Q1. Write an essay covering the history and evolution of c programming. Explain its importance and why it is still used today.

Ans : History of C programming language :

- C language is a procedural programming language(POP).
- C is one type of programming language.
- It was developed by Dennis.M.Ritchie in the year 1972 in USA.
- It was developed at "AT & T's Bell Laboratories".
- Dennis.M.Ritchie was the founder of father of C language.

➢ **Importance note of c programming :**

- C is a case sensitive language.
- C is an extension of c language file.
- C language supports 32 keywords.
- C language is a middle level language.

➢ **Importance of c programming language :**

- C language is a open source programming language.
- C language is easy to learn and understand.
- C language is portable (user easily run same code in any different windows).
- Easy debugging in c language.

➢ **Why it is still used today :**

- C language is a oldest , fastest , famous and important programming language . C language is use to games , Browser development and software application.
- C language is a very flexible , portable , open source and middle level (part of high level language and low level langauge) so c language is very used in today generation.

Q2 : Describe the Steps to install a C compiler in Dev C++.

Ans :

Step 1 : go to your google and search download dev C++(version 5.11.0) or visual studio .

Step 2 : Download opera in your system and click to Download button for Dev C++.

Step 3 : After download you install Dev C++ in your windows , linux .

Step 4 : Dev C++ complete install in your system to set your fonts , theme and size etc.

Step 5 : after installation open Dev c++ and create a new file and write your code.

Step 6 : you save your file any place and save the extension in .cand run program.

Q3 : Explain the basic structure of a C programming , including headers , main function , comments , data types , and variables . provide example.

Ans : Basic structure in c:

```
#include<stdio.h>

Main()

{

        Block of code

}
```

#include<stdio.h>

: preprocessor

include : keyword by C

<stdio.h> : standard input output (header file)
main() : () function
execution of the program will be started from here.

{ } - block of code
< > - use to include any library
[ ] – use to store index number of string or character
( ) – use to function in any method to start a code

- **Comments :**

the non-executable part of the program is known as a comments.

comments is part of program but not display output.

Comment is use to show a hint to what perform your code and what you perform in this code you easily know.

There are two types of comment :

1. single line comment

2. multi line comment

Single line comment is start with // and use to notes about single specific line.

Multi line comment start with /* and end with */ and use to multiple explanation and large block of code.

- **Data type:**

the data type of any variable tells that what kind of values will be stored inside the variable.

In simple words data type means store a data in variable.

Data type is two type :

1.primitive datatype.

2.nonprimitive datatype.

- **Primitive datatype :**

primitive datatype is provide ianguage.

Ex : int , float , char etc…

Integer : this data type is used to store a integer value and size of bytes is 2 bytes.

%d use to store the integer value in code.

Float : this data type is used to store decimal value and size of bytes is 4 bytes.

%f is used to store the floating value in code.

Char : This data type is used to store a single character .

%c use to store the character in code and use to ‘’.

Long integer : declare %ld store the positive value high range.

double : declare %lf store the decimal value high range.

- **Non primitive datatype :**

nonprimitive datatype is provide developer.

Ex : string , array , structure etc…

String : string means collection of elements. Use to store multiple character .

%s use store the string value in code.

- **Variable :**

element(memory) to store the particule value .

Ex : a=10

a=123456789 , a=34.56 , a=345678.7656 , a=’g’ , a=”tops”

Example :

include<stdio.h>

Main()

```
{

Int a=30;

Float b=20.5;

Printf("\n store the value: %d",a);

Printf("\n store the value: %f",b);

}
```

Q4 .Write notes explaining each type of operator in C:

In c programming operator means to perform some operations on the data or values.

Many types of operator in c language :

1. arithmetic
2. relational
3. logical
4. assignment
5. increment/decrement
6. bitwise
7. conditional operators

1. Arithmetic operator : The arithmetic operator are used perform the some operations on the value.

Operators	meaning
+	Addition
-	substraction
*	multiplication
/	division
%	moduls

2.Relational operator : Relational operator is also known as a comparison operator.

Operators	meaning
>	Greater than
<	Less than
>=	Greater than & Equal to
<=	Less than & Equal to
==	Equal to
!=	Not equal to

3.logical operator :

operator	meaning
&&	Logical AND
||	Logical OR
!	Logical NOT

&& - And (All the expressions must be true)
|| - Or (One of the any condition must be true)
! - Not (true expression will turn into false)

4.Assignment operator : The assignement operator is used to assign the value to the variable.

+=	a=a+b	a+=b
-=	a=a-b	a-=b
*=	a=a*b	a+=b
/=	a=a/b	a+=b
%=	a=a%b	a+=b

5.Increment / Decrement operator :

operator	meaning
++a	Prefix increment
a++	Postfix increment
--a	Prefix increment
a--	Postfix increment

Prefix : ++i, --i
Postfix : i++, i—

unary op. - a++, b-- (1 operand, 1 operator)
binary op. - a+b (2 operands, 1 operator)

Q6 :Explain decision-making statements in C (if, else, nested if-else, switch). Provide examples of each.

1. If stetment : The condition is true code to be executed.

Syntax :

```
  if(condition)
{
        //code to b execute
}
```

2.if_else stetment : IF statement condition is false so else statement condition is true.

Syntax :

```
if(condition)
        {
         //code to be execute
        }
        Else
        {
        // code to be execute
        }
```

3.nested if stetment : check if condition within if condition (if into if)

Syntax :

```
if(condition)
{
        If(condition)
        {
        //stetment
        }
        else
        {
        //stetment
        }
Else
        {
        //stetment
        }
}
```

4.switch stetment : multiple choice value and one choice use switch stetment.

Not use relation operator , switch use int , charcter datatype , switch use keyword switch , break , case , default.

Syntax :

```
switch(choice)
```

```
{
        Case 1:  // stetment
        Break;
        Case 2 : //stetment
        Break;
        Default : //stetment
        Break;
}
```

Q6.Compare and contrast while loops, for loops, and do-while loops. Explain the scenarios in which each loop is most appropriate.

ANS : in c language loops is repeat the same code a number of times.
in c language two types of loops :

1)entry loop
2)exit loop

1)entry loop : 1)while loop
2)for loop
2)exit loop : 1)do..while loop

1)while loop : A while loop is the an entry controlled loop. in while loop given condition is true then the loop is executed.and given condition false the loop is not executed.

Syntax of while loop :

```
intialization
While(condition)
{
        //block of code
                Increment / decrement
}
```

Example :

```
i = 1;
While(i<=10)
{
printf (“%d”, i);
I++;
}
```

2) for loop : for loop is easier to compare than while loop. A for loop is the an entry controlled loop. in for loop given condition is true then the loop is executed and given condition false the loop is not executed.

Syntax :

```
For(initialization ; condition ; increment/decrement)
{
        //block of code;
}
```

Example:

```
For(i=1;i<=10;i++)
{
Printf(“ %d”,i);
}
```

3)do-while loop : do-while loop is a exit control loop . in do – while loop given condition is true d0-while loop first time execute and after loop is repeat given condition is false loop is not execute.

Syntax :

```
initialization
do
{
        //block of code
        Increment/decrement

}while(condition)
```

Q7. Explain the use of break, continue, and goto statements in C. Provide examples of each.

ANS : The break statement :

- The break statement is mainly used in the switch statements .it is also useful for immediately stopping a loop.
- IN simple words break statement is used to break the code and rest of the code will not be executed.

Example of break statement:

```
#include<stdio.h>
main()
{
    int i;
    for(i=5;i>0;i--)
   {
          if(i==3)
          break;
          printf("%d\n",i);
    }
}
```

O/P : 5 4

2) The continue statement :

- When you skip the current iteration but remain in the loop you should use the continue statement .
- IN simple words you skip the current iteration and rest of the code will not be executed then.

Example of Continue statement :

```
#include<stdio.h>
main()
{
   int i;
   for(i=5;i>0;i--)
  {
          if(i==3)
          continue;
          printf("%d\n",i);
  }
}
```

O/P : 5 4 2 1

3)The go to statement :

- The go to statement used to create a label and this label to go the statement and continue the code will be executed.

Example of goto statement :

```
#include<stdio.h>
main()
{
        int  i;
        i=1;
label:

        printf("%d\n",i);
        i++;
        if(i<=5)
                goto  label;
}
```

O/P : 1 2 3 4 5

Q8. What are functions in C? Explain function declaration, definition, and how to call a function. Provide examples.

ANS : A function is a set of statements or group of block of code that performs a specific task.

- Every c Program has at least one function , which is main().
- In function you have not return any value from the Function to use void () function.

IN c language Two types of function :

1) In built function

2) User Defined Function

1) In built function : In built function is provided by system means already all rules and function defined by in c compiler.

2) User Defined Function : User Defined function is used to Reusability of the code.

- User defined functions , the user give any name to the functions except the name of key words.

- **Types of User Defined Function :**

1) function without argument without return value
2) Function with argument without return value
3) function without argument with return value
4) function with argument with return value

- **User defined functions mainly three parts follows :**

1) function Declaration (After Header file)
2) Function calling (in main function)
3) Function Definition (After main Function or Function body)

Example of Function :

```
#include<stdio.h>
void num();                 //function declaration
main()
{
        num();              //function calling
}

void num()                  //function definiton
{
        int i;
        for(i=0;i<=10;i++)
        {
                printf("%d\n",i);
        }
}
```

Q9. Explain the concept of arrays in C. Differentiate between one-dimensional and multi-dimensional arrays with examples.

ANS : An array is a Collection of elements or values with similar data types.

- syntax : data_type array_name[size of array];
- Each elements refers to the identification number called index number.
- Always array index will be started from "0".
- The array is the simplest data structure where each data element can be randomly accessed by using its index number.

- Mainly three Types of Arrays :

1) One Dimentional Array e.g int arr[5];
2) Two Dimentional Array e.g int arr[3][3];
3) Multi Dimentional Array e. g int arr[4][3][3];

Differentiate between one-dimensional and multi-dimensional arrays:

One dimensional :it has only one dimensional. Array store single line multiple element. One dimensional array isexecuteseries type.

Multi dimensional: it has three or multiple dimensional. Multi dimensional array store row and column multiple element. And multi dimensional array is execute table or matrix.

Example:

```
int array[50],i,n;
        printf("enter user value");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
        printf("enter the value: [%d]",i);
        scanf("\n%d",&array[i]);
        }
        for(i=0;i<n;i++)
        {
        printf("\n array[%d] %d",i,array[i]); }
```

Q10. Explain what pointers are in C and how they are declared and initialized. Why are pointers important in C?

ANS :

- In c programming Pointer variable can store the address of the variable.
- Pointer used to point the actual value.
- Pointer is a variable that stores the memory address of an other variable.

How pointer declared and initialized :
In c language pointer declared to (*)asterisk symbol.
Syntax :
Data_type *ponter_name;

```
int num = 10;
int *ptr = &num;
printf("%d",*ptr);
```

pointers are important in c because they allow the programmer to directly manipulate the computer's memory , which can improve performance and reduce code. And follow this purpose :
1) memory management
2) pass by reference
3) working with array and string
4) store the address in variable

Q11. Explain string handling functions like strlen(), strcpy(), strcat(), strcmp(), and strchr(). Provide examples of when these functions are useful.

ANS :

- **Strlen () :** strlen() function is a built-in function that returns the length of the string and it doesn't count the null character.
- **Strcpy() :** strcpy() function is a built-in function that copies a string from one location to another Or copies the contents from source string to destination string.
- **Strcat() :** strcat() function is concats or joined first string with second string.
- **Strcmp() :** strcmp() is a built-in functions that is used to compare two strings and both strings are equal it returns 0.
- **Strchr() :**

All String handling function example in one code :

```
#include<stdio.h>
main()
{
        char  str1[20] , str2[20] , str3[40];
        printf("\n\n\t Enter a string1 = ");
        gets(str1);
        printf("\n\n\t Enter a string2 = ");
        gets(str2);

        if(stricmp(str1 , str2)==0)
        {
                printf("\n\n\t strings are same.");
        }
        else
        {
                printf("Strings are not same.");
        }

        strcpy(str3,strcat(str1,str2));

        printf("\n\n\t concat of string3 = %s",str3);
        printf("\n\n\t string2 %s",strcpy(str3,str1));
        printf("\n\n\t Lenghth of the string = %d",strlen(str3));
}
```

Q12 . Explain the concept of structures in C. Describe how to declare, initialize, and access structure members.

ANS : Structure is a template or blueprint which is a collection of elements with different types of data.

Structure is user defined data type in c programming language.

In c programming use to structute for define struct keyword.

In Structure you create a nested structure .

Declaration of structure :

```
Struct structure_name {                    //Structure name
        Datatype Member1_name;
        Datatype Member2_name;             //structure member name
}s;
```

Initialize of structure :

```
Struct Student_info{
        Int rollno;
        Char name[20];
}s;
```

Access Structure members:

```
{
        s.rollno;
        s.name;
}
```

Q13.Explain the importance of file handling in C. Discuss how to perform file operations like opening, closing, reading, and writing files.

ANS :

File handling : file handling in c is the process in which we create , open , read , write , and close operation on a file.

- C language provides different function such as fopen() , fgets() , fputs().
- **Fopen() =** to open the file in memory by different different modes.
- **Fputs() =** to write data into file.
- **Fgets() =** to read the data from the file.
- **Fclose()=** close a file.

- **Modes in file handling :**

1) ‘r’ : the file is open un read mode.
2)’w’ : creats a text file in write mode.
3)’a’ :open a file append mode.

1) opening a file :

- The fopen() function Is used to create a file or open an existing file.
- Fptr=fopen(“example.txr”,”w”);

2) reading a file:

- The “r” mode is used to read a file.
- To use to fgets() function to read a data in display.
- Fptr=fopen(“example.txt”,”r”);

3) writing file :

- The “w” mode is used to write a file.
- To use to fputs() function to write data in file.
- fptr=fopen("File1.txt","w");

4)close file :

- Used to close a file.
- Fclose(fptr);