the difference of while and do-while loop with example. **Exam-2014,2016**
4. Write a C program to generate the number of the following sequence: **Exam-2012**
-50, 48, -46, 44, -42, ,0.
5. Write the above program using **'while' and 'for' loop. Exam-2012**
6. How do do-while and while statements differ? When is a for, do-while or while control statements preferable to use? **Exam-2013**
7. Write a loop that will calculate the sum of every third integer, beginning with i=2(i.e 2+5+8+,,,,,) for all values of I that are less than 100. Write the loop in two different ways.
   a. Using a do .....while statement (ii)Using a for statement.**Exam-2013**
8. What will be output of the following code?**Exam-2015**

```
Inti,j;
Main() {
For(i=1;i<=7;j++)
{
For(j=2;j<=7;j++)
{if(i==j-1)printf("A"); else printf("0");}
Printf("\n");
}
}
```

13. Consider the following c program. **Exam-2014**

```
#include <stdio.h>
 int main()
 {
        int i =10, n=0;
        while(i>1)
          {
                if(i&1==1)
                  {
                        i+=i<<2;
                        i=i|1;
                   }
                  else i>>=1;
                   n++;
        }
          printf("%d %d",n,i);
           return o;
}
```
Complete the output of the above program.

9. What is the purpose of break statement? suppose you are given an integer type array containing n elements. Write a C code to find the position of first occurrence of negative number in the array. It is required to stop the searching when the first negative integer is found. **Exam-2014**
10. What are the uses of 'break' statements? Give an example. **Exam-2015**
11. What is the difference between break and continue statement ?**Exam-21016**
12. Write a fragment of program that makes use of the goto statement .**Exam-21016**

Q. What do you mean loop ? APPE-2012
Ans:
Loop: Aloop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages.
Or

Loops cause program to execute the certain block of code repeatedly until test condition is false. Loops are used in performing repetitive task in programming. Consider these scenarios:

- You want to execute some code 100 times.
- You want to execute some code certain number of times depending upon input from user.

These types of task can be solved in programming using loo

C programming language provides the following types of loops to handle looping requirements.

1.forloop statements
2.while loop
3. Do while loop
4.goto statement

Q.What are the valid places for the keyword break to appear.
Ans:
Break can appear only with in the looping control and switch statement. The purpose of the break is to bring the control out from the said blocks.
Q.What is difference between including the header file with-in angular braces <> and double quotes
Ans:
If a header file is included with in <> then the compiler searches for the particular header file only with in the built in include path. If a header file is included with in " ", then the compiler searches for the particular header file first in the current working directory, if not found then in the built in include path.

Q.What is an infinite loop? CSE-ICE-2013ACCE-2011
Ans:
A loop executing repeatedly as the loop-expression always evaluates to true such as

```
while(0 == 0) {
}
or
while(1) {
                    }
```

Q.Can variables belonging to different scope have same name? If so show an example.
Ans:
Variables belonging to different scope can have same name as in the following code snippet.

```
int var;

void f() {
  int var; }

main() {
  int var;
}
```

Q.Can a pointer access the array?

Ans:

Pointer by holding array's base address can access the array.

Q.What is recursion?

Ans:

Function calling itself is called as recursion.

Q.What is a constant?

Ans:

A value which cannot be modified is called so. Such variables are qualified with the keyword const.

Q.What is the meaning of base address of the array?

Ans:

The starting address of the array is called as the base address of the array.

Question:Define the structure of a 'for' loop. Exam-2012

Q. What do you know about for loop explain with example?

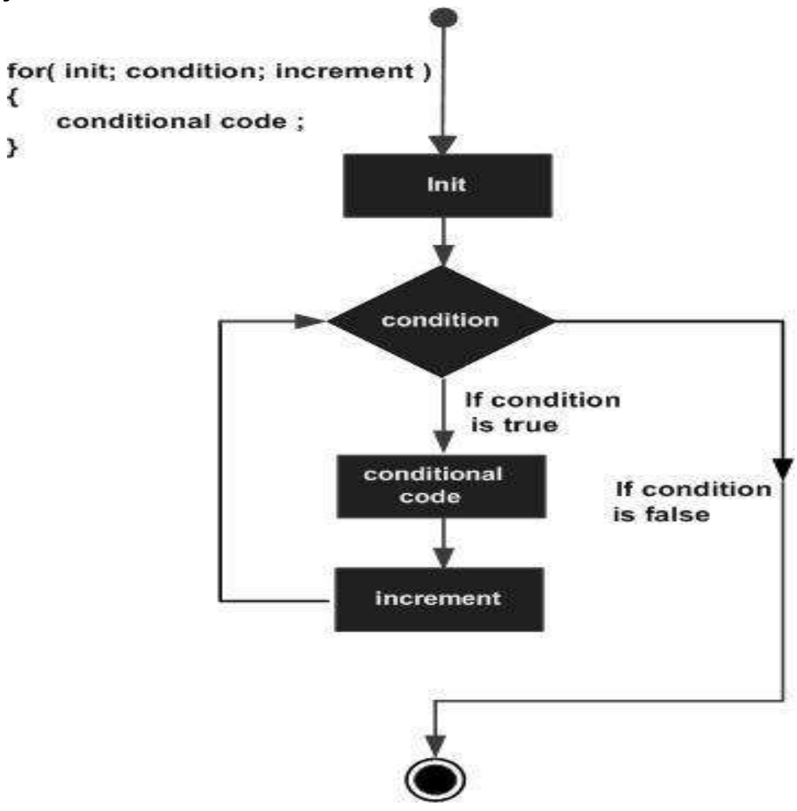
Answer:

for loop:

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times. **Syntax**

The syntax of a **for** loop in C programming language is:

```
for ( init; condition; increment )
{ statement(s);
}
```



Here is the flow of control in a for loop:

Step 1: first initialization happens and the counter variable gets initialized, here variable is I, which has been assigned by value 1.

Step 2: then condition checks happen, where variable has been tested for a given condition, if the condition results in true then C statements enclosed in loop body gets executed by compiler, otherwise control skips the loop and continue with the next statement following loop.

Step 3: After successful execution of loop's body, the counter variable is incremented or decremented, depending on the operation (++ or –).

4. The condition is now evaluated again.

Example

```
int i;
for(i=1; i<=3; i++)
{
        printf("hello, World");
}
...Output:
hello,World
hello,World
hello,World
```

or

```
#include <stdio.h>
int main ()
{
for( int a = 10; a < 20; a = a + 1 )
{
printf("value of a: %d\n", a);
}
return 0;
}
 Output:11,12,13,14…20
```

Qeustion:How many times will "Bangladesh" be printed on screen?

(i)for(i=0;i<=6;i++)printf("Bangladesh\n");
(ii)for(i=0;i<6;i++)printf("Bangladesh\n");
(iii)for(i=2;i<=9;i++)printf("Bangladesh\n");
(iv)for(i=0;i<=9;i--)printf("Bangladesh\n");

Exam-2015

Answer:

(i). 7 times
(ii).6 times
(iii).8 tiems
(iv)Infinite time

Q. **How for loop works in C programming?**

Ans:

The initialization statement is executed only once at the beginning of the for loop. Then the test expression is checked by the program. If the test expression is false, for loop is terminated. But if test expression is true then the code inside body of for loop is executed and then update expression is updated. This process repeats until test expression is false.

This flowchart describes the working of for loop in C programming

How for loop work example below:

Q.Write a program to find the sum of first *n* natural numbers where n is entered by user. Note: 1,2,3... are called natural numbers.

Ans:

```
#include <stdio.h>
                          int main(){
```

```
    int n, count, sum=0;
    printf("Enter the value of n.\n");
    scanf("%d",&n);
    for(count=1;count<=n;++count)  //for loop terminates if count>n
    {
       sum+=count;    /* this statement is equivalent to sum=sum+count
*/
    }
    printf("Sum=%d",sum);
    return 0;
}
```

Output

```
Enter the value of n.
19
Sum=190
```

Explain: In this program, the user is asked to enter the value of *n*. Suppose you entered 19 then, count is initialized to 1 at first. Then, the test expression in the for loop,i.e., (count<= n) becomes true. So, the code in the body of for loop is executed which makes *sum* to 1. Then, the expression ++count is executed and again the test expression is checked, which becomes true. Again, the body of for loop is executed which makes *sum* to 3 and this process continues. When count is 20, the test condition becomes false and the for loop is terminated.

Note: Initial, test and update expressions are separated by semicolon(;).

Q.How can ‘for’ be infinite? Exam-2011

```
For(i=0;i!=1;i--)
{
//statement
}
```

Q.Explain the syntax for for loop.

Ans:

```
for(exp-1;exp-2;exp-3) {
  //set of statements
}
```

When control reaches for exp-1 is executed first. Then following exp-2, and if exp-2 evaluates to non-zero ‘set of statements’ and exp-3 is executed, follows exp-2.

Q Write a program for factorial of a number?

Ans:

```
#include<stdio.h>
int main(){
 int i,f=1,num;
 printf("Enter a number: ");
 scanf("%d",&num);
 for(i=1;i<=num;i++)
    f=f*i;
 printf("Factorial of %d is: %d",num,f);
 return 0;
}
Sample output:
Enter a number: 5
Factorial of 5 is: 120
```

Q.Write a c program to find out the sum of series 1 + 2 + ... + n.
Ans:
Sum of 1 + 2 + ... + n series in c programming language

```
#include<stdio.h>
int main(){
   int n,i;
   int sum=0;
   printf("Enter the n i.e. max values of series: ");
   scanf("%d",&n);
   sum = (n * (n + 1)) / 2;
   printf("Sum of the series: ");
   for(i =1;i <= n;i++){
          sum=sum+i;  }
          printf("%d,",sum);
   return 0;
}
```

```
Sample output:
Enter the n: 5
Sum of the series: 1 + 2 + 3 + 4 + 5 = 15
```

Q.Code for PROGRAM TO PRINT THE SUM OF SERIES 1 + 1/2 + 1/3 + 1/4 + ... + 1/N in C Programming
Ans;

```
#include<stdio.h>
   #include<conio.h>
   void main()   {
      float n,sum=0,i;
      printf("\n Please Give The Value of N:  ");
      scanf("%f",&n);
      for(i=1;i<=n;i++)
      {
         sum = sum + (1/i);
         if(i==1)
            printf("\n 1 +");
         elseif(i==n)
            printf(" (1/%d)  ",i);
         else
            printf(" (1/%d) + ",i);
      }
      printf("\n\n THE SUM OF THIS SERIES IS %f",sum);
return 0;
   }
```

Q. **Write a program to generate the Fibonacci series ?**
Ans:

```
include<stdio.h>
int main(){
   int k,r;
int i=0,j=1,f;

    printf("Enter the number range:");
   scanf("%d",&r);
   printf("FIBONACCI SERIES: ");
   printf("%d %d",i,j); //printing firts two values.
   for(k=2;k<r;k++){
      f=i+j;
      i=j;
      j=f;
      printf(" %d",j);
   }
   return 0;
}
```

Sample output:

```
Enter the number range: 15
FIBONACCI SERIES: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Q.Find the following program.

Ans:

```
#include<stdio.h>
int main()
 {
 int i=15,j=10;
 printf("hello");
 if(i>15)j++;
 else{
   if(i<15){j--;}
   else{
     if(j<=10){
       i+=++j;
        j=i++;
      }
      else
      {
       j++;
      }
     }
    if(i>15)
     {
      i-=--j;
         }
     else{i+=j;
     }
   }
   printf("i=%d\nj=%d\n",++i,j++);
  return 0;
 }
```

Ans:i=3;j=25;

Q. Describe while statement?

Ans:

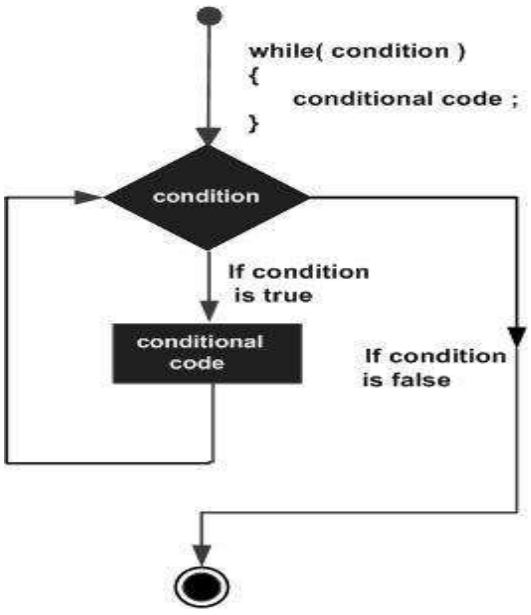
while loop:

A **while** loop statement in C programming language repeatedly executes a target statement as long as a given condition is **true.**

Syntax

The syntax of a **while** loop in C programming language is:

```
while(condition)
{
statement(s);
}
```



Here, statement(s) may be a single statement or a block of statements. The **condition** may be any expression, and **true** is any **nonzero** value. The loop iterates while the condition is **true.**

When the condition becomes **false**, program control passes to the line immediately following the loop.

Example

Q.Write a program that print 10 to 20 increment 1,using while loop.

Ans:

```
#include <stdio.h>
int main ()
{
/* local variable definition */
int a = 10;
/* while loop execution */
while( a < 20 )
{
printf("value of a: %d\n", a);
a++;
}
return 0;
Output:11,12,13,14...20
```

Q write a program for factorial of a number?

Ans:

```
#include<stdio.h>
int main(){
  int i=1,f=1,num;
  printf("Enter a number: ");
  scanf("%d",&num);
  while(i<=num){
     f=f*i;
     i++;
  }
  printf("Factorial of %d is: %d",num,f);
  return 0;
}
```

Q. Describe do while statement?

Ans:

do...while

Unlike for and while loops, which test the loop condition at the top of the loop, the **do...while** loop in C programming language checks its condition at the bottom of the loop.
A **do...while** loop is similar to a while loop, except that a **do...while** loop is guaranteed to execute at least one time.

Syntax

The syntax of a **do...while** loop in C programming language is:

```
do
{
statement(s);
}while( condition );
```

Flow Diagram

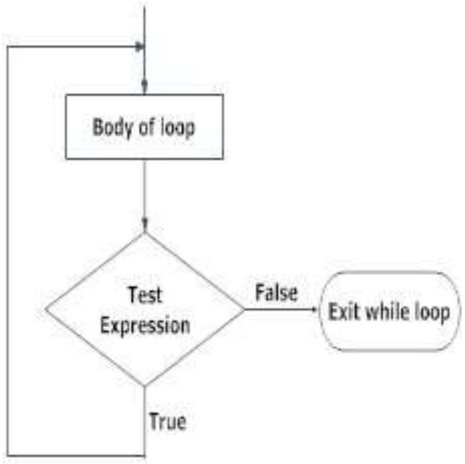


Figure: Flowchart of do...while loop

Example:

```
main()
{
int i=0
do
{
     printf("while vs do-while\n");
}while(i==1);
   printf("Out of loop");
}
```

Output:

```
while vs do-while
Out of loop
```

Q.Write a program that print 10 to 20 increment 1,using do while loop

Ans:

```
#include <stdio.h>
int main ()
{
                        //local variable definition
int a = 10;
                        // do loop execution
do
{
printf("value of a: %d\n", a);
a = a + 1;
}while( a < 20 );
return 0;
}
```

Output:10,11,12,13....20

Q.Discuss about the nested loops (nested for loop,nested while loop and nested do-while loop):

Ans:

C programming language allows to use one loop inside another loop. Following section shows few examples to illustrate the concept.

Syntax

The syntax for a nested for loop statement in C is as follows:

```
for ( init; condition; increment )
```

```
{
for ( init; condition; increment )
{
statement(s);
}
statement(s);
}
```

<u>The syntax for a **nested** while loop</u>:

```
while(condition)
{
while(condition)
{
statement(s);
}
statement(s);
}
```

<u>The syntax for a **nested do...while** loop</u>:

```
do
{
statement(s);
do
{
statement(s);
}while( condition );
}while( condition ); TUTORIA
```

Question:Write a C program to generate the number of the following sequence: Exam-2012

-50, 48, -46, 44, -42, ,0.

Write the above program using ‘while’ and ‘for’ loop. Exam-2012

Answer:

<u>A.Using While Loope:</u>

```
#include<stdio.h>
    #include<conio.h>
   main()
   {
       int i;
 i=-50;
 while(i<=0)
  {
   if(i%4==0)
     {
      printf("%d ",-1*i);
     }
     else
     {  printf("%d ",i);
     }
   i=i+2;
   }
      getch();
   }
```

B.Using Do While Loop:

```
                         #include<stdio.h>
                                  #include<conio.h>
                                 main()
                                 {
                                      int i;
                              i=-50;
                             do{
                                 if(i%4==0)
                                    {
                                     printf("%d ",-1*i);
                                    }
                                    else
                                    {  printf("%d ",i);
                                    }
                                 i=i+2;
                                 } while(i<=0);
                                     getch();
                                 }
```

Question:Write a loop that will calculate the sum of every third integer, beginning with i=2(i.e 2+5+8+,,,,,) for all values of I that are less than 100. Write the loop in two different ways.

(i)Using a do .....while statement (ii)Using a for statement.Exam-2013

Question:What will be output of the following code? Exam-2015

```
Inti,j;
Main() {
For(i=1;i<=7;i++)
{
For(j=2;j<=7;j++)
{if(i==j-1)printf("A"); else printf("0");}
Printf("\n");
}
}
```

Answer:

```
 A00000
 0A0000
00A000
000A00
0000A0
00000A
000000
```

Question: Consider the following c program. Exam-2014

```
#include <stdio.h>
 int main()
 {
      int i =10, n=0;
      while(i>1)
{
        if(i&1==1)
          {
               i+=i<<2;
               i=i|1;
           }
            else i>>=1;
             n++;
        }
          printf("%d %d",n,i);
           return o;
}
```

Complete the output of the above program.

:Answer

n=16;

i=-2030932031

Question:Explain the difference of while and do-while loop with example. Exam-2014,2016
Q.Difference between While and do-while loop? CSE-2011,ICE,APPE,ACEE-2012
Ans:

While loop	**Do While loop**
In While loop the condition is tested first and then the statements are executed if the condition turns out to be true. In do while the statements are executed for the first time and then the conditions are tested, if the condition turns out to be true then the statements are executed again.	A do while is used for a block of code that must be executed at least once. These situations tend to be relatively rare, thus the simple while is more commonly used.
while loop do not run in case the condition given is false.	A do while loop runs at least once even though the the condition given is false
In a while loop the condition is first tested and if it returns true then it goes in the loop	In a do-while loop the condition is tested at the last.
While loop is entry control loop where as do while is exit control loop.	

while loop :
while (condition)
{
Statements;
}
	do while loop :
	Do
	{
	Statements;
	}while(condition);

while(choice !=0){
System.out.println("Inside the WHILE loop.");
}
	do{
	System.out.println("Inside the DO-WHILE loop.");
	}while(choice !=0);

Or
Question:How do do-while and while statements differ? When is a for, do-while or while control statements preferable to use? Exam-2013

Question:What are the uses of 'break' statements? Give an example. Exam-2015
Question:What is the difference between break and continue statement ? Exam-21016
Question: Write a fragment of program that makes use of the goto statement . Exam-21016

Q.Discuss the Break and Continue Statement with example. Exam:ACCE-2013,11CSE,ICE.
Ans;

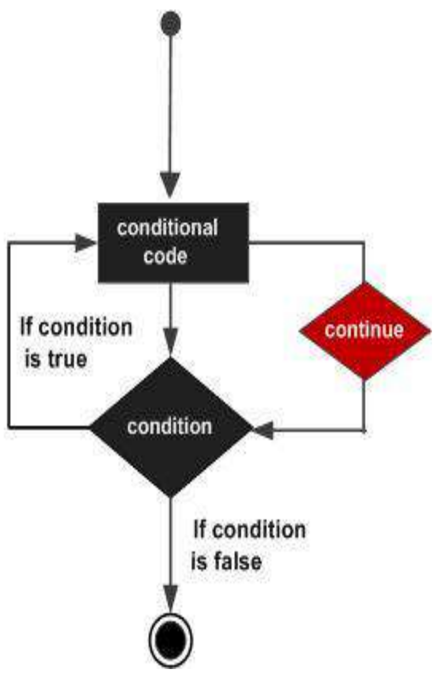
break	continue
A break can appear in both switch and 3 loop (for, while, do) statements.	A continue can appear only in 3 loop (for, while, do) statements.
A break causes the switch or loop statements to terminate the moment it is executed. Loop or switch ends abruptly when break is encountered.	A continue doesn't terminate the loop, it causes the loop to go to the next iteration. All iterations of the loop are executed even if continue is encountered. The continue statement is used to skip statements in the loop that appear after the continue.
When a break statement is encountered, it terminates the block and gets the control out of the switch or loop.	When a continue statement is encountered, it gets the control to the next iteration of the loop.
A break causes the innermost enclosing loop or switch to be exited immediately.	A continue inside a loop nested within a switch causes the next loop iteration

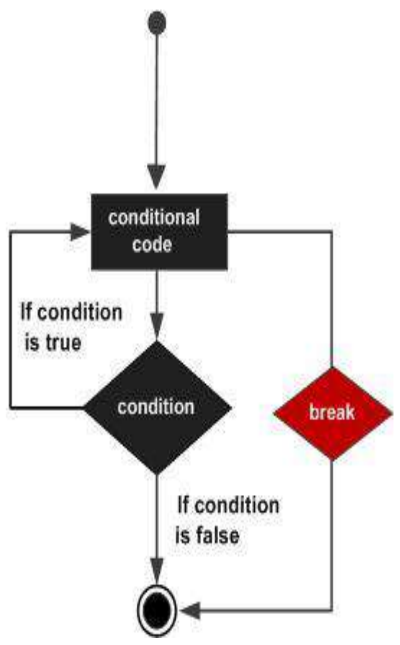
Example break statement using break using while loop and for loop:

```
int counter=10;
while(counter >=0)
{
if(counter==7)
{
      counter--;
break;
}
  printf("%d  ", counter);
  counter--;
}
Output: 1098Out of while-loop

for(int j=0; j<=8; j++)
{
if(j==4)
{
Break;
}

   printf("%d ", j);
}
..
Output:
0123 Out of for loop
```

Example continue statement using break using while loop and for loop:

```
int counter=10;
while(counter >=0)
{
if(counter==7)
{
      counter--;
continue;
}
  printf("%d  ", counter);
  counter--;
}
Output: 10986543210

for(int j=0; j<=8; j++)
{
if(j==4)
{
continue;
}

   printf("%d ", j);
}
..Output:
01235678
```



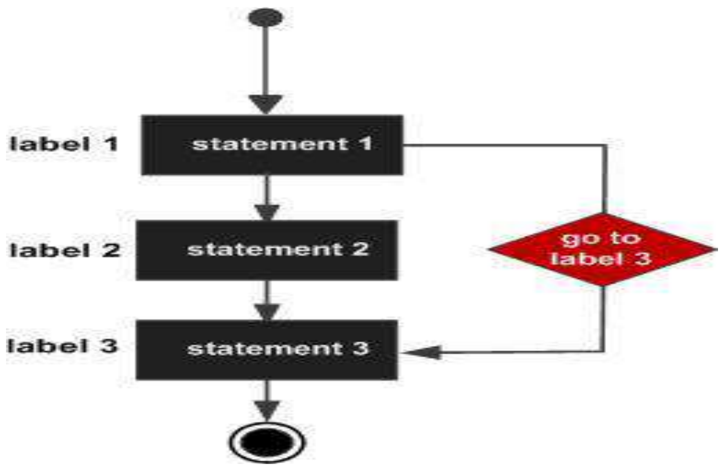


Q.What is Go to statement?
Ans:
A **goto** statement in C programming language provides an unconditional jump from the **goto** to a labeled statement in the same function.

Syntax
The syntax for a **goto** statement in C is as follows:

goto label; .
label: statement;
Here **label** can be any plain text except C keyword and it can be set anywhere in the C program above or below to **goto** statement.

Flow Diagram:



Q.Write a C program that will draw the following pyramid with height n on the console.[the example given bellow is a triangle with height n=4] Marks:4 Exam-ACCE-2014
Ans:

```
      *
    * * *
   * * * * *
 * * * * * * *
```

```
#include<stdio.h>
int main()
 {
  int i,j,n;
printf("Enter How many line?");
scanf("%d",&n);
for(i=1;i<=n;i++)
 {
   for(j=1;j<=n-i;j++)
    {
     printf("  ");
    }
   for(j=1;j<=i*2-1;j++)
     {
      printf("*");
     }
   printf("/n");

}
return 0;
}
```

Q.Consider the following program Marks:2.75 Exam-ACCE-2014

```
#include<stdio.h>
int main()
 {
int i;
 for(i=5;i!=3;i++)
  printf("%d\n",i%=8);
return 0;
```

```
                    }
```
computer the outpute of above program.
Ans:5,6,7,0,1,2;

Q.Find errors, if any in the following program: Exam: ACCE 2012

```
Include<math.h>
FLOAT X:
X=2.5
Y=exp(x);
Print(x,y);
```
Ans:there are many error in this program, we know c programming is case sensitive
1.First error Include I capital letter and also # is absence.
2.FLOAT is capital letter.
Y=initialize element is not constant;

Q.**What will be output of following c code? Exam: ACCE 2012**

```
#include<stdio.h>
int main(){
 static int i;
 for(++i;++i;++i) {
 printf("%d ",i);
 if(i==4) break;
 }
 return 0;
}
```

(A) 4
(B) 24
(C) 25
(D) Infinite loop
(E) Compilation error
Answer: (b)

Explanation:
Default value of static int variable in c is zero. So, initial value of variable i = 0
First iteration:
For loop starts value: ++i i.e. i = 0 + 1 = 1
For loop condition: ++i i.e. i = 1 + 1 = 2 i.e. loop condition is true. Hence printf statement will print 2
Loop incrimination: ++I i.e. i = 2 + 1 =3

Second iteration:
For loop condition: ++i i.e. i = 3 + 1 = 4 i.e. loop condition is true. Hence printf statement will print 4.
Since is equal to four so if condition is also true.

But due to break keyword program control will come out of the for loop.

Q.Change the following for loop to while loop Exam:ACCE-2012

```
for(m=1;m<10;m=m+1)
  {
    printf("%d",m);
  }
```

Ans:

```
m=1;
while(m<10)
    {
 printf("%d",m);
m++;
}
```

Q. Write a program that will take integer number as input and and print negative number is entered?

Write the above using 'do-while', 'while', and 'for loop'. Exam:ACCE-2012,CSE

Ans:

For loop:

```
int main()
   {

int i,n,a[12];
scanf("%d",&n)
for(i=0;i<n;i++)
{
Sscanf("%d",&a[i])
}
for(i=0;i<n;i++)
{
if(a[i>=0])
{printf("%d",a[i]);}
}
return 0;
}
```

While loop:

```
int main()
   {
```

```
int i,n,a[12];
scanf("%d",&n);
i=0;
while(i<n)
{
scanf("%d",&a[i])
}
i=0;
while(i<n)
{
if(a[i>=0])
{printf("%d",a[i]);}
i++;
}
return 0;
}
```

Do while loop:

```
int main()
    {

int i,n,a[12];
scanf("%d",&n);
do{
i=0;
scanf("%d",&a[i])
} while(i<n)

do{
i=0;
if(a[i>=0])
{printf("%d",a[i]);}
i++;
} while(i<n)


return 0;
}
```

GCD and LCM:

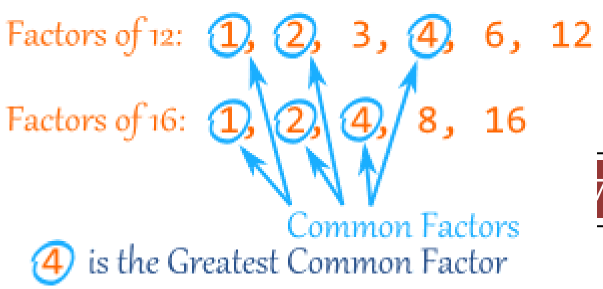
Q. Discuss about Greatest common devisor with example?

Ans:

It is simply the **largest** of the common factors.

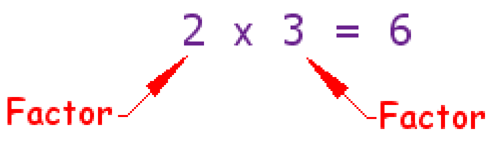
Greatest Common Factor of 12 and 16

1. Find all the **Factors** of each number,
2. Circle the **Common** factors,
3. Choose the **Greatest** of those


Factors of 12: 1, 2, 3, 4, 6, 12
Factors of 16: 1, 2, 4, 8, 16
Common Factors
4 is the Greatest Common Factor

Factor:

Factors are the numbers we multiply together to get another number:



A number can have many factors:
Factors of 12 are **1, 2, 3, 4, 6** and **12** ...

... because **2** × **6** = 12, or **4** × **3** = 12, or **1** × **12** = 12.

. We can:

- find all **factors** of both numbers .
- then select the ones that are **common** to both, and
- then choose the **greatest.**

Example:

Two Numbers	Factors	Common Factors	Greatest Common Factor
9 and 12	**9**: 1,3,9 **12**: 1,2,3,4,6,12	1,3	**3**

Q.Discuss about Least common Multiple(LCM) with example?

Ans:

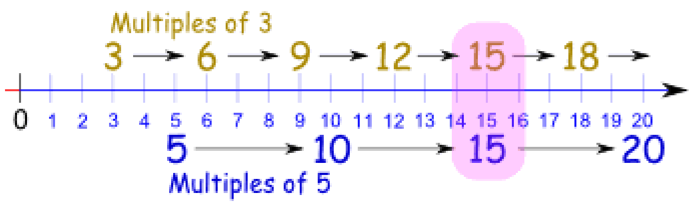
It is simply the **smallest** of the common multiples.

Least Common Multiple of 3 and 5:

List the **Multiples** of each number,

The multiples of **3** are 3, 6, 9, 12, 15, 18, ... etc

The multiples of **5** are 5, 10, 15, 20, 25, ... etc



Find the first **Common** (same) value:

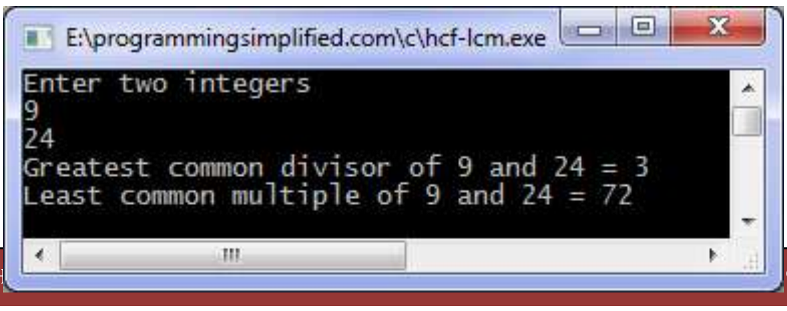
The **Least Common Multiple** of 3 and 5 is **15**

(15 is a common multiple of 3 and 5, and is the smallest, or least, common multiple)

Q.Write a program to find a greatest common divisor(GCD)and least common multiplier(LCM)?
Ans:

```
#include <stdio.h>
int main() {
  int a, b, x, y, t, gcd, lcm;
  printf("Enter two integers\n");
  scanf("%d%d", &x, &y);
  a = x;
  b = y;
  while (b != 0) {
    t = b;
    b = a % b;
    a = t;
  }
   gcd = a;
  lcm = (x*y)/gcd;

  printf("Greatest common divisor of %d and %d =
%d\n", x, y, gcd);
  printf("Least common multiple of %d and %d =
%d\n", x, y, lcm);

  return 0;
}
```

Output of program:


E:\programmingsimplified.com\c\hcf-lcm.exe

```
Enter two integers
9
24
Greatest common divisor of 9 and 24 = 3
Least common multiple of 9 and 24 = 72
```

Q.Write a program to find a least common multiplier(LCM)?
Ans:

```
#include<stdio.h>
int main()
{
int n1,n2,temp1,temp2;
   printf("Enter two positive integers: ");
   scanf("%d %d",&n1,&n2);
   temp1=n1;
   temp2=n2;
while(temp1!=temp2)
{
if(temp1>temp2)
         temp1-=temp2;
else
         temp2-=temp1;
}
   printf("LCM of two numbers %d and %d is %d", n1,
n2,(n1*n2)/temp1);
return0;
}
```

The output of these two programs is same.

Output

```
Enter two positive numbers: 15
9
LCM of two numbers 15 and 9 is 45
```

3.1 Basic User Datatypes:

1. What is user-defined variable? Why it is required for C programming? **Exam-2014**

Answer:

User-defined variables are variables which can be created by the user and exist in the session. This means that no one can access user-defined variables that have been set by another user, and when the session is closed these variables expire.

However, these variables can be shared between several queries and stored programs.
User-defined variables names must be preceded by a single *at* character (@). While it is safe to use a reserved word as a user-variable name, the only allowed characters are ASCII letters, digits, dollar sign ($), underscore (_) and dot (.). If other characters are used, the name can be quoted in one of the following ways:

@`var_name`
@'var_name'
@"var_name"

These characters can be escaped as usual.
User-variables names are case insensitive, though they were case sensitive in MySQL 4.1 and older versions.
User-defined variables cannot be declared. They can be read even if no value has been set yet; in that case, they are NULL. To set a value for a user-defined variable you can use:

SET statement;
:= operator within a SQL statement;
SELECT ... INTO

Necessity of Userdefined Variable:

Programming languages are next to useless without variables. These special information-holding areas can store numbers, text strings, objects, and other data types. Once stored, this information can be used later in your program. With a variable, a person's name could be stored and used at some point in the script.
Variables are temporary holders of information. They can hold:

- **Numeric values ("numbers") -- numbers that can be added together. Example: 2+2 results in 4**
- **Character strings -- a collection of text, such as "JavaScript" or "My name is Mudd"**
- **True/false values -- the Boolean true and false**

3.2 ARRAY

Important Question

1. In what way does an array differ from an ordinary variable? **Exam-2013**
2. What conditions must be satisfied by all of the elements of any given array ?**Exam-21016**
3. There is an array of integers 'a' that holds 100 integers. Write a C program that will copy the content of array 'a' to another array 'b', with the condition that all the integers stored in 'a' greater than 100 will be stord in the upper part of array 'b'. **Exam-2012**
4. Suppose, you are given an array of 'n' integers. You are asked to develop a program to sort that array in ascending order using at most one extra variable. Draw a flowchart to solve the problem. **Exam-2015**
5. What will be the output of the following code?

```
 int x[10]={1,4,3,6,8,2,9,0,5,7};
int i,j,k,tmp,big,p;
main() {
 for(i=1;i<=5;i++)
{
Big=x[i];
 For(j=I;j<=5;j++)
   {if(x[j]> big) p=j+1;}
tmp=x[p];   x[p]=x[i];   x[i]=tmp;
}
for(k=1;k<8;k++) printf("%d –th
%d\n",k,x[k]);
}
```
Exam-2015

6. List the syntax error (if any) of each line of the following code ?
   (Object of the question: To check the knowledge of basic c syntax)

```
#include<conio.h>
int  1x, 2x, y1, y2;
float z;
char a[10], b[10];
Main( )
{
 scanf("%d%d%f",y1,z);
 scanf("%c%c%c), &a[1],a[2],&a[3]);
 b[2]=a[2];
 y2=b[2]+a[1]+y1;
 printf("%f%f%f%d%d",&y1,z,y2,z,a[3]);
 }
```
Exam-21016

7. There are two matrix A and B of size 5*5. Write a C program that will add A and B and store the result in A. **Exam-2012**
8. There ia a two dimentional matrix M of size 50*50, each cell of the matrix contains

either ‘0’ or ‘1’. There is another array N of size 50. Write a C program that will sum thr total number of ‘1’ in i^th coloumn of M and store in the i^th cell of N where i=1,2,….,50. **Exam-2012**

9. Write a program to take two matrices A[n][n] an B[n][n] from the keyboard. Set the value of each cell of a row of the matrix C[n][n] with biggest value of respective two rows of matrix A and B. The maximum size of ‘n’ is 10. **Exam-2015**
10. Can initial values be specified within an external array definition? Can they be specified within a static array definition? **Exam-2013**
11. When a multidimensional array is passed to a function, how are the formal argument declarations written? Compare with one-dimensional arrays. **Exam-2013**
12. How pointers and arrays are closely related? What are the advantages of using pointer?
    What is the difference between int *p[5] and int(*p)[5]. **Exam-2013**
13. Write a program that will read a **5 ∗ 5** matrix of integer numbers and then calculate the row sum and column sum and put the result in a **2 ∗ 5**matrix where first row represent the row sum and second row the column sum. **Exam-2013**
14. What will be the output of the following program?**Exam-2016**

(Object of the problem: Check the capacity of four-layer nested loop control)

```
#include<stdio.h>
#include<conio.h>
int x[5][5]={ {1, 4, 3, 6, 8},
              {2, 9, 0, 5, 7},
              {5, 9, 6, 7, 6},
              {9, 0, 2,6, 8},
              {3, 6, 0, 1, 7} };
 int i, j, k, l, temp,big,p;
 main() {
 for(i=0;i<=4;i++)
  {
   for(j=0;j<=4;j++)
   {
     for(k=0;k<=4;k++)
      {
       for(l=0;l<=4;l++)
       {
          }
        }
     }
 }
   for(i=0;i<=4;i++)
   {
     for(j=0;j<=4;j++)
     {
```

15. Can entire arrays be processed with single instructions , without repetition ? **Exam-2016**
16. You are given an array containing some real numbers. You are asked to develop a program find the average of positive and negative numbers separately. Draw the flowchart to solve the problem. **Exam-2014**

Q.Define an Array? state the necessity of an array?
Ans;

An array is a fixed-size sequence collection of elements of the same data type. It is simply a grouping of like tupe data in its simplest form. An array can be used to represent a list of numbers.

Some examples wher the concept of an array can be used:

- List of temperatures recorded every hour in a day.
- List of employees in an organization.

Or

Array is a Group of consecutive memory locations

- ❖ Same name and type

  To refer to an element, specify
- ❖ Array name
- ❖ Position number

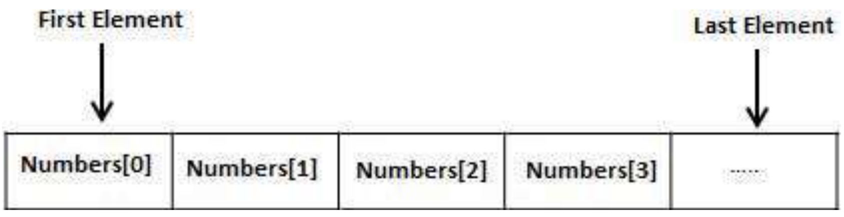
Format:

arrayname[*position number*]

Example:

- First element at position **0**
- **n** element array named **c**:
- **a[0], a[1]...a[n – 1]**

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables



Question:What the Necessity of an array or Q. what is the importance of array in c:

- Consider a scenario wherein you have to store 100 integer numbers, entered by user, in order to find out the average of them. To program this scenario you have two ways – 1) Define 100 variable of integer type and at last perform the average operation. 2) Have a single integer array to store all the values.
- Which solution is better as per you? Obviously the second solution, it is convenient to store same data types in one single variable and later access them using array indexArrays are an important structure to hold data.
- It allows us to hold many objects of the same type, and more importantly, to use a for loop to access the elements by their index.

Question: In what way does an array differ from an ordinary variable? Exam-2013
Answer:

Array	**Ordinary Variable**
Array holds multiple values	An ordinary variable hold a single value
An array can be thought as a single variable.	However, a variable can also be a structure or a union, capable of holding multiple values,
Example: int arr[]={1,2,3},String str[]={"hello","world"};	Example: int i=3,String str="hi";

Question: What conditions must be satisfied by all of the elements of any given array ? Exam-21016
Answer:
The elements of any given array have common characteristics. Elements can be characters, integers, floating-point numbers, structures, and so on. They must have to be of the same type and the same storage class

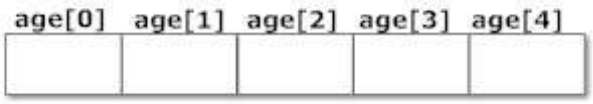
The entire elements of any given array have common characteristics. Then can be characters, integers, floating-point numbers, structures, and so on. They must all, however, be of the same type and the same storage clas

Q.How to Declaring and Initialization one dimensional Array? Exam:ACCE-2013
Ans:
Declaring one dimensional Arrays: Like any other variable, arrays must be declared before they are used.
To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows:

– Type of array
– Name
– size

The general from of array declaration is .
Type variable-name[size]

int age[5];
Where type=int
variable-name=a;
size=5;

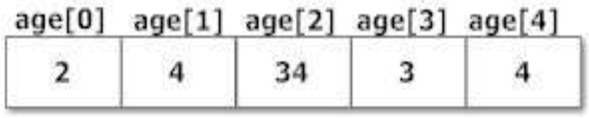


Array elements

This is called a single-dimensional array. The **arraySize** must be an integer constant greater than zero and type can be any valid C data type.
Initialization:

Arrays can be initialized at declaration time in this source code as:

```
int age[5]={2,4,34,3,4};
int age[]={2,4,34,3,4};
```



Initialization of one-dimensional array

In this case, the compiler determines the size of array by calculating the number of elements of an array.

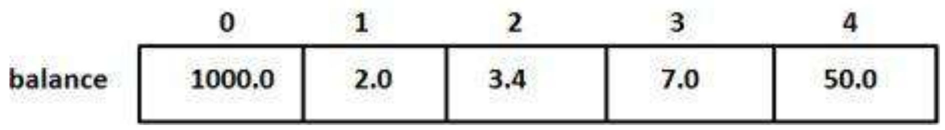
Q.How can we Accessing Array Elements? Exam:
Ans:
An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array.
Index starts with 0, which means array_name[0] would be used to access first element in an

array.
or
In C programming, arrays can be accessed and treated like variables in C.
For example:

```
scanf("%d",&age[2]);
statement to insert value in the third element of array age[].

scanf("%d",&age[i]);  i=0,1,2,3,...n
The first element of array is age[0], second is age[1], ith is age[i-1]

printf("%d",age[0]);
statement to print first element of an array.

printf("%d",age[i]);
statement to print (i+1)th element of an array.

Int a=b[4];
```

You can initialize array in C either one by one or using a single statement as follows:

❖ double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};



Now if want to access this arry or print any value from this array.
printf("%lf", balance[0]);
output:1000.0

Question: There is an array of integers 'a' that holds 100 integers. Write a C program that will copy the content of array 'a' to another array 'b', with the condition that all the integers stored in 'a' greater than 100 will be stord in the upper part of array 'b'. Exam-2012
Answer:

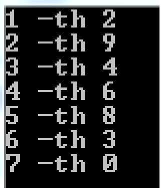
```
#include<stdio.h>
    #include<conio.h>
   main()
   {
    int a[100],b[100],i,j;
    int n;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
      {
       scanf("%d",&a[i]);
      }
      j=n+1;
    for(i=1;i<=n;i++)
      {
      if(a[i]>100)
         {
         j--;
         b[j]=a[i];
         }
         }
```

```
 for(i=n;i>=j;i--)
   {
     printf("%d ",b[i]);
   }
getch();
 }
```

Question: What will be the output of the following code? Exam-2015

```
 int x[10]={1,4,3,6,8,2,9,0,5,7};
int i,j,k,tmp,big,p;
main() {
 for(i=1;i<=5;i++)
{
Big=x[i];
 For(j=I;j<=5;j++)
   {if(x[j]> big) p=j+1;}
tmp=x[p];    x[p]=x[i];   x[i]=tmp;
}
for(k=1;k<8;k++)
printf("%d –th %d\n",k,x[k]);
}
```

Answer:



```
1 -th 2
2 -th 9
3 -th 4
4 -th 6
5 -th 8
6 -th 3
7 -th 0
```

Question: List the syntax error (if any) of each line of the following code ?
(Object of the question: To check the knowledge of basic c syntax)

```
#include<conio.h>
int  1x, 2x, y1, y2;
float z;
char a[10], b[10];
Main( )
{
 scanf("%d%d%f",y1,z);
 scanf("%c%c%c), &a[1],a[2],&a[3]);
 b[2]=a[2];
 y2=b[2]+a[1]+y1;
 printf("%f%f%f%d%d",&y1,z,y2,z,a[3]);
}
```

Error List:

1. **first character in variable must letter or under_score;**
2. **expected `)' before ';' token**

or

In line 1 : There is no 'stdio.h' header file which contains the functions in main. Such as scanf, printf.

In line 2: there is int 1x, 2x… Here numbers can not be prefix of a variable.
In line 7: in scanf function there the memory location is not declared in case of one integer input and also the sequence of integer and float is not right.
In line 8: There is no address sign before telling the memory location of a[2] in scanf function.
In line 9 and 10: b[2] is used though it has no value till then.
In line 11: printf("%f%f%d%d",&y1,z,y2,z,a[3]);

There is given that %f %f%%d%d that means first and second output will be float type but at the time of telling memory location there is &y1 where & is not defined.

Q.Write a program to find the sum marks of n students using arrays?
Ans;

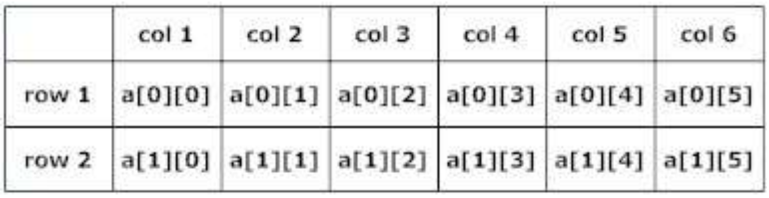
```
#include<stdio.h>
int main(){
int marks[10],i,n,sum=0;
    printf("Enter number of students: ");
    scanf("%d",&n);
for(i=0;i<n;++i){
        printf("Enter marks of student%d: ",i+1);
        scanf("%d",&marks[i]);
        sum+=marks[i];
}
    printf("Sum= %d",sum);
return0;
}
```

Q.Why array is useful in c programming?   Exam-ACCE-2011,CSE-APPE,MSE,
Q.Write Advantage of using array?
Ans:
1. An array provides singe name .So it easy to remember the name of all element of an array.
2. Array name gives base address of an array .So with the help increment operator we can visit one by one all the element of an array.
3. Array has many application data structure.

Q. How declare multidimensional array? Exam:ICE ACCE-2013
<u>**Multi-dimensional Arrays:**</u>
C programming language allows programmer to create arrays of arrays known as multidimensional arrays.
or
C programming language allows **multidimensional arrays**.
type name[size1][size2]...[sizeN];
Tow dimensional array:

    type arrayName [x][y];
For example:
    float a[2][6];

Here, *a* is an array of two dimension, which is an example of multidimensional array.



	col 1	col 2	col 3	col 4	col 5	col 6
row 1	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]	a[0][5]
row 2	a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]	a[1][5]

Figure: Multidimensional Arrays

A two-dimensional array can be think as a table which will have x number of rows and y number of columns.
A 2-dimentional array **a**, which contains three rows and four columns can be shown as below:

__Initializing Two-Dimensional Arrays:__
Multidimensional arrays may be initialized by specifying bracketed values for each row. Following is an array with 3 rows and each row has 4 columns.

```
int a[3][4] = {
{0, 1, 2, 3} , {4, 5, 6, 7}
}
Or
 int disp[2][4]={
 {10,11,12,13},
 {14,15,16,17}
 };
```

Question: There are two matrix A and B of size 5*5. Write a C program that will add A and B and store the result in A. Exam-2012
Q.Write a program that add two matrics?
Matrix addition:

```
#include<stdio.h>
int main(){
 int a[5][5],b[5][5],c[5][5],i,j;
 printf("Enter the First matrix->");
 for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&a[i][j]);
 printf("\nEnter the Second matrix->");
 for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&b[i][j]);
 printf("\nThe First matrix is\n");
 for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",a[i][j]);
 }
 printf("\nThe Second matrix is\n");
 for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
    printf("%d\t",b[i][j]);
  }
  for(i=0;i<3;i++)
     for(j=0;j<3;j++)
         c[i][j]=a[i][j]+b[i][j];
  printf("\nThe Addition of two matrix is\n");
  for(i=0;i<3;i++){
     printf("\n");
     for(j=0;j<3;j++)
         printf("%d\t",c[i][j]);
  }
  return 0;
}
```

Question: Write a program that will read a 5 $*$ 5 matrix of integer numbers and then

calculate the row sum and column sum and put the result in a $2 * 5$ matrix where first row represent the row sum and second row the column sum. Exam-2013

Question: There ia a two dimentional matrix M of size 50*50, each cell of the matrix contains either '0' or '1'. There is another array N of size 50. Write a C program that will sum thr total number of '1' in i^th coloumn of M and store in the i^th cell of N where i=1,2,....,50. Exam-2012

Question: Write a program to take two matrices A[n][n] an B[n][n] from the keyboard. Set the value of each cell of a row of the matrix C[n][n] with biggest value of respective two rows of matrix A and B. The maximum size of 'n' is 10. Exam-2015

```
#include <stdio.h>

int main()
{
      int n;
      printf("Enter your n: ");
      scanf("%d", &n);
  int A[n][n], B[n][n], C[n][n];
  int row, col;
  printf("Enter elements in matrix A of size %dx%d: \n", n, n);
  for(row=0; row<=n-1; row++)
  {
    for(col=0; col<=n-1; col++)
    {
      scanf("%d", &A[row][col]);
    }
  }
  printf("\nEnter elements in matrix B of size %dx%d: \n", n, n);
  for(row=0; row<=n-1; row++)
  {
    for(col=0; col<=n-1; col++)
    {
      scanf("%d", &B[row][col]);
    }
  }
  for(row=0; row<=n-1; row++)
  {
    for(col=0; col<=n-1; col++)
    {
      C[row][col] = A[row][col] + B[row][col];
    }
  }
  printf("\nSum of matrices C = \n");
  for(row=0; row<=n-1; row++)
  {
    for(col=0; col<=n-1; col++)
    {
      printf("%d ", C[row][col]);
    }
    printf("\n");
  }

  return 0;
}
```

Question: When a multidimensional array is passed to a function, how are the formal argument declarations written? Compare with one-dimensional arrays. Exam-2013

Q.Write a program Passing Multi-dimensional Arrays to Function?

Ans:

To pass two-dimensional array to a function as an argument, starting address of memory area reserved is passed as in one dimensional array

```
#include <stdio.h>
void Function(int c[2][2]);
int main(){
  int c[2][2],i,j;
  printf("Enter 4 numbers:\n");
  for(i=0;i<2;++i)
    for(j=0;j<2;++j){
       scanf("%d",&c[i][j]);
    }
  Function(c);
  return 0;
}
void Function(int c[2][2]){
  int i,j;
  printf("Displaying:\n");
  for(i=0;i<2;++i)
    for(j=0;j<2;++j)
       printf("%d\n",c[i][j]);
}
```

Output

```
Enter 4 numbers:
2
3
4
5
Displaying:
2
3
4
5
```

Q.How can arrays passing as a function arguments?

<u>Passing Arrays as Function Arguments:</u>

If you want to pass a single-dimension array as an argument in a function you would have to declare function formal parameter in one of following three ways

Way-1

Formal parameters as a pointer as follows. You will study what is pointer in next chapter.

```
void myFunction(int *param)
{
.
.}
```

Way-2

Formal parameters as a sized array as follows:

```
void myFunction(int param[10])
 { .
}
```

Way-3
Formal parameters as an unsized array as follows:

```
void myFunction(int param[])
{
. .
}
```

Question: Can initial values be specified within an external array definition? Can they be specified within a static array definition? Exam-2013
Q. what do you mean by static array ?
Ans:
Static arrays are allocated memory at compile time and the memory is allocated on the stack. Whereas, the dynamic **arrays** are allocated memory at the runtime and the memory is allocated from heap.
Example:
int arr[] = { 1, 3, 4 }; // **static** integer **array**

Question: What will be the output of the following program? Exam-2016
(Object of the problem: Check the capacity of four-layer nested loop control)

```
        #include<stdio.h>
        #include<conio.h>
int x[5][5]={ {1, 4, 3, 6, 8},
              {2, 9, 0, 5, 7},
              {5, 9, 6, 7, 6},
              {9, 0, 2,6, 8},
              {3, 6, 0, 1, 7} };
int i, j, k, l, temp,big,p;
main() {
for(i=0;i<=4;i++)
    {
for(j=0;j<=4;j++)
     {
for(k=0;k<=4;k++)
       {
for(l=0;l<=4;l++)
        {
          }
        }
      }
    }
for(i=0;i<=4;i++)
     {
for(j=0;j<=4;j++)
       {
printf ("%d",x[i][j]);
        }
```

Answer;
{ {26, 29, 28, 31, 33},

```
                                  {27, 34, 25, 30, 32},
                                   {30, 34, 31, 32, 31},
                                      {34, 25, 27,31, 33},
                                      {28, 31, 25, 26, 32} };
```

Q.Write a program Find out average of 20 integers values?

Ans:

```
#include<stdio.h>
int main()
{
int avg =0;
int sum =0;
int x=0;

/* Array- declaration – length 20*/
int num[20];

/* for loop for receiving inputs from user and storing it in array*/
for(x=0; x<=19;x++)
{
      printf("enter the integer number %d\n", x);
      scanf("%d",&num[x]);
}
for(x=0; x<=19;x++)
{
      sum = sum+num[x];
}

   avg = sum/20;
   printf("%d", avg);
return0;
}
```

.Q. write a program that will read the values of matrices a and b and multiply the above two materics to produce the matrix c. Marks:3.50 Exam-ACCE-2013

```
include<stdio.h>
int main(){
 int a[5][5],b[5][5],c[5][5],i,j,k,sum=0,m,n,o,p;
 printf("\nEnter the row and column of first matrix");
 scanf("%d %d",&m,&n);
 printf("\nEnter the row and column of second matrix");
 scanf("%d %d",&o,&p);
 if(n!=o){
    printf("Matrix mutiplication is not possible");
    printf("\nColumn of first matrix must be same as row of second matrix");
 }
 else{
    printf("\nEnter the First matrix->");
    for(i=0;i<m;i++)
    for(j=0;j<n;j++)
```

```
            scanf("%d",&a[i][j]);
         printf("\nEnter the Second matrix->");
         for(i=0;i<o;i++)
         for(j=0;j<p;j++)
            scanf("%d",&b[i][j]);
         printf("\nThe First matrix is\n");
         for(i=0;i<m;i++){
         printf("\n");
         for(j=0;j<n;j++){
            printf("%d\t",a[i][j]);
         }
         }
         printf("\nThe Second matrix is\n");
         for(i=0;i<o;i++){
         printf("\n");
         for(j=0;j<p;j++){
            printf("%d\t",b[i][j]);
         }
         }
         for(i=0;i<m;i++)
         for(j=0;j<p;j++)
            c[i][j]=0;
         for(i=0;i<m;i++){ //row of first matrix
         for(j=0;j<p;j++){  //column of second matrix
            sum=0;
            for(k=0;k<n;k++)
              sum=sum+a[i][k]*b[k][j];
            c[i][j]=sum;
         }
         }
      }
      printf("\nThe multiplication of two matrix is\n");
      for(i=0;i<m;i++){
         printf("\n");
         for(j=0;j<p;j++){
            printf("%d\t",c[i][j]);
         }
      }
      return 0;
    }
```

Q. Define a one-dimensional five element floating point array named "class " and initialized the array to zero.
Ans:
Int class[]={0,0,0,0,0};

Q. write down the meaning of the following arrays:

(i) Float stack[10];
(ii) Int list[6][4]
Ans:
(i) Float type one-dimensional array with 10 elements.

(ii) Integer type two –dimensional array with 24 elements.

Question: Suppose, you are given an array of 'n' integers. You are asked to develop a program to sort that array in ascending order using at most one extra variable. Draw a flowchart to solve the problem. Exam-2015

Q. write a program to read numbers in a one-dimensional array and then sort the numbers in ascending order? CSE-2011,ICE-2013,APPE

Ans:

```
#include<stdio.h>
int main()
{
int a[30],i,j,t;
printf("enter 30 numbersn");
for(i=0;i<30;i++)
scanf("%d",&a[i]);

for(i=0;i<30;i++)
{
for(j=i+1;j<30;j++)
{
if(a[j]>a[j])
{
t=a[j];
a[j]=a[j+1];
a[j+1]=t;
}}}
return 0;
}
```

Q.Write a program to read numbers in a one-dimensional array and then sort the numbers in descending order?

Ans:

```
#include <stdio.h>

voidmain ()
{
 inti,j,a,n,number[30];

 printf("Enter the value of N\n");
 scanf("%d", &n);

 printf("Enter the numbers \n");
 for(i=0; i<n; ++i)
  scanf("%d",&number[i]);

 for(i=0; i<n; ++i)
 {
  for(j=i+1; j<n; ++j)
  {
   if(number[i] < number[j])
   {
    a= number[i];
    number[i] = number[j];
    number[j] = a;
   }
  }
 }
 printf("The numbers arrenged in descending order are given below\n");
 for(i=0; i<n; ++i)
  printf("%d\n",number[i]);}                    /* End of main() *
```

Q. Write a program that accept the marks of 100 students from the user ad then shows the highest ,lowest and average marks.Exam:ACCE-CSE,ICE,APPE
Ans;

```
#include<stdio.h>
int main() {
  int a[30], i, num, largest,min,sum,average;
  printf("\nEnter no of elements :");
  scanf("%d", &num);
  //Read n elements in an array
  for (i = 0; i < num; i++)
    scanf("%d", &a[i]);
  //Consider first element as largest
  largest = a[0];
  for (i = 0; i < num; i++) {
    if (a[i] > largest) {
      largest = a[i];
    }
  }
 //Consider first element as minimum
min = a[0];
  for (i = 0; i < num; i++) {
    if (a[i] < min) {
      min = a[i];
    }
  }
//Consider Average
for (i = 0; i < num; i++) {
 sum=sum+I;
} average=sum/num;

  // Print out the Result
  printf("\nLargest Element : %d", largest);
  printf("\nMinimum Element : %d", min);
printf("\nAverage Mark : %d", average);
  return (0);
       }
```

.Q We want to declare a two-dimensional inter type array called matrix for 3 rows and 5 columns which of the following declarations are correct? Exam:ACCE-2012

(i)int matrix[3],[5];
Ans: its wrong , is not allowed here.
(ii)int matrix[5],[3];
Ans: its wrong , is not allowed here.
(iii)int matrix[1+2][2+3];

Ans: its valid
(iv)int matrix[3,5];
Ans: its not valid
(v)int matrix[5][5];
Ans: its valid

Q.How Passing array to function ?
Ans:

Array can be passed as an argument to a function:
In this method of calling a function, the actual arguments gets copied into formal arguments.
Or function calling by base address and length of array.those are arr[] and n

In this example passing a single element of an array to function

```
#include <stdio.h>
void display(int a)
  {
  printf("%d",a);
  }
int main(){
  int c[3]={2,3,4};
  display(c[2]);    //Passing array element c[2] only.
  return 0;
}
```
Output 4

Q. **Write a C program to pass an array containing age of person to a function. This function should find average age and display the average age in main function.**

```
#include <stdio.h>
float average(int a[],int n);
int main(){
    float avg;
   int n, c[]={23, 55, 22, 3, 40, 18};
   n=6;
    avg=average(c,n);     /* Only name of array is passed as argument. */
    printf("Average age=%.2f",avg);
    return 0;
  }
float average(int a[],int n){
    int i;
    float avg, sum=0.0;
    for(i=0;i<n;++i){
      sum+=a[i];
    }
    avg =(sum/6);
    return avg;
}
```
Question: Can entire arrays be processed with single instructions , without repetition ? Exam-2016
Answer

; The entire arrays cannot be processed with single instruction without repitation.the entire array can pass by reference only.

Part-B

Marks-26.5

3.3 POINTERS

Importent Question

1. What is a pointer? Is there any relation between a pointer and the name of a one-dimensional array? **Exam-2015**
2. What is the relation between pointer and single-dimension array? give example. **Exam-2014**
3. Explain 'array name is a pointer'. **Exam-2012**
4. What are the difference between malloc() and calloc()?**Exam-2013**
5. Explain the output of the following C code: **Exam-2014**

```
main() {
    int *a, b = 30;
     a = &b;
     b = *a + 40;
     a = b /5;
printf("%d %d ", *a , b);   }
```

6. Define dynamic memory allocation. Why it is required? **Exam-2014**
7. Explain the following C declarations. i) int *p[10]; ii) int (*p)[10];
   iii) int *p(char *a[ ]);**Exam-2014**
8. What will be the output of following code ?**Exam-21016**
   (Objective of the question : Check the concept of pointers)

```
    void func1(int*p, int*q ,int*r, int*s);

main(){
inta,b,c,d,*x,*y;
a=15;b=100,x=&d;y=&c;c=25;d=300;
printf("Before calling %d %d %d %d \n",a,b,*x,*y);
func1(&c,&d,&a,&b);
printf("After calling %d %d %d %d \n",a,b,*x,*y);
getch()
}

    void func1(int*x1, int*x2 ,int*x3, int*x4);
    {
        *x1=100; *x2=200; *x3=300;  *x4=400;
    }
```

Exam-21016

8. How pointers and arrays are closely related? What are the advantages of using pointer? What is the difference between int *p[5] and int(*p)[5]. Exam-2013
Question: What is the relation between pointer and single-dimension array? give example. Exam-2014
Question:What is a pointer? Is there any relation between a pointer and the name of a one-dimensional array? Exam-20
Q. What is pointer? How pointers and arrays are closely related? Briefly discuss with example?Marks: 3 Exam:ACCE-2014,13,ICE-2013,15 CSE-2011,2013

Pointer:

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

i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before you can use it to store any variable address.

Relation between array & pointer:

A pointer variable is used to store the address of another variable while an array is used to group related data items of same data type.
The name of the array is a pointer to the first element of the array. That means it holds the address of the very first element of the array.
For eg, if we declare an array - int arr[10];
Then - arr = &arr[0]

Consider and array:
int arr[4];

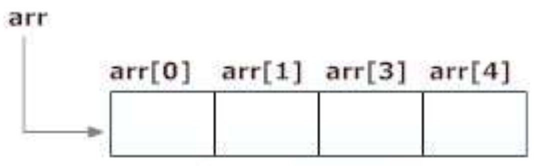


Figure: Array as Pointer

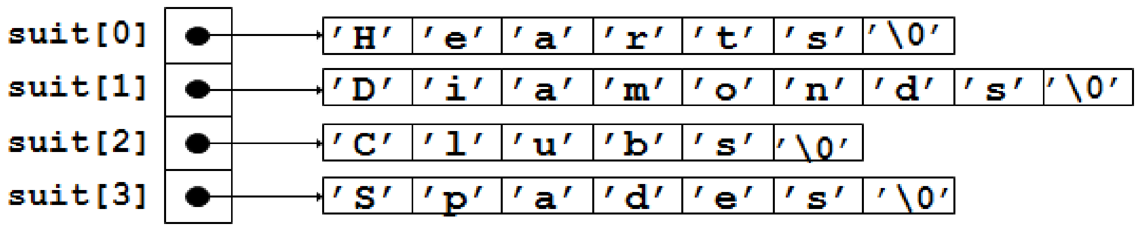
In arrays of C programming, name of the array always points to the first element of an array.

Here, address of first element of an array is &arr[0].
Hence, arr[0] is equivalent to *arr.
Similarly,
&a[1] is equivalent to (a+1) AND, a[1] is equivalent to *(a+1).
&a[2] is equivalent to (a+2) AND, a[2] is equivalent to *(a+2).
&a[3] is equivalent to (a+1) AND, a[3] is equivalent to *(a+3).
.
.
&a[i] is equivalent to (a+i) AND, a[i] is equivalent to *(a+i).
In C, you can declare an array and can use pointer to alter the data of an array.

Program to find the sum of six numbers with arrays and pointers.

```
#include <stdio.h>
int main(){
  int i,class[6],sum=0;
  printf("Enter 6 numbers:\n");
  for(i=0;i<6;++i){
      scanf("%d",(class+i)); // (class+i) is equivalent to &class[i]
      sum += *(class+i);   // *(class+i) is equivalent to class[i]
  }
  printf("Sum=%d",sum);
  return 0;
}
```

Question:Explain 'array name is a pointer'. Exam-2012
Answer:
Arrays can contain pointers
For example: an array of strings
char *suit[4] = { "Hearts", "Diamonds", "Clubs", "Spades" };


suit[0]
suit[1]
suit[2]
suit[3]
'H' 'e' 'a' 'r' 't' 's' '\0'
'D' 'i' 'a' 'm' 'o' 'n' 'd' 's' '\0'
'C' 'l' 'u' 'b' 's' '\0'
'S' 'p' 'a' 'd' 'e' 's' '\0'

- suit are pointers to the first character
- **char *** – each element of **suit** is a pointer to a **char**
- **suit** array has a fixed size, but strings can be of any size

....For Study:...

A Pointer is just an address of the data stored in memory.

Example:

Reference operator(&)

If *var* is a variable then, &var is the address in memory.

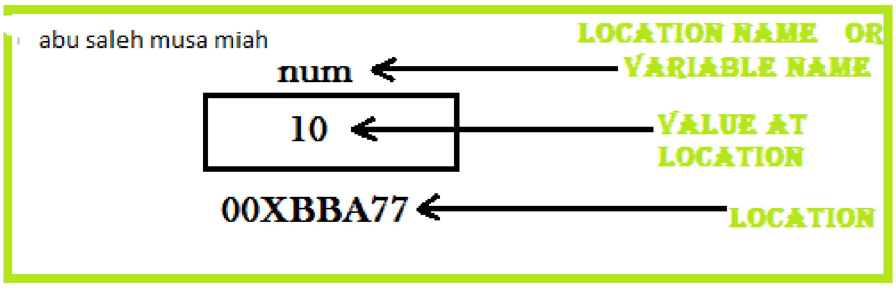
Consider the below code:

```
intvar=1;
int main()
{
int num =10;
   printf("Value of var is: %d", num);
   printf("Address of var is: %u",&num);
return0;
}
```

Output:

```
Value of varis:10
Address of varis:00XBBA77
```

n the above code we have two variables of type integer. One is num and another one is var. The value of num is 10 and this value should be stored somewhere in the memory, rite? Yes compiler allocates a memory space for each of the variable which is known as **address of the variable and it should be in hexadecimal format.**



& Operator – "Address of" Operator

As you can see that in above example, I have used & operator to find out the address of variable name. **& operator** is also known as "**Address of**" Operator.

printf("Address of var is: %u", &num);

*** Operator – "Value at Address" Operator**

* Operator is also known as **Value at address** operator.

As we discussed above, we can store the address of a variable in another variable which is known as pointer.

Q.How to declare a pointer?

Ans:

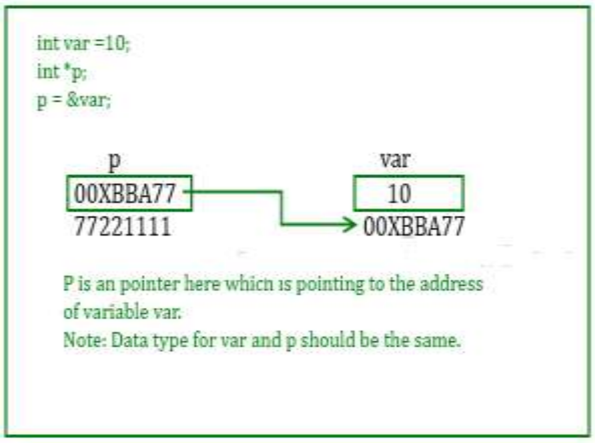
The general form of a pointer variable declaration is:

```
type *var-name;
int *ip;        pointer to an integer
double *dp;   Pointer to a variable of data type double to a double
float *fp;     pointer to a float
char *ch      pointer to a character
```

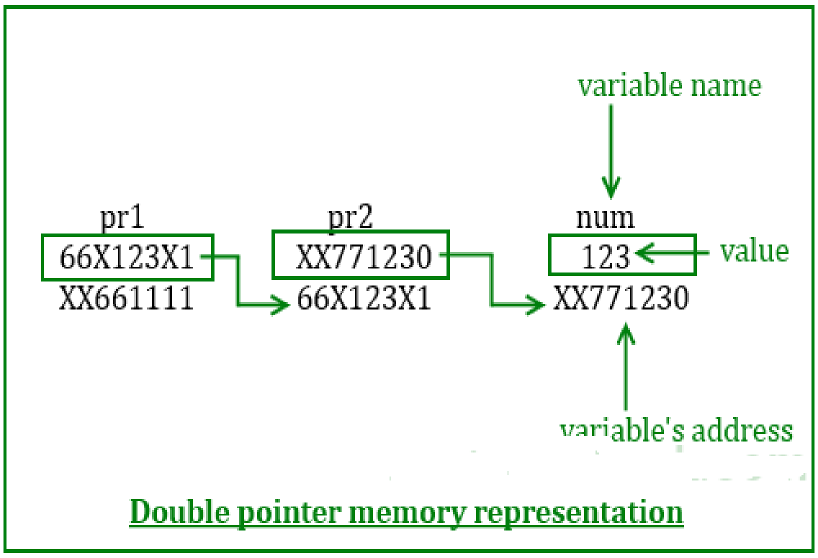
The above are the few examples of pointer declarations. **If you need a pointer to store the address of integer variable then the data type of the pointer should be int.**

Lets take the same example with a pointer variable –

```
int main()
{

int*p
intvar=10;
  p=&var;
printf("Value of var is: %d",var);
  printf("Address of var is: 
%u", p);
return0;

}
```


int var =10;
int *p;
p = &var;
p
00XBBA77
77221111
var
10
00XBBA77
P is an pointer here which is pointing to the address of variable var.
Note: Data type for var and p should be the same.

Consider the below figure and program to understand this concept better – Diagram



As per the figure, pr2 is a pointer for num (as pr2 is having address of variable num), similarly pr1 is a pointer for another pointer pr1 (as pr1 is having the address of pointer pr2). A pointer which points to another pointer is known as double pointer. In this example pr1 is a double pointer.

Values from above diagram –

Variable num has address: XX771230
Address of Pointer pr1 is: XX661111
Address of Pointer pr2 is:66X123X1

Close Relation Between Array And Pointer:

In C there is a very close connection between pointers and arrays. In fact they are more or less one and the same thing! When you declare an array as:

int a[10];

We are in fact declaring a pointer a to the first element in the array. That is, a is exactly the same as &a[0]. The only difference between a and a pointer variable is that the array name is a constant pointer - you cannot change the location it points at. **When you write an expression such as a[i]this is converted into a pointer expression that gives the value of the appropriate element. To be more precise, a[i] is exactly equivalent to *(a+i) i.e. the value pointed at by a + i . In the same way *(a+1) is the same as a[1] and so on.**
Being able to add one to a pointer to get the next element of an array is a nice idea, but it does raise the question of what it means to add 'one' to a pointer.

For example, in most implementations an int takes two memory locations and a float takes four. So if you declare an int array and add one to a pointer to it, then in fact the pointer will move on by two memory locations. However, if you declare a float array and add one to a pointer to it then the pointer has to move on by four memory locations. In other words, adding one to a pointer moves it on by an amount of storage depending on the type it is a pointer to.

This is, of course, precisely why you have to declare the type that the pointer is to point at! Only by knowing that a is a pointer to int and b is a pointer to float can the compiler figure out that

a + 1

means move the pointer on by two memory locations i.e. add 2, and

b + 1

means move the pointer on by four memory locations i.e. add 4. In practice you don't have to worry about how much storage a pointer's base type takes up. All you do need to remember is that pointer arithmetic works in units of the data type that the pointer points at. Notice that you can even use ++ and -- with a pointer, but not with an array name because this is a constant pointer and cannot be changed. So to summarise:
An array's name is a constant pointer to the first element in the array that is a==&a[0] and *a==a[0].
Array indexing is equivalent to pointer arithmetic - that is a+i=&a[i] and *(a+i)==a[i].
It is up to you whether you want to think about an array as an array or an area of storage associated with a constant pointer. The view of it as an array is the more sophisticated and the further away from the underlying way that the machine works. The view as a pointer and pointer arithmetic is more primitive and closer to the hardware. In most cases the distinction is irrelevant and purely a matter of taste.

Q.How to use Pointer? Why use Pointer or Advantage of pointer? Marks: Exam:ACCE-2013 ICE,CSE,APPECSE-2013

<u>How use Pointer:</u>

There are few important operations which we will do with the help of pointers very frequently.

(A) we define a pointer variable
(b) assign the address of a variable to a pointer and
(c) finally access the value at the address available in the pointer variable

Why use Pointer

They allow you to refer to large data structures in a compact way

- ❑ They facilitate sharing between different parts of programs
- ❑ They make it possible to get new memory dynamically as your program is running
- ❑ They make it easy to represent relationships among data items.

**Question:Explain the following C declarations. i) int *p[10]; ii) int (*p)[10];
(iii) int *p(char *a[ ]);Exam-2014**

<u>Answer</u>:

Question: What is the difference between int *p[5] and int(*p)[5]. Exam-2013

Answer:

- 🕮 int *a[5] - It means that "a" is an array of pointers i.e. each member in the array "a" is a pointer
  of type integer; Each member of the array can hold the address of an integer.
- 🕮 int (*a)[5] - Here "a" is a pointer to the array of 5 integers, in other words "a" points to an array that holds 5 integers.
- 🕮 int*p(char*[]): is a function named "p" that takes a char* argument "a" and returns a pointer to an int.

**Q.Suppose, the programmer want pointer pc to point to the address of c .
int c, *pc; pc=c; *pc=&c; Explain it.**

Ans;

```
pc=c;  /* pc is address whereas, c is not an address. */
*pc=&c; /* &c is address whereas, *pc is not an address. */
```

Question: Define dynamic memory allocation. Why it is required? Exam-2014
Ans:
<u>Dynamic Memory Allocation:</u>
Dynamic memory allocation allows a program to obtain more memory space, while running or to release space when no space is required.
Although, C language inherently does not has any technique to allocated memory dynamically, there are 4 library functions under "stdlib.h" for dynamic memory allocation.

Function	Use of Function
malloc()	Allocates requested size of bytes and returns a pointer first byte of allocated space
calloc()	Allocates space for an array elements, initializes to zero and then returns a pointer to memory
free()	dellocate the previously allocated space
realloc()	Change the size of previously allocated space

malloc()
The name malloc stands for "memory allocation". The function malloc() reserves a block of memory of specified size and return a pointer of type void which can be casted into pointer of any form.
Syntax of malloc()
ptr=(cast-type*)malloc(byte-size)
calloc()
The name calloc stands for "contiguous allocation".
Syntax of calloc()
ptr=(cast-type*)calloc(n,element-size);
free()
Dynamically allocated memory with either calloc() or malloc() does not get return on its own. The programmer must use free() explicitly to release space.
syntax of free()
free(ptr);
This statement cause the space in memory pointer by ptr to be deallocated.

<u>We Need To Use Dynamic Memory When:</u>

- we cannot determine the maximum amount of memory to use at compile time;
- we want to allocate a *very* large object;
- we want to build data structures (containers) without a fixed upper size;

Question:What are the difference between malloc() and calloc()?Exam-2013
Answer:
<u>Different between malloc and calloc function:</u>
There are two major differences between malloc() and call0c() in c programming language .

- First in the number of arguments , the malloc() takes a single argument while calloc() takes two.
- Second malloc() doesnot initialize the memory allocated while calloc() initializes the allocated memory to zero

Example of Dynamic memory Allocation:

Write a C program to find sum of n elements entered by user.

Ans:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
   int n,i,*ptr,sum=0;
   printf("Enter number of elements: ");
   scanf("%d",&n);
   ptr=(int*)malloc(n*sizeof(int));  //memory allocated using malloc
   if(ptr==NULL)
   {
      printf("Error! memory not allocated.");
      exit(0);
   }
   printf("Enter elements of array: ");
   for(i=0;i<n;++i)
   {
      scanf("%d",ptr+i);
      sum+=*(ptr+i);
   }
   printf("Sum=%d",sum);
   free(ptr);
   return 0;
}
Why required:
```

Q.Describe Use of Pointer ?

Ans: use of pointer:

- A primitive (int, char, float)
- An array
- A struct or union
- Dynamically allocated memory
- Another pointer
- A function

Q. **Passing pointers to functions in C with example?**

Ans: Pointers can also be passed as an argument to a function. Below example will help you understand how to do it.

```
int salaryhike(int*var,int hike)
{
*var=*var+hike;
return*var;
}
int main()
{
int sal=0;
int hike=0;
    printf("Enter the employee current salary:");
    scanf("%d",&sal);
    printf("Enter hike amount:");
    scanf("%d",&hike);
int op = salaryhike (&sal, hike);
    printf("Final salary: %d", op);
return0;
}
```

Output:

```
Enter the employee current salary:35000
Enter hike amount:3500
Final salary:38500
```

Q. Pointer to an array in C programming with example?

Or

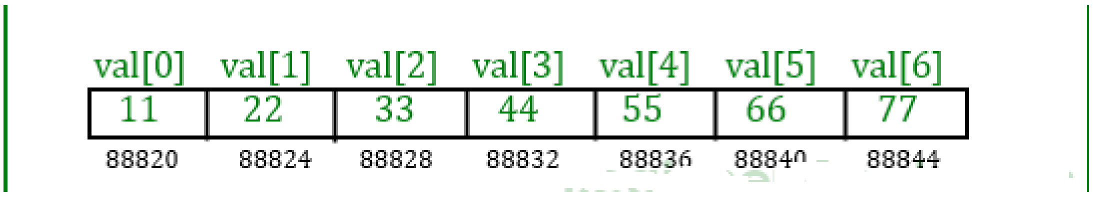
How pointer use as an array give example?

Ans:

```
include<stdio.h>
int main()
{
int val[7]={11,22,33,44,55,66,77};
for(int i =0; i <=6; i++)
{
     printf("val[%d]: value is %d and address is %u", i, val[i],&val[i]);
}
return0;
}
```

Output:

```
val[0]: value is11and address is88820
val[1]: value is22and address is88824
val[2]: value is33and address is88828
val[3]: value is44and address is88832
val[4]: value is55and address is88836
val[5]: value is66and address is88840
val[6]: value is77and address is88844
```



In the above example I have used &val[i] to get the address of ith element of the array. We can also use a pointer variable instead of it.

Consider the same above example with pointers –

```
#include<stdio.h>
int main()
{

int*p;
int val[7]={11,22,33,44,55,66,77};
   p =&val[0];
for(int i =0; i <=6; i++)
{
    printf("val[%d]: value is %d and address is %u", i,*p, p);
    p++;
}
return0;
}
```

The above program would result the same output, that our first program returned.

)
a[5]={10,20,30,40,50}

ptr=a

ptr++=20 ,++ptr=30 ++*ptr=31

Differenc *ptr++ AND (*ptr)++

Question:Explain the output of the following C code: Exam-2014

```
main() {
int *a, b = 30;
    a = &b;
    b = *a + 40;
    *a = b /5;
printf("%d %d ", *a , b);   }
```

Answer:

A=14;

B=14

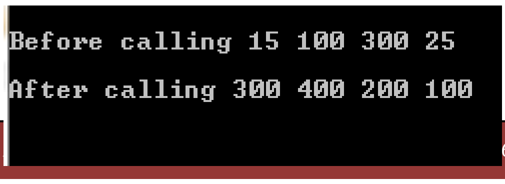
Question: What will be the output of following code ?Exam-21016

(Objective of the question : Check the concept of pointers)

```
    void func1(int*p, int*q ,int*r, int*s);

main(){
inta,b,c,d,*x,*y;
a=15;b=100,x=&d;y=&c;c=25;d=300;
printf("Before calling %d %d %d %d \n",a,b,*x,*y);
func1(&c,&d,&a,&b);
printf("After calling %d %d %d %d \n",a,b,*x,*y);
getch()
}
    void func1(int*x1, int*x2 ,int*x3, int*x4);
    {
        *x1=100; *x2=200; *x3=300;  *x4=400;
    }
```

Exam-21016

Answer:



```
Before calling 15 100 300 25
After calling 300 400 200 100
```

3.4 STRING

Importent Question

1. What sis the difference between ‘s’ and ‘’s”. **Exam-2013**

Q. Write a program to display character from A to Z using loops.
Ans:

```
#include<stdio.h>
int main()
{
char c;
for(c='A'; c<='Z';++c)
    printf("%c ",c);
return0;
}
```

Output

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Q.Write a program find vowel or consonant?
Ans:

```
#include<stdio.h>
int main(){
char c;
  printf("Enter an alphabet: ");
  scanf("%c",&c);
if(c=='a'||c=='A'||c=='e'||c=='E'||c=='i'||c=='I'||c=='o'||c=='O'||c=='u'||c
=='U')
    printf("%c is a vowel.",c);
else
    printf("%c is a consonant.",c);
return0;
}
```

Output 1

```
Enter an alphabet: i
i is a vowel.
```

Output 2

```
Enter an alphabet: G
G is a consonant.
```

Q. Write a program to display character from A to Z using loops either in uppercase or lowercase depending upon the data from user ?
Ans:

```
#include<stdio.h>
int main()
{
char c;
   printf("Enter u to display characters in uppercase and l to
display in lowercase: ");
   scanf("%c",&c);
if(c=='U'|| c=='u')
{
for(c='A'; c<='Z';++c)
      printf("%c ",c);
}
if(c=='L'|| c=='l')
{
for(c='a'; c<='z';++c)
      printf("%c ",c);
}
if(c!='U'|| c!='L'|| c=='u'|| c=='l')
      printf("Error !!!");
return0;
}
```

Output

```
Enter U to display characters in Uppercase and L to display in lowercase: L
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

Q. What is string? CSE,ICE,APPE

Ans:

A string is a sequence of characters that is treated as single data item. Any group of characters defined between double quotation marks is a string constant.

If we want to read complete sentence. or string is a set of character.

Example;

"c is programming language"

Q how you declare and initialize a string variable?

Ans:

Delclaration of Sring:

Since **string** is an array, the declaration of a string is the same as declaring a char array.

Or

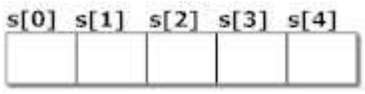
String is nothing but an array of characters.

char **string name [size];**

The size determines the numbers of characters in the string name.

Strings are declared in C in similar manner as arrays. Only difference is that, strings are of char type.

char s[5];



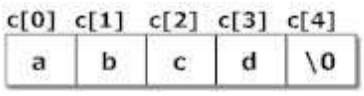
Strings can also be declared using pointer.

char *p

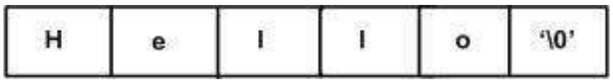
Initialization string:

String can be initialized in different number of ways

```
char c[]="abcd";
   OR,
char c[]={'a','b','c','d','\0'};
   OR;
char c[5]={'a','b','c','d','\0'};
   OR,
char c[5]="abcd";    //Here declaration NULL character (\0) will automatically be
inserted at the end of the string.
```



char students[6] = "Hello"



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

String can also be initialized using pointers
char *c="abcd";

Q.How Reading string from user?
Ans:

Reading a words from user:

```
char c[20];
scanf("%s",c);
```

String variable *c* can only take a word. It is because when white space is encountered, the scanf() function terminates.

Write a C program to illustrate how to read string from terminal?

```
#include<stdio.h>
int main(){
char name[20];
   printf("Enter name: ");
   scanf("%s",name);
   printf("Your name is %s.",name);
return0;
}
```

Output

```
Enter name: Dennis Ritchie
Your name is Dennis.
```

ere, program will ignore Ritchie because, scanf() function takes only string before the white space.

Reading a line of text:

C program to read line of text manually.

```
#include<stdio.h>
int main(){
char name[30],ch;
int i=0;
   printf("Enter name: ");
while(ch!='\n')// terminates if user hit enter
{
      ch=getchar();
      name[i]=ch;
      i++;
}
   name[i]='\0';// inserting null character at end
   printf("Name: %s",name);
return0;
}
```

This process to take string is tedious. There are predefined functions gets() and puts in C language to read and display string respectively.

```
int main(){
char name[30];
   printf("Enter name: ");
   gets(name);//Function to read string from user.
   printf("Name: ");
   puts(name);//Function to display string.
return0;
}
Or
   int main(){
char name[30];
   printf("Enter name: ");
  scanf("%[^/n]",name);//Function to read string from user.
   printf("Name: ");
   puts(name);//Function to display string.
return0;
}
```

Both, the above program has same output below:

Output

```
Enter name: Tom Hanks
Name: Tom Hanks
```

Question:What sis the difference between 's' and "s". Exam-2013
Q. what is the difference between 'A' and "A"? Exam:ACCE-2012,CSE,MSE
Ans:
here 'A' is a charater type data and "A" is a string . in program
Char ch='A'-------------→1 byte
Char ch[]="A"-------→2 byte;

Q. List four string related standard library function with their prototype?
Ans:
Include <string.h> as a header file. The following functions are available for use.
•Concatenate two strings: **strcat(s1, s2)**
•Compare two strings : **strcmp(s1, s2)**
•Length of string : **strlen(s)**
•Copy one string over other: **strcpy(s1, s2)**
–Here contents of s2 are copied to s1
•Locating substring: **strstr(s1,s2)**
–Gives the position of s1 in s2

Q. How do you use pointer in string? ICE-2013
Ans:
c supports an alternative method to crate strings using pointer variables of type char.

Example:
 Char * str="good"
This creates a string for the literal and stores its address in the pointer variable str.
 We can also use the runtime assignment for giving values to a string pointer. Example

```
Char * string 1;
 String 1="good";
```

Let a example to explain it

```
 Int main()
{ char * st;
Printf("Enter your text:");
Scanf("%s",st);
 Printf("%s",st);
Return 0;
}
```

Q. what the difference between array and string?
Ans:
following are the differences:

Array	string
Where as an array can hold any data type.	String can hold only char data
- An array size can not be changed	.. Where as a string size can be changed if it is a char pointer
The last element of an array is an element of the specific type	The last character of a string is a null – '\0' character
The length of an array is to specified in [] at the time of declaration (except char[]).	The length of the string is the number of characters + one (null character
Example : char str[100];	Example: *str[10]

To explain pointer as example:

```
 int main(){
   char *str[5];
  Int I;
printf("Enter your text");
For(i=0;i<5;i++)
  { scanf("%s",str[i]);
for(i=0;i<n;i++)
{ printf("%s",str[i]);
}
return 0;}}..
-
```

Q. How can you use string variable to the function parameter?
Ans: Since the strings are treated as character arrays in c. the rules for passing strings to functions are very6 similar to those for passing array to function.
basic rules are

1. The string to be passed must be declared as a formal argument of the when it is defined as

```
Void display(char item_name[])
    {
    }
```

2. The function prototype must show that the argument is a string . for the above function

Definition, the prototype can be written as

```
    Void display (char str[]);
```

3. A call th the function must have a string name without subscripts.As its actual arguments As,

```
     Display(names);
```

Where names is a properly declared string array in the calling function. Let an example to explain it .

```
int display(char str[]);
              int main()
                     {
                   Char str[100];
                   printf("Enter your text :");
                   gets(str);
                   display(str);
                   return 0;
                     }
              int display(str[])
                        {
                         printf("display your text:");
                         printf("%s",str);
                          }
```

Q. What is input function of a string? Exam:ACCE-2011,ICE-2015,APPE

Ans:

There are different types of input function oin string. Such as

(a) scanf
(b) getchar
(c) gets

(a) **scanf**:

the familiar input function scanf can be used with %s format specification to read in a strin g of characters. Unlike preyious sccanf calls, in the case of character arrays. Ampersand (&) is not required before the variable name.

If we want to input a single word:

```
Char address[10]
Scanf("%s",address);
```

If we want to input a multiple word until \n found:

```
scanf(" %[^\n]s",name);
```

\n just sets the delimiter for the scanned string

(b) **getchar:**

The **char getchar** function reads the next available character from the screen and returns it as an integer. This function reads only single character at a time. You can use this method in the loop in case you want to read more than one character from the screen.

```
Char ch;
Ch =getchar();
```

The getchar function has no parameter.

(c) gets:

The **char gets()** function reads a line from **stdin** into the buffer pointed to by **s** until either a terminating newline or EOF (End of File).

Example:

```
char str[100];
printf("Enter a value :");
gets( str )
```

Q What is the output function of string? Exam:ICE-2015,ACCE-2013

Ans:

C supports different types of output function such as

(a) printf
(b) Putchar
(c) puts

(a)printf:

We have used extensively the printf function with %s format to print strings to output screen. The format %s can be used to display an array of characters that is terminated by the null character. For example the statement.

Printf("%s",st);

We can also specify the precision with which the array is displayed. For instance the specification

Printf("%10.4s",st);

This prints the first d acahracters of the string in the field width of w.

Char st[]="ICE"

Printf("%d.*s",d.st**);**

(b)putchar:

The **putchar** function puts the passed character on the screen and returns the same character. This function puts only single character at a time. You can use this method in the loop in case you want to display more than one character on the screen. Check the following example

```
printf("\nYou entered: ");
  putchar( c );
```

(c)puts:

The **puts** function writes the string 's' and 'a' trailing newline to **stdout.**prints the string pointed to by str to the screen

```
printf("\nYou entered: ");
  puts( str );
```

Q. What do you mean by two dimensional string?

Ans:

We often use lists of character strings. Alist of name can be treated as table of string and a two dimentional character array can be used to strore the entire list.

For example a character array.

Strudent[30][5];

- char names[People][Length];
- char month[5][10] = {"January", "February", "March", "April", "May"};

It may be used to strore a list of 30 names eache of length not more than 15 characters.

Q. write a program that accepts a string from the user and calculates its length without using any string related library functions.
Ans:

```
 #include<stdio.h>
  int main()
       {int l;
char st[100];
printf("Enter any string:");
 scanf("%s",st);
for(l=0;st[l];l++);
printf("Length %d",l);
return 0;       }
```

Q. describe the functions of strcmp() and strcpy()
Ans:

Strcmp():
Description
The C library function int strcmp(const char *str1, const char *str2) compares the string pointed to, by str1 to the string pointed to by str2.
Declaration
Following is the declaration for strcmp() function.
int strcmp(char*str1,char*str2)
Parameters

- str1 -- This is the first string to be compared.
- str2 -- This is the second string to be compared.

Return Value
This function return values that are as follows:

- if Return value < 0 then it indicates str1 is less than str2.
- if Return value > 0 then it indicates str2 is less than str1.
- if Return value = 0 then it indicates str1 is equal to str2.

Example

Q. Write a program that read two string and compare them and print largest of them?

The following example shows the usage of strncmp() function.

```
#include <stdio.h>
#include <string.h>
int main ()
{   char str1[15];
char str2[15];
  int ret;
  strcpy(str1, "abcdef");
  strcpy(str2, "ABCDEF");
  ret = strcmp(str1, str2);
  if(ret < 0)   {
    printf("str1 is less than str2");
  }
  else if(ret > 0)
  {
    printf("str2 is less than str1");
  }
  else
  {
    printf("str1 is equal to str2");
  }
  return(0);
}
```

Let us compile and run the above program that will produce the following result:
str2 is less than str1

strcpy():
Description
The C library function **char *strcpy(char *dest, const char *src)** copies the string pointed to, by **src** to **dest**.
Declaration
Following is the declaration for strcpy() function.
char *strcpy(char *dest, const char *src)
Parameters

- **dest** -- This is the pointer to the destination array where the content is to be copied.
- **src** -- This is the string to be copied.

Return Value
This returns a pointer to the destination string dest.
Example
The following example shows the usage of strcpy() function.

Q.Write a program that read a string and copy this string into another variable?
Ans;

```
#include <stdio.h>
#include <string.h>
int main()
{
  char src[40];
  char dest[100];
  memset(dest, '\0', sizeof(dest));
  strcpy(src, "This is tutorialspoint.com");
  strcpy(dest, src);
  printf("Final copied string : %s\n", dest);

  return(0);
}
```

Let us compile and run the above program that will produce the following result:
Final copied string : This is tutorialspoint.

3.5 STRUCTURE AND UNION

Importent Question

1. Distinguish between structure and union. **Exam-2014,2012**
2. Define self-referential structure with example. **Exam-2014**
3. Define a structure of name 'student' that has the members: student name, roll as integer and grade as floating. **Exam-2012**
   (a).How many bytes are required for the above structure definition? **Exam-2012**
   (b).How many bytes are required if the above definition was 'union' instead of 'structure'? **Exam-2012**
4. What are the advantages of structure over array? **Exam-2012**
5. Write a structure named 'familyInfo' having the members: 'mothername' and 'fathername' as string, 'childNo' as integer, 'income' and 'expenses' as floating point numbers. **Exam-2015**
   (a).How many bytes are required for the above mentioned structure definition?
   (b).How many bytes are required if the above definition is 'union' instead of 'structure'? Write a C program using that structure to take 10 families' information from keyboard and save into structure.
   Exam-2015
6. Write a program using structure that will allow you to enter and display the following information of your family:
   (i). Name (ii). Age (iii).Occupation (iv) Salary. **Exam-2013.**
7. What is the difference between "structure" and "union"? Give an example to explain it. **Exam-2015**
8. What is wrong with the following code? Correct the error, and also give the output.

```
#include<stdio.h>
int main()
   {
   Structs{
    Char  *s;
                                  }
        a={"Ritche"}, *p=&a;
           printf("%s",*p.s);
                   }
                }
```

Q. What is structure? How a structure is declared? How does a structure differ from an array? Marks:3 Exam-ACCE-2014
Ans:
Structure:

A structure is a collection of variables under a single name. These variables can be of different types, and each has a name which is used to select it from the structure. A structure is a convenient way of grouping several pieces of related information together.

<u>**Declaring Structure Variable in C:**</u>

In C we can group some of the user defined or primitive data types together and form another compact way of storing complicated information is called as Structure. Let us see how to declare structure in c programming language

syntax Of Structure in C Programming :

```
struct tag
{
  data_type1 member1;
  data_type2 member2;
  data_type3 member3;
};
```

Question: How does a structure differ from an array? How the embers of a structure are accessed? Exam-2013
Question: What are the advantages of structure over array? Exam-2012
Q. What is the difference between array and structure? Exam:ACCE-2011,ICE-2013,CSE
Ans:

The following are the differences between structures and arrays:

Array elements are homogeneous. Structure elements are of different data type.
Array allocates static memory and uses index / subscript for accessing elements of the array. Structures allocate dynamic memory and uses (.) operator for accessing the member of a structure.
Array is a pointer to the first element of it. Structure is not a pointer
Array element access takes less time in comparison with structures.

Question: Define self-referential structure with example. Exam-2014,2012

Self-Referential Structure:

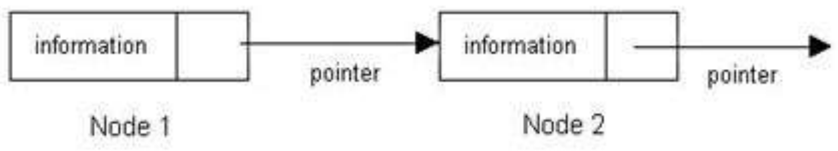
A self referential data structure is essentially a structure definition which includes at least one member that is a pointer to the structure of its own kind.

A chain of such structures can thus be expressed as follows.

```
struct name {
                member 1;
                member 2;
                . . .
                struct name *pointer;
     };
```

The above illustrated structure prototype describes one node that comprises of two logical segments. One of them stores data/information and the other one is a pointer indicating where the next component can be found. .Several such inter-connected nodes create a chain of structures.

The following figure depicts the composition of such a node. The figure is a simplified illustration of nodes that collectively form a chain of structures or linked list.



Such self-referential structures are very useful in applications that involve linked data structures, such as lists and trees. Unlike a static data structure such as array where the number of elements that can be inserted in the array is limited by the size of the array, a self-referential structure can dynamically be expanded or contracted. Operations like insertion or deletion of nodes in a self-referential structure involve simple and straight forward alteration of pointers.

Example:

```
typedef struct NODE {
        struct NODE *new;
        int value;
  }Node;

int main(void)
{
  Node previous, current;

  /* accessing members of 'previous' */
  previous.new = &current;
  /* previous.new is a 'pointer-to-struct NODE' */
  previous.value = 100;
}
```

Q. How to assign values to struct members?

Ans:

There are three ways –

1) Using Dot(.) operator

```
var_name.memeber_name = value;
```

2) Whole structure's variables assignment in one go –

```
    struct struct_name    var_name =
  {value for memeber1, value for memeber2 ...so on for all the members}
```

3) Designated initializers –

Example:

```
#include<stdio.h>
structStudentData{
char*stu_name;
int stu_id;
int stu_age;
}

int main()
{
structStudentData   student;
    student ={"Chaitanya",1234,25};

    printf("Student Name is: %s", student.stu_name);
return0;
}
```

There are two things to notice in above example –
1) Both the below code snippets are same.

```
student ={"Chaitanya",1234,25};
OR(both are same!!)
student.stu_name ="Chaitanya";
student.stu_id =1234;
student.stu_age =25;
```

2) Dot (.) operator can be used to assign values to individual members of structure or can be used to access values stored in structure's individual members.

Q. What is union? What is its general form? Give an example showing how different types of data share the same memory location? Marks:3 Exam-ICE-2013,CSE-ACCE-2014

Unions:

A union is a special data type available in C that enables you to store different data types in the same memory location.
or
A *union*is an object similar to a structure except that all of its members start at the same location in memory.

general form of union:

```
union [union tag]
{
member definition;
member definition;
...
member definition;
} [one or more union variables];
```

The **union tag** is optional and each member definition is a normal variable definition, such as int i; or float f; or any other valid variable definition. At the end of the union's definition, before the final **semicolon**

Example:

```
union number
{
short shortnumber;
long longnumber;
double floatnumber;
} anumber
```

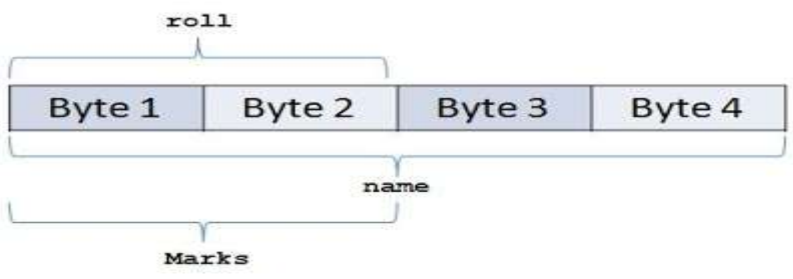
In the above example this is 8 bytes for the double.

Q.Explain how memory is allocated for a union in C? Exam:2011ICE-2013

```
int roll;
char name[4];
int marks;
```

We have collected three variables of different data type under same name together.

All Union Members Occupy Same Memory Area



For the union maximum memory allocated will be equal to the data member with maximum size. In the example character array 'name' have maximum size thus maximum memory of the union will be 4 Bytes.

Maximum Memory of Union = Maximum Memory of Union Data Member

Only one Member will be active at a time.

Suppose we are accessing one of the data member of union then we cannot access other data member since we can access single data member of union because each data member shares same memory. By Using Union we can **Save Lot of Valuable Space**

Simple Example :

```
union u
{
char a;
int b;
}
```

Question: What is the difference between "structure" and "union"? Give an example to explain it. Exam-2015,2014
Q.Distinguish between structure and union? Marks:1.50 Exam-ACCE-2013
Ans:

Difference between structure and union in C	
structure	**union**
Keyword struct defines a structure.	Keyword union defines a union.

Example structure declaration:
struct s_tag
{
 int ival;
 float fval;
 char *cptr;
}s;
	Example union declaration:
	union u_tag
	{
	 int ival;
	 float fval;
	 char *cptr;
	}u;

Within a structure all members gets memory allocated and members have addresses that increase as the declarators are read left-to-right. That is, the members of a structure all begin at different offsets from the base of the structure. The offset of a particular member corresponds to the order of its declaration; the first member is at offset 0. The total size of a structure is the sum of the size of all the members or more because of appropriate alignment.	For a union compiler allocates the memory for the largest of all members and in a union all members have offset zero from the base, the container is big enough to hold the WIDEST member, and the alignment is appropriate for all of the types in the union. When the storage space allocated to the union contains a smaller member, the extra space between the end of the smaller member and the end of the allocated memory remains unaltered.
Within a structure all members gets memory allocated; therefore any member can be retrieved at any time.	While retrieving data from a union the type that is being retrieved must be the type most recently stored. It is the programmer's responsibility to keep track of which type is currently stored in a union; the results are implementation-dependent if something is stored as one type and extracted as another.
One or more members of a structure can be initialized at once.	A union may only be initialized with a value of the type of its first member; thus union u described above (during example declaration) can only be initialized with an integer value.

Question: Define a structure of name 'student' that has the members: student name, roll as integer and grade as floating.
a).How many bytes are required for the above structure definition?
(b).How many bytes are required if the above definition was 'union' instead of 'structure'? Exam-

Question: Write a structure named 'familyInfo' having the members: 'mothername' and 'fathername' as string, 'childNo' as integer, 'income' and 'expenses' as floating point numbers. Exam-2015
(a)How many bytes are required for the above mentioned structure definition? Exam-2015

(b) How many bytes are required if the above definition is 'union' instead of 'structure'?
Exam-2015
Question: Write a C program using that structure to take 10 families' information from keyboard and save into structure.
Exam-2015

Question: Write a program using structure that will allow you to enter and display the following information of your family:
(i). Name (ii). Age (iii).Occupation (iv) Salary. Exam-2013.
Question: What is wrong with the following code? Correct the error, and also give the output.

Q.Write a program that read name and roll and print?

```
#include <stdio.h>
struct student{
   char name[50];
   int roll;
};
int main(){
   struct student s1;
   printf("Enter student's name: ");
   scanf("%s",&s1.name);
   printf("Enter roll number:");
   scanf("%d",&s1.roll);
   printf("Output\nName: %s",s1.name);
   printf("\nRoll: %d",s1.roll);
   return 0;
}
```

Q.Discuss Characteristics of Structure?
Ans:

1) A struct is a collection of other data types.
2) The length of a struct is fixed based on the data type of the members.
3) A struct can be nested.
4) typedef creates a type name for a struct and makes the code easier and readable.
5) struct fields can be accessed using Dot(.) operator.

Q. Write a program using structure that store 100 student's name roll and total marks?
Ans:

```
#include  <stdio.h>
#include  <conio.h>
int main()
    {
    struct student
         {
         int rollno;
         char name[25];
         int totalmark;
         }stud[100];
    int n,i;
    clrscr();
    printf("Enter total number of students\n\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
```

```
            {
            printf("Enter details of %d-th student\n",i+1);
            printf("Name:\n");
            scanf("%s",&stud[i].name);
            printf("Roll number:\n");
            scanf("%d",&stud[i].rollno);
            printf("Total mark:\n");
            scanf("%d",&stud[i].totalmark);
            }
 printf("STUDENTS DETAILS:\n");
 for(i=0;i<n;i++)
          {
          printf("\nRoll number:%d\n",stud[i].rollno);
          printf("Name:%s\n",stud[i].name);
          printf("Totel mark:%d\n",stud[i].totalmark);
         }
getch();
return 0;
}
```

Important Question

1. What is a function? State three advantage to use of functions. **Exam-2013,2016**
2. What do you mean by 'function declaration' and 'function definition'? **Exam-2012,2015**
3. Compare global variable and local variable with example. **Exam-2014,2015**
4. What is 'recursive function'? Give an example. **Exam-2012**
5. Write a function in C that will take two integers as argument and return the largest value. **Exam-2012**
6. Suppose an array is passed to a function as an argument .if the value of an array element is altered within the function, will this change be recognized within the calling portion of the program. **Exam-2013**
7. What are the difference between passing an array to a function and passing a single-valued item to a function ?**Exam-21016**
8. Summarize the rules governing the use of the return statement. Can multiple expressions be included in a return statement/ can multiple return statements be included in a function. **Exam-2013**
9. Define function and function prototype with examples. Why function is used in C? **Exam-2014**
10. Explain formal and actual parameter with examples. **Exam-2014**
11. Write a recursive function in C programming that will return the sum of the series 1+2+3+...+n. Here n is a positive integer. **Exam-2014**
12. Distinguish between " function call by value " and "function call by reference " . Explain with example. **Exam-2014**
13. Define command line arguments. Give an example for passing command line argument in a C program. **Exam-2014**
14.) What will be output of the following code?**Exam-2015**

```
main() {
    inta,b,c,d,*x,*y;
    a=5;b=10;x=&c;y=&d;c=20;d=30;
    printf("Before calling %d %d %d %d\n",a,b,*x,*y);
    func1(&c,&d,&a,&b);
    printf("After calling %d %d %d %d\n",a,b,*x,*y);
}
Void func1(int *p,int *q, int *r, int *s){
    *p=100; *q=200; *r=300; *s=400;
}
```

15. What will be the output of the following code ? If you think any value displayed may be garbage , mention it as garbage too .

```
Void func1(int a, int p, int q[]);
        Int x, y;
        Main(){
                     Inta,b,c[3];
                     A=10;b=20;c[1]=1,c[2]=2;
                     X=100;y=200;
                     Printf("Before %d %d %d %d %d %d\n",a,b,c[1],c[2],x,y);
Func1(a,b,c);
Printf("After %d %d %d %d %d %d\n",a,b,c[1],c[2],x,y);
getch();
}
Void func1(int a, int p, int q[]){
        Int x;
        a=100;
        b=200;
        q[1]=q[1]+2;
        q[2]=q[2]+2;
        x=102;
        y=202;
}
```
Exam-2015

16. Explain how the variables take the values .**Exam-2016**

(Objective of the question : To check concepts of local and global variables)

```
#include<stdio.h>
#include<conio.h>
voidadd_int(int n);
int main(){
int p;
q=200;
x=10;
printf("\nBefore calling x=%d p=%d q=%d",x,p,q);
add_int(x);
printf("\nAfter calling x=%d p=%d q=%d",x,p,q);
}
voidadd_int(int x){
x=50;
p=200;
q=300;
}
```

17. What will be the output of the following code?**Exam-2015**

Question: What is a function? State three advantage to use of functions. Exam-2013,2016

Answer:

Function:

Function is a group of statements that together perform task.

Or

Functions are "self contained" modules of code that accomplish a specific task. Functions usually "take in" data, process it, and "return" a result. Once a function is written, it can be used over and over and over again. Functions can be "called" from the inside of other functions.

Necessary of function:

Functions are used because of following reasons –

a) To improve the readability of code.

b) Improves the reusability of the code, same function can be used in any program rather than writing the same code from scratch.

c) Debugging of the code would be easier if you use functions as errors are easy to be traced.

d) Reduces the size of the code, duplicate set of statements are replaced by function calls.

Question Define function and function prototype with examples. Why function is used in C? Exam-2014

Q.What is a function prototype and what are the benefits of it? Exam:ACCE-2013

Answer:

Function Prototype:

A function prototype declares a function before it is used and prior to its definition. A prototype consists of a functions' name, its return type and its parameter list. It is terminated by a semicolon. The compiler needs to know this information in order for it to properly execute a call to the function.

```
For example:
Int myfunc()
 {
Printf("this is a test");
}
Its prototyps is
Int myfunc();
```

The only function that does not need a prototype is main() since it is predefined by the c language.

Question What do you mean by 'function declaration' and 'function definition'? Exam-2012,2015

Answer:

Q.Describe about function Declaration,function arguments and function calling with example?

Ans:

Function Declarations

A **function declaration** tells the compiler about a function name and how to call the function. The actual **body of the function** can be defined separately.

A function declaration has the following parts:

return_type function_name(parameter list);

Suppose we have a function name is max(), following is the function declaration:

int max(int num1, int num2);

Parameter names are not important in function declaration only their type is required, so following is also valid declaration:

int max(int, int);

The Function Definition:

The general form of a function definition in C programming language is as follows:

```
return_type function_name( parameter list )
{
body of the function
}
```

A function definition in C programming language consists of a function header and a function body.

Calling a Function:

When a program calls a function, program control is transferred to the called function. A called function performs defined task, and when its return statement is executed or when its function-ending closing brace is reached, it returns program control back to the main program.

Q.What is the difference between function declaration and function definition with example?
Answer:

A declaration provides basic attributes of a symbol: its type and its name.

int func();

This is a function declaration; it does not provide the body of the function, but it does tell the compiler that it can use this function and expect that it will be defined somewhere.

. Defining a function means providing a function body;
You can write code like this:

```
int func();
int main()
{
   int x = func();
}
int func()
{
   return 2;
}
```

Since the compiler knows the return value of func, and the number of arguments it takes, it can compile the call to func even though it doesn't yet have the definition. In fact, the definition of the method func could go into another file!

A definition provides all of the details of that symbol--if it's a function, what it does; if it's a class, what fields and methods it has; if it's a variable, where that variable is stored
. If no source file ever defines a symbol, but it is declared, you will get errors at link time complaining about undefined symbols

Q. Distinguish between automatic and static variable?
Ans:

Static variable	Auto variable
We have to specify the storage class to make a variable static.	It is the default storage class.
If it is not assigned any value then it will give 0 as out put.	If it is not assigned any value then it will give garbage value as output.
It is visible to the block in which it is declared and also in the function where it will passed.	It is visible to the block in which the variable is declared.
It retains its value between different function calls. It holds its last value.	It retains its value till the control remains in the block in which the variable is declared.
Static variable should be compile by compiler first.	Auto variable will compile by the compiler after the static variable.

Q. Distinguish between Global and extern variable.
Ans:
Global variable: A variable whose value can be accessed and modified by any statement in a program, not merely within a single routine in which is defined.
Extern variable: Global variable known to all functions in the file.

Question Compare global variable and local variable with example. Exam-2014,2015
Q. What is difference between local and global variable? Exam.Acce 2014,13
Answer:

Local	**Global**
1.Locals are defined inside a function.	1. Globals are defined out side a function.
2.Local variables are known only to the block in which it is defined.	2.Global variables are known through out the entire program.
3.To storage for local variable is on the stack.	3. storage for global variable in a fixed region of memory set.

Q.Describe about the several part of function?
Ans:
Here are all the parts of a function:

- **Return Type**: A function may return a value. The **return_type** is the data type of the value the function returns.
- **Function Name:** This is the actual name of the function. The function name and the parameter list together

- **Parameters:** A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument.

- **Function Body:** The function body contains a collection of statements that define what the function does.

Q.Write a program that explain user define function and calling that function from main function?

```
                #include <stdio.h>
………………….. ……………………………..function declaration……………………..
                int max(int num1, int num2);
                int main ()
                {
                ………………………………………………... local variable definition ………………………
                int a = 100;
                int b = 200;
                int ret;
                ……………………………………………….. calling a function to get max value
                ………………………..
                ret = max(a, b);
                printf( "Max value is : %d\n", ret );
                return 0;
                }
                …………………………….. function returning the max between two numbers …..
                int max(int num1, int num2)
```

```
{
int result;
if (num1 > num2)
result = num1;
else
result = num2;
return result
}
```

Function Arguments:

If a function is to use arguments, it must declare variables that accept the values of the arguments. These variables are called the formal parameters of the function.

Q.What are the different ways of passing parameters to the functions? Which to use when?
Ans:

- **Call by value** − We send only values to the function as parameters. We choose this if we do not want the actual parameters to be modified with formal parameters but just used.
- **Call by reference** − We send address of the actual parameters instead of values. We choose this if we do want the actual parameters to be modified with formal parameters.

Q. Distinguish between function call by value and function call by value and function call by reference. Explain with example? Exam-acce-2014
Q.What is Call by value and Call by reference describe with example?
Ans:

Function call by value:

The **call by value** method of passing arguments to a function copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.
It is the default way of calling a function in C

Example:

```
int increment(intvar)
{
var=var+1;
returnvar;
}

int main()
{
int num1=20;
int num2 = increment(num1);
  printf("num1 value is: %d", num1);
  printf("num2 value is: %d", num2);
return0;
}
Output:
num1 value is:20
num2 value is:21
```

Why did it happen? num1 is still have 20 even after increment operation, why?
The reason is simple, function is called by value in above program, which means num1's value gets copied into var and the variable var got incremented (not variable num1), which later stored in num2, via call.

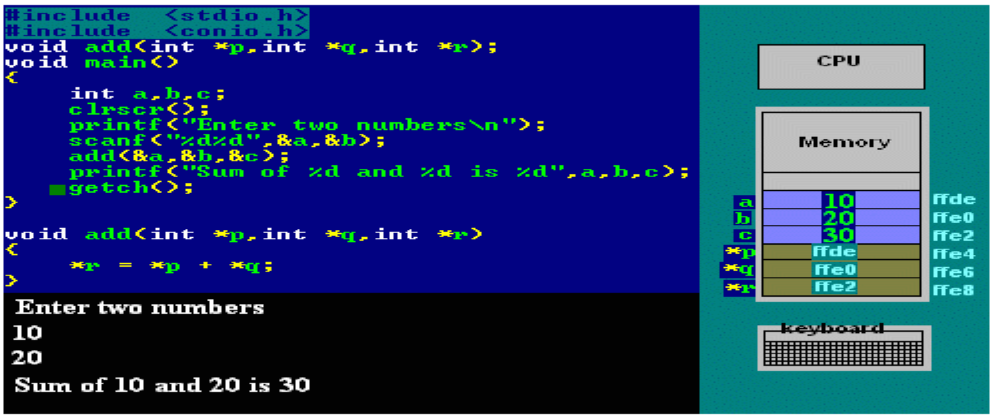
Function Call By Reference:

The **call by reference** method of passing arguments to a function copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call. This means that changes made to the parameter affect the passed argument.

```
int increment(int*var)
{
*var=*var+1;
return*var;
}
int main()
{
int num1=20;
int num2 = increment(&num1);
    printf("num1 value is: %d", num1);
    printf("num2 value is: %d", num2);
return0;
}
```
Output:

```
num1 value is:21
num2 value is:21
```

Unlike "call by value", in this method the value of num1 got changed because the address of num1 is passed as an argument so the increment operation is performed on the value stored at the



```
#include   <stdio.h>
#include   <conio.h>
void add(int *p,int *q,int *r);
void main()
{
     int a,b,c;
     clrscr();
     printf("Enter two numbers\n");
     scanf("%d%d",&a,&b);
     add(&a,&b,&c);
     printf("Sum of %d and %d is %d",a,b,c);
     getch();
}

void add(int *p,int *q,int *r)
{
     *r = *p + *q;
}
```

```
Enter two numbers
10
20
Sum of 10 and 20 is 30
```

address.Figure another example of call by reference:

Question Explain formal and actual parameter with examples. Exam-2014
Q.What is the difference between actual and formal parameters?
Ans:
The parameters sent to the function at calling end are called as actual parameters while at the receiving of the function definition called as formal parameters.

Q.What is the purpose of built-in stricmp() function.
Ans:
It compares two strings by ignoring the case.

Question What is 'recursive function'? Give an example. Exam-2012
Q. Define recursive function with example, state the rules that a recursive function must have to satisfy.
Marks:2.750 Exam-ACCE-2014,2013,ICE-2013,15 CSE-2011,APPE
Ans:
Recursive function:
A function that calls itself is known as recursive function and this technique is known as recursion in C programming
A **recursive function** (DEF) is a **function** which either calls itself or is in a potential cycle of **function** calls.
As the definition specifies, there are two types of recursive functions. Consider a function which calls itself: we call this type of recursion **immediate** recursion.

One can view this mathematically in a directed call graph.

```
A --- |
^     |
|     |
|--- -|
```

```
void A() {
  A();
  return;
}
```
Or.

A recursive function is a function that calls itself during its execution. This enables the function to repeat itself several times, outputting the result and the end of each iteration. Below is an example of a recursive function.

```
function Count (integer N)
   if (N <= 0) return "Must be a Positive Integer";
   if (N > 9) return "Counting Completed";
   else return Count (N+1);
end function
```

Question Write a recursive function in C programming that will return the sum of the series 1+2+3+...+n. Here n is a positive integer. Exam-2014
Answer:

```
#include <stdio.h>
int addNumbers(int n);

int main()
{
   int num;
   printf("Enter a positive integer: ");
   scanf("%d", &num);
   printf("Sum = %d",addNumbers(num));
   return 0;
}

int addNumbers(int n)
{
   if(n != 0)
      return n + addNumbers(n-1);
   else
      return n;
}
```

For better visualization of recursion in this example series sum 1+2+3+4+5:

```
sum(5)
=5+sum(4)
=5+4+sum(3)
=5+4+3+sum(2)
=5+4+3+2+sum(1)
=5+4+3+2+1+sum(0)
=5+4+3+2+1+0
=5+4+3+2+1
=5+4+3+3
=5+4+6
=5+10
=15
```

Rules /condition of Recursive function:

- Every recursive function must have two or more paths.
- One or more of these paths must be non-recursive, which corresponds to the absolute definition.
- Every recursive path must modify its parameter, in order that the recursive call is closer to not recursing than the current call.
- These paths are usually controlled by If statements.

The Factorial Example

The recursive factorial function is quite simple:

```
int fact(int n) {
if(n < 2)
return 1;
return fact(n-1) * n;
}
```

Question Write a function in C that will take two integers as argument and return the largest value. Exam-2012

Answer:

```
include <stdio.h>

/* function declaration */
int max(int num1, int num2);

int main () {
  /* local variable definition */
  int a = 100;
  int b = 200;
  int ret;
   /* calling a function to get max value */
  ret = max(a, b);
   printf( "Max value is : %d\n", ret );
   return 0;
}

/* function returning the max between two numbers */
int max(int num1, int num2) {

  /* local variable declaration */
  int result;
   if (num1 > num2)
     result = num1;
  else
     result = num2;

   return result;
}
```

Question: Suppose an array is passed to a function as an argument .if the value of an array element is altered within the function, will this change be recognized within the calling portion of the program. Exam-2013

Answer:

No

```
#include <stdio.h>
#define MAX 10

int read_intaray(int scores[], int lim);
void print_intaray(int scores[], int lim);
main()
{   int n, exam_scores[MAX];

    printf("***List of Exam Scores***\n\n");
    n = read_intaray(exam_scores, MAX);
    print_intaray(exam_scores, n);
}

/* Function reads scores in an array. */
int read_intaray(int scores[], int lim)
{   int n, count = 0;

    printf("Type scores, EOF to quit\n");

    while ((count < lim) && (scanf("%d", &n) != EOF)) {
         scores[count] = n;
         count++;
    }
    return count;
}

/* Function prints lim elements in the array scores. */
void print_intaray(int scores[], int lim)
{   int i;

    printf("\n***Exam Scores***\n\n");
    for (i = 0; i < lim; i++)
         printf("%d\n", scores[i]);
}
```

Question What are the difference between passing an array to a function and passing a single-valued item to a function ?Exam-21016

Passing Array	Single value
Its passing address of first element from array	Its pass value

`void myFunction(int param[10]) {`
`  .`
`  .`
`  .`
`}`
	`void myFunction(int param) {`
	`  .`
	`  .`
	`  .`
	`}`

Question : Summarize the rules governing the use of the return statement. Can multiple expressions be included in a return statement/ can multiple return statements be included in a function. Exam-2013

Rules of Return Statement:

- If control reaches the end of a function with the return type void (possibly cv-qualified), end of a constructor, end of a destructor, or the end of a function-try-block for a function with the return type (possibly cv-qualified) void without encountering a return statement, return; is executed.

- If control reaches the end of the main function, return 0; is executed.
- Flowing off the end of a value-returning function (except main) without a return statement is undefined behavior.
- In a function returning void, the return statement with expression can be used, if the expression type is void.
- Returning by value may involve construction and copy/move of a temporary object, unless copy elision is used.

Multiple Return Statements Be Included In A Function:

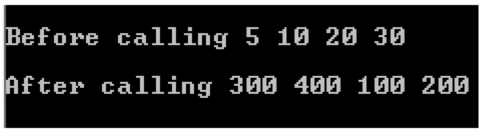
- The first return statement that is executed will terminate the function and its value will be used.
- However, there can obviously be multiple execution paths - and they can return different values.
- Actually in a non-void function every possible execution path has to return something

Question Define command line arguments. Give an example for passing command line argument in a C program. Exam-2014

Question) What will be output of the following code?Exam-2015

```
        #include<stdio.h>
        #include<conio.h>
        void func1(int *p,int *q, int *r, int *s);

      main() {
        int a,b,c,d,*x,*y;
        a=5;b=10;x=&c;y=&d;c=20;d=30;
        printf("\n\nBefore calling %d %d %d %d\n\n",a,b,*x,*y);
        func1(&c,&d,&a,&b);
        printf("After calling %d %d %d %d\n",a,b,*x,*y);
        getch();
      }
      void func1(int *p,int *q, int *r, int *s){
        *p=100; *q=200; *r=300; *s=400;
              }IN pointer Chapter:
```

Answer:



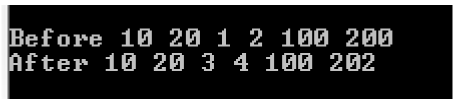
```
Before calling 5 10 20 30

After calling 300 400 100 200
```

Question What will be the output of the following code ? If you think any value displayed may be garbage , mention it as garbage too .

```
    #include<stdio.h>
    #include<conio.h>
   void func1(int a, int p, int q[]);
    int x, y;
   main(){
          int a,b,c[3];
          a=10;b=20;c[1]=1,c[2]=2;
          x=100;y=200;
   printf("\n\nBefore %d %d %d %d %d %d\n",a,b,c[1],c[2],x,y);
   func1(a,b,c);
   printf("After %d %d %d %d %d %d\n",a,b,c[1],c[2],x,y);
   getch();
```

```
}

void func1(int a, int p, int q[]){
  int x;
  a=100;
   p=200;
  q[1]=q[1]+2;
  q[2]=q[2]+2;
  x=102;  y=202;
}
```

Exam-2015
Answer:

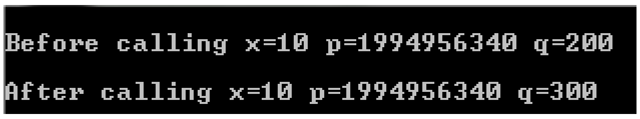


```
Before 10 20 1 2 100 200
After 10 20 3 4 100 202
```

Question Explain how the variables take the values .Exam-2016
(Objective of the question : To check concepts of local and global variables)

```
#include<stdio.h>
#include<conio.h>
void add_int(int n);
int x,p,q;
int main(){
int p;
 q=200;
x=10;
printf("\nBefore calling x=%d p=%d q=%d\n",x,p,q);
add_int(x);
printf("\nAfter calling x=%d p=%d q=%d",x,p,q);
getch();
}
void add_int(int x){
x=50;
p=200;
q=300;
}
```

Answer:



```
Before calling x=10 p=1994956340 q=200

After calling x=10 p=1994956340 q=300
```

Q. Write a recursive function in C programming that will receive an integer n as input and return the nth number of Fibonacci series. Marks:3 Exam-ACCE-2014,ICE-2013
Ans:

```
#include<stdio.h>

int fibonaci(int i){

if(i ==0){
return0;
}
if(i ==1){
return1;
}
return fibonaci(i-1)+ fibonaci(i-2);
}
int  main(){
int i;
for(i =0; i <10; i++){
    printf("%d\t\n", fibonaci(i));
}
return0;
}
```

When the above code is compiled and executed, it produces the following result −

```
0        1        1        2        3        5        8        13        21        34
```

Q.Write a recursive function in C programming that print factorial number?
Ans:

```
#include<stdio.h>
 int fact(int n) {
 if(n < 2)
 return 1;
 return fact(n-1) * n;
 }
int  main(){
int i =15;
  printf("Factorial of %d is %d\n", i, fact (i));
return0;
}
```

Q.Discuss the Advantages and Disadvantages of Recursion?
Ans:
Advantage:
Recursion is more elegant and requires few variables which make program clean. Recursion can be used to replace complex nesting code by dividing the problem into same problem of its sub-type.
Disadvantage:
In other hand, it is hard to think the logic of a recursive function. It is also difficult to debug the code containing recursion.

Q. Describe different categories of function based on their arguments and return type.
Ans:

For better understanding of arguments and return type in functions, user-defined functions can be categorised as:

1. Function with no arguments and no return value
2. Function with no arguments and return value
3. Function with arguments but no return value
4. Function with arguments and return value.

Example of each category:

1.Function with no arguments and no return value

```
#include <stdio.h>
void prime();
int main(){
   prime();     //No argument is passed to prime().
   return 0;
}
void prime(){
                 /* There is no return value to calling function main().
Hence, return type of prime() is void */
   int num,i,flag=0;
   printf("Enter positive integer enter to check:\n");
   scanf("%d",&num);
   for(i=2;i<=num/2;++i){
      if(num%i==0){
          flag=1;
      }
   }
   if (flag==1)
      printf("%d is not prime",num);
   else
     printf("%d is prime",num);
              }
```

Function prime() is used for asking user a input, check for whether it is prime of not and display it accordingly. No argument is passed and returned form prime() function.

2.Function with no arguments but return value

C program to check whether a number entered by user is prime or not using function with no arguments but having return value

```
#include <stdio.h>
int input();
int main(){
   int num,i,flag = 0;
   num=input();    /* No argument is passed to input() */
   for(i=2; i<=num/2; ++i){
   if(num%i==0){
      flag = 1;
      break;
   }
   }
   if(flag == 1)
      printf("%d is not prime",num);
   else
```

```
      printf("%d is prime", num);
   return 0;
}
int input(){      /* Integer value is returned from input() to calling
function */
   int n;
   printf("Enter positive integer to check:\n");
   scanf("%d",&n);
   return n;
}
```

There is no argument passed to input() function But, the value of *n* is returned from input() to main() function.

Function with arguments and no return value

Program to check whether a number entered by user is prime or not using function with arguments and no return value

```
#include <stdio.h>
void check_display(int n);
int main(){
   int num;
   printf("Enter positive enter to check:\n");
   scanf("%d",&num);
   check_display(num);
   return 0;
}
void check_display(int n){
/* There is no return value to calling function. Hence, return
type of function is void. */
   int i, flag = 0;
for(i=2; i<=n/2; ++i){
   if(n%i==0){
      flag = 1;
      break;
   }
   }
   if(flag == 1)
      printf("%d is not prime",n);
   else
      printf("%d is prime", n);
}
```

Here, check_display() function is used for check whether it is prime or not and display it accordingly. Here, argument is passed to user-defined function but, value is not returned from it to calling function.

Function with argument and a return value

Program to check whether a number entered by user is prime or not using function with argument and return value

```
#include <stdio.h>
int check(int n);
int main(){
   int num,num_check=0;
```

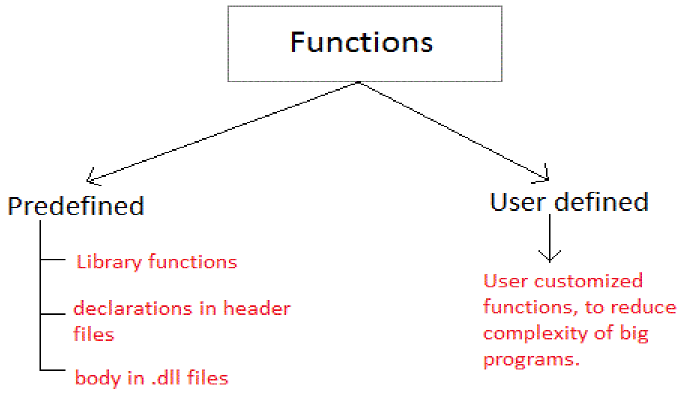
```
    printf("Enter positive enter to check:\n");
    scanf("%d",&num);
    num_check=check(num); /* Argument num is passed to
check() function. */
    if(num_check==1)
      printf("%d is not prime",num);
    else
      printf("%d is prime",num);
    return 0;
}
int check(int n){
/* Integer value is returned from function check() */
    int i;
    for(i=2;i<=n/2;++i){
    if(n%i==0)
        return 1;
}
  return 0;
}
```

Q.What is function? Discuss different type of function?why need user defined function? Exam:ICE-2013ACCE-2011,CSE

A function is a block of code that performs a particular tas.

C functions can be classified into two categories,

- **Library functions**
- **User-defined functions**


Functions
Predefined
Library functions
declarations in header files
body in .dll files
User defined
User customized functions, to reduce complexity of big programs.

Library functions are those functions which are defined by C library, example **printf()**, **scanf()**, **strcat()** etc. You just need to include appropriate header files to use these functions. These are already declared and defined in C libraries.

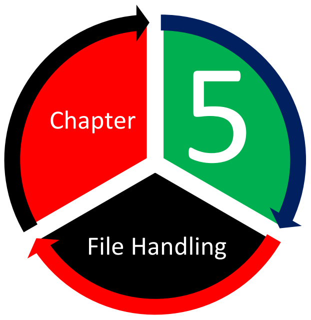
User-defined functions are those functions which are defined by the user at the time of writing program. Functions are made for code reusability and for saving time and space.

Or

The functions which we can create by ourselves, for example in the above code I have created a function abc and I called it in main() in order to use it.

Benefits of Using Functions

1. It provides modularity to the program.
2. Easy code Reuseability. You just have to call the function by its name to use it.
3. In case of large programs with thousands of code lines, debugging and editing becomes easier if you use functions.



Important Question

1. What is difference between file and stream? What are the purpose of the following functions
   (i)fopen() (ii)feof() (iii) fflush()**Exam-2013**
9. Describe different types of file opening mode to open a data file. **Exam-2014**
10. Why do we need to use 'fclose(fp)'?**Exam-2015**
11. Let the contents of file named "data.dat":**Exam-2015**

```
12345
23456
34567
45678
56789
```

12. Write a program to read those values from that file and print the sum of each row in another file named "output.dat" The contents of "output.dat" look like:

```
15
20
25
30
35
```

Question:What is difference between file and stream? What are the purpose of the following functions
(i)fopen() (ii)feof() (iii) fflush()Exam-2013

Stream:
Stream is represention of flow of data from one side to another e.g from disk to memory and from memory to disk.
File:
File is represention to store data on disk file.file use streams to store and load data. File class for typical operations such as copying , moving , renaming , creating , opening , deleting and appending to files and stream is an abstract base class of all streams. A stream is an abstraction of a sequence of bytes. Such as a file an input/output device an inter-process communication pipe or a tcp/ip socket.

(I)Fopen()
The c library function file *fopen (const char *filename , const char *mode). Opens the file name pointed to by filename using the given mood. The fopen() function opens a file whose name is pointed to by fname and returns the stream that is associated with it. The type of operation that will be allowed on the file are defined by the value of mode.

(Ii) Feof()
feof() the c library function int feof (FILE *stream) test the end of file indicator for the given stream. The feof() function is particularly useful when working with binary files. The feof() determines whether the end of the file associated with stream has been reached.

(Iii)Fflush()
The c library function int fflush (File *stream) . If stream is associated with a file opened for writing a call to fflush () cause the contents of the output buffer to be physically written to the file remains open.

Q.Describe different types of file operation in data file?Marks:3 Exam-ACCE-2014
Answer:
A file represents a sequence of bytes, does not matter if it is a text file or binary file. C programming language provides access on high level functions as well as low level (OS level) calls to handle file on your storage devices.
In C programming, file is a place on disk where a group of related data is stored.
File Operations
(i) Creating a new file.
(ii) Writing into a file.
(iii) Opening an existing file.
(iv) Reading a file.
(v) Closing a file.

Opening Files
You can use the fopen( ) function to create a new file or to open an existing file, this call will initialize an object of the type FILE, which contains all the information necessary to control the stream. Following is the prototype of this function call:

```
FILE *fopen("file_name","Mode );
```

for example:

```
FILE  *fp;
fp = fopen("MYABC.C","r");
```

Above code will open a file MYABC.C in r mode (read only mode). The address of the first character is stored in pointer fp.

Writing a File

Following is the simplest function to write individual characters to a stream:

```
int fputc( int c, FILE *fp );
int fputs( const char *s, FILE *fp );
#include <stdio.h>
main()
{
FILE *fp;
}
```

Closing a File

To close a file, use the fclose( ) function.

The prototype of this function is:

int fclose(FILE *fp);

Here fclose() function closes the file and returns **zero** on success, or **EOF** if there is an error in closing the file. This **EOF** is a constant defined in the header file **stdio.h.**

No, it is a structure defined in stdio.h.

Question: Describe different types of file opening mode to open a data file. Exam-2014
Q.Briefly discuss Opening Modes in Standard I/O?
Answer:

File can be **opened in basic 3 modes** : Reading Mode, Writing Mode, Appending Mode

If File is not present on the path specified then **New File can be created using Write and Append Mode.**

File Opening Mode Chart :

Mode	Meaning	fopen Returns if FILE-	
		Exists	Not Exists
r	Reading	–	NULL
w	Writing	Over write on Existing	Create New File
a	Append	–	Create New File
r+	Reading + Writing	New data is written at the beginning overwriting existing data	Create New File
w+	Reading + Writing	Over write on Existing	Create New File
a+	Reading + Appending	New data is appended at the end of file	Create New File

Question: Why do we need to use 'fclose(fp)' ? (Exam -2015)

Need of 'fclose(fp):

```
#include <stdio.h>
int fclose(FILE *stream);
```

The fclose() function closes the file associated with stream and flushes its buffer. After a call to fclose() stream is no longer connected with the file and any automatically allocated buffer are deallocated.
If fclose() is successful , zero is returned; otherwise EOF is returned . Removing the storage media before closing a file will also generate an error , as will lack of sufficient free disk space.

Q.How can we working with file? Given with example? Exam:

Working with file

While working with file, you need to declare a pointer of type file. This declaration is needed for communication between file and program.
FILE *ptr;

Opening a file

Opening a file is performed using library function fopen(). The syntax for opening a file in standard I/O is:
ptr=fopen("fileopen","mode")

For Example:
fopen("E:\\cprogram\program.txt","w");

```
/* ----------------------------------------------------------- */
 E:\\cprogram\program.txt is the location to create file.
 "w" represents the mode for writing.
/* ----------------------------------------------------------- */
```

Q. Why files are needed?
Ans:

When the program is terminated, the entire data is lost in C programming. If you want to keep large volume of data, it is time consuming to enter the entire data. But, if file is created, these information can be accessed using few commands.

There are large numbers of functions to handle file I/O in C language. In this tutorial, you will learn to handle standard I/O(High level file I/O functions) in C.

High level file I/O functions can be categorized as:

1. Text file
2. Binary file

Q. write a c program that will read a text file and count the total number of vowels within the text file. The file only contains uppercase latter? Exam:ACCE-2011

```
#include<stdio.h>
```

```
void main()
{
        FILE *fp;
int vowel=0,consonant=0;
char ch;
fp=fopen("name.txt","r");
if(fp==NULL)
{
printf("Source can't be opened");
exit(0);
}
ch=fgetc(fp);
while(ch!=EOF)
            {               if((ch=='a')||(ch=='A')||(ch=='e')||(ch=='E')||(ch=
='i')||(ch=='I')||(ch=='o')||(ch=='O')||(ch=='u')||(ch=='U'))
{  vowel++;
}
else
{
consonant++;
}   ch=fgetc(fp);
}
printf("\n Number of vowels are = %d",vowel);

return 0;
}
```

Q.Discuss you we can use File Read operation ?
Ans:

```
fp = fopen("test.txt", "w+");
fprintf(fp, "This is testing for fprintf...\n");
fputs("This is testing for fputs...\n", fp);
fclose(fp);
}
```

Q.Discuss fprintf() and fscanf() functions.
Ans:

The functions fprintf() and fscanf() are the file version of printf() and fscanf(). The only difference while using fprintf() and fscanf() is that, the first argument is a pointer to the structure FILE

fprintf() can be used to write data in file.

Writing to a file

```
#include<stdio.h>
int main()
{
int n;
  FILE *fptr;
  fptr=fopen("C:\\program.txt","w");
if(fptr==NULL){
    printf("Error!");
exit(1);
}
  printf("Enter n: ");
  scanf("%d",&n);
  fprintf(fptr,"%d",n);
  fclose(fptr);
return0;
}
```

Reading a File

fscanf() can be used to read data from file.

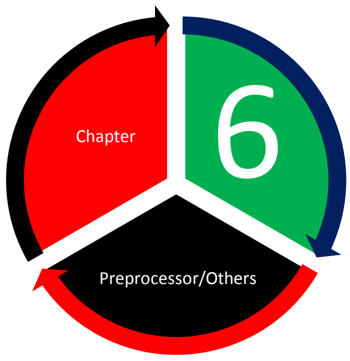
```
#include<stdio.h>
int main()
{
int n;
  FILE *fptr;
if((fptr=fopen("C:\\program.txt","r"))==NULL){
     printf("Error! opening file");
exit(1);/* Program exits if file pointer returns NULL. */
}
  fscanf(fptr,"%d",&n);
  printf("Value of n=%d",n);
  fclose(fptr);
return0;
}
```

Question: Let the contents of file named "data.dat":Exam-2015

```
12345
23456
34567
45678
56789
```

Write a program to read those values from that file and print the sum of each row in another file named "output.dat" The contents of "output.dat" look like:

```
15
20
25
30
35
```



Important Question

1. Write a program to read a **n** bit long binary stringed then search how times pattern '000' occurs . Do not consider same '0' in two adjacent '000' pattern . For example '100001' or '1000001' has only one '000' pattern but '1000000001' has two '000'.
   Sample input: 10100011100100001011000001000001110
   Output: 5

Exam-21016

2. What is the difference between #include<filename> and #include "filename"? **Exam-2013**
3. What is meant by "bit masking"? how do you check the 3rd bit of an integer variable is "1". **Exam-2013**
4. What is a macro, and how do you use it? What is the output of the following code? Explain. **Exam-2013**

```
#define mul(x,y)(x)*y
Int main()
   {printf("%d",mul(3+2,4+5));
```

5. What are the syntax error (if any) of the following code?**Exam-2015**

```
#include <stdio.h>
int 1x,2x,y1,y2;
float z;
char a[10], b[10];
main()
{
Scanf("%d%d%f",y1,z);
Scanf("%c%c%c), &a[1],a[2],&a[3];
b[2]=a[2];
y2=b[2]+a[2]+y1;
printf("%f%f%f%d%d,&y1,z,y2,z,a[3]);
}
```

6. Write C program for the following problem, it doesn't need to think about run time optimization.**Exam-21016**
   (Objective of the question: To check the problem understanding capacity)

One day, one of the students of CSE dept. named Sumon is having a party, and he has invited his friends, p of them have arrived already, but some other are running late. To occupy his guests, Sumon tried playing some team game with them, but he found that it was impossible to divide the p guests into any number of equal sized groups of more than one person. As a result, he had to wait until q guest(s) arrived, q may be single guest or group of guests arrived at this time. Finally, summon could make teams of equal sized from the arrived (p+q) guests and started his games .
Sample Input
The input will consist of 5 test cases. Each test case will be given as a non-negative integer (p+q) and (p+q)<=50;
Sample Output
For each test case, output will be an integer q that is closest to (p+q)..

Sample Input	Corresponding Output
8	1
22	3

7. Write a C program that will define a structure time which contains three integer variables hours, minutes and seconds where 0<=hours<=23 minutes, seconds<=59. Then the program will read the beginning and ending time of an event from input console and pass them to a function that compute and print the duration of the event. For example , if an event begins at 21:55:34 and ends at 03:25:21 then the duration of the event is 05 hours 29 minutes and 47 seconds. Consider that the maximum duration of an event is less than 24hours. **Exam-2014**

Question: Write a program to read a n bit long binary stringed then search how times pattern '000' occurs . Do not consider same '0' in two adjacent '000' pattern . For example '100001' or '1000001' has only one '000' pattern but '1000000001' has two '000'.

Sample input: 10100011100100001011000001000001110

Output: 5

Exam-21016

Question:What is the difference between #include<filename> and #include "filename"? Exam-2013

Question:What is meant by "bit masking"? how do you check the 3rd bit of an integer variable is "1". Exam-2013

Question:What is a macro, and how do you use it? What is the output of the following code? Explain. Exam-2013

```
#define mul(x,y)(x)*y
Int main()
  {printf("%d",mul(3+2,4+5));
```

Question:What are the syntax error (if any) of the following code?Exam-2015

```
#include <stdio.h>
int 1x,2x,y1,y2;
float z;
char a[10], b[10];
main()
{
Scanf("%d%d%f",y1,z);
Scanf("%c%c%c), &a[1],a[2],&a[3];
b[2]=a[2];
y2=b[2]+a[2]+y1;
printf("%f%f%f%d%d,&y1,z,y2,z,a[3]);
}
```

Question:Write C program for the following problem, it doesn't need to think about run time optimization.Exam-21016

(Objective of the question: To check the problem understanding capacity)

One day, one of the students of CSE dept. named Sumon is having a party, and he has invited his friends, p of them have arrived already, but some other are running late. To occupy his guests, Sumon tried playing some team game with them, but he found that it was impossible to divide the p guests into any number of equal sized groups of more than one person. As a result, he had to wait until q guest(s) arrived, q may be single guest or group of guests arrived at this time. Finally, summon could make teams of equal sized from the arrived (p+q) guests and started his games .

Sample Input

The input will consist of 5 test cases. Each test case will be given as a non-negative integer (p+q) and (p+q)<=50;

Sample Output

For each test case, output will be an integer q that is closest to (p+q)..

Sample Input	Corresponding Output
8	1
22	3

1. **Write a C program that will define a structure time which contains three integer variables hours, minutes and seconds where 0<=hours<=23 minutes, seconds<=59. Then the program will read the beginning and ending time of an event from input console and pass them to a function that compute and print the duration of the event. For example , if an event begins at 21:55:34 and ends at 03:25:21 then the duration of the event is 05 hours 29 minutes and 47 seconds. Consider that the maximum duration of an event is less than 24hours. Exam-2014**

Q.Discuss about the preprocessor?
Ans:

Preprocessor directives are actually the instructions to the compiler itself. They are not translated but are operated directly by the compiler. Note that preprocessor statements begin with a #symbol, and are NOT terminated by a semicolon. Traditionally, preprocessor statements are listed at the beginning of the source file.They are also termed as macros.

The most common preprocessor directives are

- include directive
- define directive

1.include directive:

The include directive is used to include files like as we include header files in the beginning of the program using #include directive like
#include<stdio.h>
#include<conio.h>

2.define directive:

It is used to assign names to different constants or statements which are to be used repeatedly in a program. These defined values or statement can be used by main or in the user defined functions as well. They are used for
a. defining a constant
b. defining a statement
c. defining a mathematical expression

For example
#define PI 3.141593
#define TRUE 1
#define floatingpointno float

Q. Write a c function to evaluate the series or sine series? Exam:ACCE-2013

$$\sin(x) = x - \left(\frac{x^3}{3}\right) + \frac{x^5}{5} - \left(\frac{x^7}{7}\right)$$

Ans:

```
include<stdio.h>
#include<math.h>
#define ACC 0.000001
#define PI 3.14
Int  main()
{       int x,i;
        float x1,x2,sine=0,term;
        printf("Enter the X value : ");
        scanf("%d",&x);
        x1=(x*PI)/180;
        x2=x1*x1;
        term=x1;
        sine=term;
        for(i=3;fabs(term)>ACC;i=i+2)
           {
                term =(x2*(-term))/(i*(i-1));
                sine = sine+term;
           }
                printf("\n%f",sine);
return 0;
}
Output:
Enter the X value :100
0.984961
```

About the Authors

Abu Saleh Musa Miah(abid) was born in Rangpur,Bangladesh in 1992. He received the B.Sc. Engineering degree with Honours and M.Sc. Engineering degree in Computer science and Engineering Department in 2014 and 2015 with FIRST merit Position Engineering first batch from University of Rajshahi ,Bangladesh. He also completed research on the field **Brain Computer Interface(BCI),COMPUTER VISION** specifically Moving Object Detection with BoofCV tools(java). Currently he is working towards the M.Phil. Department of Computer Science And Engineering University of Rajshahi, Bangladesh.

His research interests include **Brain Computer Interfacing.** Sparse signal recovery/compressed sensing, blind source separation, neuroimaging and computational and cognitive neuroscience.

Email: abusalehcse.ru@gmail.com;

https://www.facebook.com/abusalehmusa.miah