International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 04 | April -2020 ISSN: 2582-3930

ADDING ADVANCE FUNCTIONS THROUGH A HEADER FILE IN C PROGRAMMING LANGUAGE

Asst. Prof Amanjot Kaur ,Rishabh Govind Rao ,Sheetala Prasad Mishra

Department of Computer Science Engineering,

MIMIT Malout

Abstract

In the recent years it has been noticed that the C language is best suited for the problem solving aspect and since the development of this language it has very concise and fixed pre-defined function and keeping this concern in mind. In this paper some extras much needed functions are proposed to be added in C language in a particular header file named as "advanceC.h". So our focus is to extend the functionality of the C language. Initially we will introduce the functions we are willing to add and after that we will show the methodology and uses of these functions for the desired task.

Key Words: C Language, Pre-defined functions, header file.

1.INTRODUCTION

There are a number of programming languages and still many new programming languages is being developed day by day, so when one wants to learn new programming language then he/she encounters with a lot of confusions. And as we know C language is the basic fundamental language due to better interaction with the hardware of the system. So C language is always best choice to be , the first language for learning because although in recent years a lot of new languages have been developed but still according to the survey of the (TIOBE Index January 2020), C language is best choice for the problem solving and logic based coding. But noticeable thing is that since the development of the C language the inbuilt function in C are still same and limited in the aspect of the numbers. So in this paper, our main focus is to extend the functionality of the C language. Hence, some extras much needed functions are proposed to be added in C language in a particular header file named as "advanceC.h". And the functions are named as "ap ()", "gp ()", "root ()" and "arr_rev()" and they are used for the sum of arithmetic progression, sum of the geometric progression, to find the nth root of any number in integer form and the reversal of a given array elements respectively. These functions are build considering their need in problem solving area to provide ease of code for the desired task. And if we go towards the practical implementation of our task, so we will write the definition of these functions then save it within a user built header file in our case named as "advanceC.h" and then we can call these function whenever we need by including the header file "advanceC.h".

2.PURPOSE

The main purpose behind adding on these functions in the C programming language is to provide some extended functionality to the C language and as well as to provide the ease of the code from a programmer prospective for some of the desired task related to the these functions individually. And if we go through the purpose

behind each of these functions individually we will get to know

- ap () : - Suppose we are given a task to find the sum of a series of numbers that are in a particular progression called arithmetic progression then if we solved this problem through the traditional method then it will consist a lot of time and effort while if we use this proposed function called "ap ()" our desired task will be performed within small time and less effort.

- gp () : - Similarly, suppose we are given the task to find the sum of the series of the numbers that are in the geometrical progression then we can use the proposed function called "gp ()" to save our time and effort as well.

- root () : - We know that we have function called sqrt () to find the square root of any number but here we are going to introduce a function called root and and it can be used to find not only the square root, cube root but also to find the any number of root of any number in integer form.

- arr_rev () : - Although reversal of an array is so common and often used in coding but there is not a single pre-defined function for it so to save time and to provide ease of the code we are proposed to add the function called "arr_rev ()" to reverse a series of the numbers.

3.METHODOLOGY

When it comes to methodology we are supposed to provide the complete detail of practical implementation of the task we have introduced earlier, so for this part we will use some steps to provide the details of the implementation of our task means how we have performed it practically.

Step 1: -To decide the functions which are to be added:
The very first step from the practical point of view is to decide the functions which will be added to extend the required functionality in the C language. So here we have decided to add these four functions "ap ()", "gp ()", "root ()" and "arr_rev ()" according to their individual need as ap () is used to find the sum of the arithmetic progression, gp () is used to find the sum of the geometrical progression and root () is used to find the nth root of any given integer number in integer form and arr_rev () is used to find the reversal of any given set of numbers.

Step 2:-To write the definition of these functions:
After deciding the functions and their need, the next step is to write the definition of these function, so that later it can be added to a targeted header

files. So the definition of these functions are given below one by one.

- **ap () : -**

```
int ap(int a1,int a2,int a3)
{
int l;
if((a3-a2)==(a2-a1))
{
printf("Enter the last term of the AP:");
scanf("%d",&l);
int d=a2-a1;
int n=(float)((l-a1)/d)+1;
int sum=(float)n/2*(l+a1);
return sum;
 }
else
{
printf("Not in AP:");
return 0;
}
```

- **gp () : -**

```
int gp(int a1,int a2,int a3)
 {
int l;
if((float)a2/a1==(float)a3/a2)
 {
printf("Enter the last term of the GP:");
scanf("%d",&l);
int r=a2/a1;
int res=(l/a1);
inti,count=1,check=1;
for(i=1;i<res;i++)
  {
check=check*r;
count++;
if(check==res)
break;
   }
int num=a1*((power(r,count))-1);
   r=r-1;
int sum=num/r;
return sum;
 }
else
 {
printf("not in gp:");
return 0;
  }
  }
int power(int n,int r)
{
inti,temp=1;
for(i=0;i<r;i++)
temp=temp*n;
return temp;
}
```

- **arr_rev (): -**

```
#include<stdarg.h>
void arr_rev(int num,...)
{
va_listvalist;
int a[num];
inti;
va_start(valist, num);
for (i = 0; i< num; i++)
 {
a[i]= va_arg(valist, int);
 }
va_end(valist);
int start=0;int end=num-1;
int temp;
while (start < end)
  {
temp = a[start];
a[start] = a[end];
a[end] = temp;
start++;
end--;
```

```
  }
for(i=0;i<num;i++)
 {
printf("%d,",a[i]);
 }
}
```

- **root () : -**

```
int root(int n,int m)
 {
inti,flag,j;
for(i=1;i<n;i++)
      {
int k=1;
flag=0;
for(j=0;j<m;j++)
  k=k*i;
if(k==n)
 {
printf("%d",i);
flag=1;
break;
  }
   }
if(flag==0)
printf("Not an integer value");
}
```

Step 3: - To add the definition of these functions into a user built header files:

This is the step where these functions are being added into a user built header file named as "advanceC.h" and for doing this task we have added the definition of every functions into this header file, and after doing this whenever we need the functionalities of these functions we can use them by adding the header file "advanceC.h" using pre-processor #include as follows:

#include"advanceC.h"

Step 4: - To use the functionalities of these functions:

This is the final part of our practical work and here we will test the functionalities of every of the given functions by suitable syntax and example so we can this is the part where we are going to get the result of our task which we have performed earlier. So a separate section called "syntax and example" is introduced here to test these functions.

4.SYNTAX and EXAMPLES

Now we are going to test each of the given functions one by one by using the syntax of these functions and with the suitable examples.

➢ ap () :-

Syntax:-int ap (int a1,int a2,int a3);

Example: - #include<stdio.h>

#include"advanceC.h"

```
void main()
 {
ap (2,4,6);
 }
```

On running the code the consoler will check that the given numbers are in arithmetic progression (ap) or not, and if the numbers are not in ap the consoler will display that "Not in ap" and if the numbers are in ap as shown here then the consoler will ask for the:

"Enter the last term of the ap:" 10(let it be)

Output: - 30.

➢ gp () : -

Syntax: - int gp (int a1,int a2,int a3);

Example: - #include<stdio.h>

#include"advanceC.h"

void main()

{

gp(3,9,27);

}

On running the code the consoler will check that the given numbers are in geometric progression (gp) or not, and if the numbers are not in gp the consoler will display that "Not in gp" and if the numbers are in gp as shown here then the consoler will ask for the:

"Enter the last term of the gp:" 243(let it be)

Output: - 363.

- **arr_rev () : -**

Syntax:-void arr_rev(int num,...);

Example: - #include<stdio.h>

#include"advanceC.h"

#include<stdarg.h>

void main()

{

arr_rev(5,10,20,30,40,50);

}

Here the first argument "5" decides the number of element of the array.
On running the code we will get the output as the reversal of the array which is shown below:

Output: - 50,40,30,20,10.

- **root () : -**

Syntax: -int root (int, int);

Example: - #include<stdio.h>

#include"advanceC.h"

void main()

{

root(1024,10);

}

On running the code output will be 10^{th} root of 1024

Output: - 2.

5.CONCLUSIONS

The main reason would be implementing our own header files and used in c language to make it more efficient. This reduce the dependencies so that code that uses the header doesn't necessarily need to know all the details. This is also reduce the compilation times and also the amount of recompilation needed when something in the implementation changes.

REFRENCES

[1] http://stackoverflow.com/questions/7109964/creatingyour-own-header-file-in-c

[2] http://www.programmingspark.com/2012/12/createyour-own-header-file-in-c.html

[3] http://c-programming-language tut.blogspot. in/2011/12/how-to-create-header-file-in-c-language.html>

[4] http://www.tutorialspoint.com/cprogramming/c_ header_files.html

[5] http://gcc.gnu.org/onlinedocs/cpp/Header-Files.html>

[6] https://practice.geeksforgeeks.org/problems/header-files-in-c

[7] https://en.wikipedia.org/wiki/C_standard_library