**Programs on Functions (or User defined functions), Programs on Parameter Passing mechanisms/or Argument Passing techniques, Programs on recursion**

1. **Write a program to add two numbers using function with no argument and no return value**

#include<stdio.h>

void add(); //Function declaration

int main()

{

add(); //Function calling

return 0;

}

//Function definition

void add()

{

int a,b,sum;

printf("\n Enter two numbers:");

scanf("%d%d",&a,&b);

sum=a+b;

printf("\n Sum of two numbers is:%d",sum);

}

1. **Write a program to add two numbers using function with argument but no return value**

#include<stdio.h>

void add(int,int); //Function declaration

int main()

{

int a,b;

printf("\n Enter two numbers:");

scanf("%d%d",&a,&b);

add(a,b); //Function calling //a, b are actual arguments

return 0;

}

//Function definition

void add(int x,int y) //x and y are formal arguments

{

int sum;

sum=x+y;

printf("\n Sum of two numbers is:%d",sum);

}

1. **Write a program to add two numbers using function no arguments but return a value**

#include<stdio.h>

int add(); //Function declaration

int main()

{

int result;

result=add(); //Function calling

printf("\n Sum is:%d",result);

return 0;

}

//Function definition

int add()

{

int a,b,sum;

printf("\n Enter two numbers:");

scanf("%d%d",&a,&b);

sum=a+b;

return sum;

}

1. **Write a program to add two numbers using function with arguments and return a value**

#include<stdio.h>

int add(int,int); //Function declaration

int main()

{

int a,b,result;

printf("\n Enter two numbers:");

scanf("%d%d",&a,&b);

result=add(a,b); //Function calling //a, b are actual arguments

printf("\n Sum is:%d",result);

return 0;

}

//Function definition

int add(int x,int y) //x and y are formal arguments

{

int sum;

sum=x+y;

return sum;

}

1. **Write a program to calculate factorial of a number using user defined function which accepts argument and returns value**

#include<stdio.h>

long long int factorial(int);

int main()

{

int n;

long long int result;

printf("\n Enter number:");

scanf("%d",&n);

result=factorial(n);

printf("\n Factorial is:%lld",result);

return 0;

}

long long int factorial(int x)

{

long long int fact=1;

int i;

if(x==0)

{

return 1;

}

else

{

for(i=1;i<=x;i++)

{

fact=fact\*i;

}

return fact;

}

}

1. **WAP to display the reverse of a number using user defined function.**

//When type of user defined function is not specified, then we can write the code with any one type out of the four types

#include<stdio.h>

void reverse(int);

int main()

{

int n;

printf("\n Enter number:");

scanf("%d",&n);

reverse(n);

return 0;

}

void reverse(int x)

{

int rem,rev=0;

while(x!=0)

{

rem=x%10;

rev=rev\*10+rem;

x=x/10;

}

printf("\n Reverse is:%d",rev);

}

1. **WAP to check whether the given number is prime or composite using user defined function**

#include<stdio.h>

int prime(int);

int main()

{

int n,result;

printf("\n Enter number:");

scanf("%d",&n);

result=prime(n);

if(result==0)

{

printf("\n Number is prime");

}

else

{

printf("\n Number is composite");

}

return 0;

}

int prime(int x)

{

int i,status=0;

for(int i=2;i<=x/2;i++)

{

if(x%i==0)

{

status=1;

break;

}

}

return status;

}

**Practice questions:**

* WAP to display n terms of Fibonacci series using user defined function which accepts argument and return no value
* WAP to check whether the entered number is palindrome or not using user defined function
* WAP to count total number odd numbers from m to n using user defined function
* WAP to display the largest of three numbers using user defined function
* WAP to swap the values of two variables without temporary variable using user defined function
* WAP to perform all arithmetic operations using individual user defined functions for every operation
* WAP to implement area calculator [any 5 shapes], using individual user defined functions for every area calculation[ Display menu on the screen, and call the user defined functions in the cases of switch depending upon the choice entered by user]
* WAP to perform the sum of the digits of a number using user defined function
* Design 5 more questions based on user defined functions of your choice and write their solutions also

**Parameter Passing mechanisms/ or Argument Passing techniques/ or Calling mechanisms in C**

**Program example 1—Call by value**

#include<stdio.h>

void add( int n);

int main()

{

int num = 2;

printf("\n The value of num before calling the function = %d", num);

add(num);

printf("\n The value of num after calling the function = %d", num);

return 0;

}

void add(int n)

{

n = n + 10;

printf("\n The value of num in the called function = %d", n);

}

**Program example 1- Call by Reference**

#include<stdio.h>

void add( int \*n);

int main()

{

int num = 2;

printf("\n The value of num before calling the function = %d", num);

add(&num);

printf("\n The value of num after calling the function = %d", num);

return 0;

}

void add( int \*n)

{

\*n = \*n + 10;

printf("\n The value of num in the called function = %d", \*n);

}

**2) Write a program to swap the values of two variables using call by value and call by reference in C**

**Call by value**

#include <stdio.h>

void swapByValue(int, int);

int main()

{

int n1,n2;

printf("\n Enter values of two numbers:");

scanf("%d%d",&n1,&n2);

swapByValue(n1,n2);

printf("\n Values of variables after swapping in main function are:%d,%d",n1,n2);

}

void swapByValue(int a, int b)

{

int t;

t = a;

a = b;

b = t;

printf("\n Values of variables after swapping in function definition are:%d,%d",a,b);

}

**Call by reference**

#include <stdio.h>

void swapByReference(int \*, int \*);

int main()

{

int n1,n2;

printf("\n Enter two numbers:");

scanf("%d%d",&n1,&n2);

swapByReference(&n1,&n2);

printf("\nValues of variables after swapping in main function are:%d,%d",n1,n2);

}

void swapByReference(int \*a, int \*b)

{

int t;

t = \*a;

\*a = \*b;

\*b = t;

printf("\nValues of variables after swapping in function definition are:%d,%d",\*a,\*b);

}

**Practice questions:**

* WAP to calculate factorial of a number using call by value and call by reference parameter passing mechanisms
* WAP to display the reverse of a number using call by value and call by reference parameter passing mechanisms
* WAP to display the smallest of three numbers using call by reference parameter passing mechanism
* WAP to check whether the entered number is Armstrong or not using call by value parameter passing mechanism.

**Programs on Recursion and Mathematical functions**

**1)** **Write a program to find the factorial of a number using recursion(or recursive function)**

#include <stdio.h>

long long int factorial(int);

int main()

{

int n;

printf("Enter a positive integer: ");

scanf("%d", &n);

printf("Factorial of %d = %lld", n, factorial(n));

return 0;

}

long long int factorial(int n)

{

if (n ==0||n==1)

return 1;

else

return n\*factorial(n-1);

}

**2)** **Write a program to find the sum of all numbers from 1 to n using recursion(or recursive function)**

#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;

else

return n + addNumbers(n-1);

}

**3)** **WAP to display n terms of Fibonacci series using recursion(or recursive function)**

#include<stdio.h>

int Fibonacci(int);

int main()

{

int n, i;

printf("\n Enter number of terms you want to print:");

scanf("%d",&n);

printf("Fibonacci series\n");

for(i=0;i<n;i++)

{

printf("%d\n", Fibonacci(i));

}

return 0;

}

int Fibonacci(int n)

{

if ( n == 0 )

return 0;

else if ( n == 1 )

return 1;

else

return ( Fibonacci(n-1) + Fibonacci(n-2) );

}

**Program example for Mathematical functions**

#include<stdio.h>

#include<math.h>

int main()

{

double x=9.0,y=8.0,z=7.0;

printf("\nLog value is:%lf",log(x));

printf("\nLog value with base 10 is:%lf",log10(x));

printf("\nExponential value is:%lf",exp(x));

printf("\n Ceil value is:%lf",ceil(8.94));

printf("\n Floor value is:%lf",floor(2.34));

printf("\n Power:%lf",pow(3.0,2.0));

printf("\n Floating absolute is:%lf",fabs(-2.9));

printf("\n Square root value is:%lf",sqrt(9));

printf("\n Sin:%f,cos:%f,tan:%lf",sin(x),cos(y),tan(z));

printf("\n fMod:%f",fmod(2.0,1.5));

return 0;

}