UNIT:3

Functions

Topics To be Covered…

3.1 Basics of Functions

3.2 Built-in and user defined Functions

3.3 Using String, Math and other built-in functions

3.4 Working with Functions

3.5 The Return Statement

3.6 Call by Value and Call by Reference

3.7 Types of User-define Functions in C

3.8 Advantages of using Functions

3.9 Recursion

3.10 Advantages and Disadvantages of Recursion

3.1 Basics of Functions

- The function is a collection of statement which combine to perform a particular task

- Every C program has atleast one function which is main function
- The program is divided into number of functions and each function perform a specific task

3.2 Built-in and user defined functions

Functions are of two type:-

1) Built-in Function / Predefined Function / Library function
2) User Defined Functions

1). Library function

- It is pre-defined function or built-in function which is defined in C Library. For example – printf() , scanf() , clrscr() , etc... are example of Library Function

- For that Library Function you have to use appropriate headerfile to use the function

2). User-defined functions

- It is defined by the user at the time of writing the program. There is no need to include the specific headerfile for user defined function

- It is used for code reusability and for saving the time and space

3.3 Using String, Math and other built-in functions

- **String Function :**

There are many function used in string the function are mentioned below:

1. strlen( ) : It calculate the length of the string.

Syntax : variable_name=strlen(string_name);

2. strrev( ) : This function is used to reverse the string.

Syntax : strrev(stringname);

3. strcpy( ) : This function is used to copy one string into another string.

Syntax : strcpy(destination,source);

4. strcat( ) : This function is used to concatenate two string or This function is used to append(join) the value of ine string at the end of another string.

Syntax : strcat(str1,str2);

5. strcmp( ) : This function is used to compare two string. It performs case sensitive comparison. This function returns an integer number if it returns zero then both the string are same otherwise both the string are different.

Syntax : n=strcmp(str1,str2);

6. strlwr( ) : This function is used to convert the given string into lower case. It require only 1 argument.

Syntax : strlwr(str);

7. strupr( ) : This function is used to convert the given string into upper case.It require only 1 argument.

Syntax : strupr(str);

- **Character Function :**

1. **isdigit( )** :- It returns non-zero if argument is 0 to 9.

2. **isalpha( )** :- It returns non-zero if argument is letter or alphabet.

3. **isalnum( )** :- It returns non-zero if argument is digit or letter.

4. **islower( )** :- It returns non-zer0 if argument is lower case letter.

5. **isupper( )** :- It returns non-zero if argument is upper case letter.

- **Mathematics Function :**

- The mathematics function is used to calculate the numerical or mathematical operations.
- Like : find a square root of a number , find power of any number , etc…
- For using this function we have to include <math.h> header file.
- We also find trignomatric calculation by using sin( ), cos( ), etc… function.

3.4 User defined function

1. Function Declaration:

- The Function Declaration is declared behind the headerfile and above the main function
- **Syntax :**

  Return_type function_name(argument(s));
- For example: **void sum(int,int);**
- In this example sum is a funtion which is User Defined Function
- This function has two integer argument x and y and the function type of this function is no return type which is declared by “Void”

2. Function Definition:

- The syntax of function definition is :

```
return_type function_name(argument list)
{
    body of the function;
}
```

- The function definition is declared after the program. Here the return type function name and argument list is same as function declaration . Here the function definition is divided into two parts:

i. Function header
ii. Function body

3. Function call:

- The function call is declared inside the main function and the syntax of function call is:

function_name(argument list);

- **W.A.P to find square of given integer number using UDF.**

```
#include<stdio.h>
#include<conio.h>
int sqrt(int x);
void main()
{
        int m,n;
        printf (“Enter m=“);
        scanf (“%d”,&m);
        n=square(m);
        printf (“The square is %d”,n);
        getch();
}
```

```
int square(int x)
{
    int p;
    p=x*x;
    return(p);
}
```

3.5 The Return Statement

- The syntax of Return Statement is:

```
{
    body of function
    return variable_name;
}
```

- **For Example:**

```
int max(int num1,int num2)
{
  int result;
  if (num1>num2>
    result=num1;
  else
     result=num2;
  return result;
 }
```

3.6 Call by Value and Call by Reference

- There are TWO ways that a C function can be called from a program. They are,

1. Call by value
2. Call by reference

1. Call by value

- In call by value method , the value of the variable is passed to the function as parameter or argument
- The value of the actual argument can not be modified by formal argument of dummy argument
- The actual argument is call use in function call and the formal argument is use in function definition
- In call by value method if there is change in dummy argument then it can not affect in actual argument

- **W.A.P to swap two integer value using call by value method.**

```
#include<stdio.h>
#include<conio.h>
    void swap(int a, int b);
    void main()
{
    int m=22,n=44;
    printf (“Before swaping the value m=%d and n=%d”,m,n);
    swap(m,n);
     getch();
}
```

```
    void swap(int a, int b)
{
    int tmp;
    tmp=a;
    a=b;
    b=tmp;
   printf ("After swapping the value m=%d and
n=%d",a,b);
}
```

2. Call by reference

- In call by reference method, the address of the variable is passed to the function as a argument
- The value of actual argument can be modified by formal argument
- In call by reference method the change of dummy argument can be affected in actual argument. So in this method the value of a and b exchange then the value of m and n also be change
- So there is a difference between call by value and call by reference

- **W.A.P to swap two integer value using call by reference method.**

```
#include<stdio.h>
#include<conio.h>
    void swap(int *a, int *b);
    void main()
{
    int m=22,n=44;
    printf (“Before swaping the value m=%d and n=%d”,m,n);
    swap(&m,&n);
     getch();
}
```

```
 void swap(int *a, int *b)
{
    int tmp;
    tmp=*a;
    *a=*b;
    *b=tmp;
   printf (“After swapping the value m=%d and
n=%d”,*a,*b);
}
```

3.7 Types of User-define Functions in C

- A User Defined Function has four types:

1. A function can return a value and no any argument

Ex: int sum( );

2. A function can return a value and have some argument

Ex: int sum (int a,int b);

3. A function does not return a value and no any argument

Ex: void sum( );

4. A function does not return a value and have some argument

Ex: void sum(int a,int b);

3.8 Advantages of using Functions

- It is used to avoid rewriting same logic or same code in the program
- There is no limit in calling the function to use the same functionality
- The C program is divided into number of User Defined Function
- It provide the reusability of code and divide a big task into small modual

3.9 Recursion

- Recursion is the process by which function call itself
- The recursion is very useful to implement a function for the taske which perform the same operation again and again
- The syntax of recursion is :

```
returntype functionname(arguments)
{
   body of the function
}
```

- The best example of recursion is finding factorial of given number

- **W.A.P to find factorial of given number.**

```
#include<stdio.h>
#include<conio.h>
int fact(int n);
void main()
{
        int n;
        printf (“Enter the integer number:”);
        scanf (“%d”,&n);
        printf (“The factorial is %d”,fact(n));
        getch();
}
        int fact(int n)
{
        if (n!=1)
        return(n*fact(n-1))
}
```

3.10 Advantages and Disadvantages of Recursion

- **Advantage:**
- It avoid the unnecessary calling of function
- It reduce the size of code
- It is very useful when some function call again and again

- **Disadvantage:**
- It is very difficult to understand
- It is difficult to trace the logic of the function

THANK YOU

Review questions

- Explain User defined function with its elements and proper example.
- Explain call by value and call by reference with proper example.
- Give the difference between call by value and call by reference.
- Explain Recursion with example.
- List the advantages and disadvantages of recursion.
- Write a program to find the factorial of given number using recursion.
- List the advantages of User defined function.