ASSIGNMENT – 4

QUESTION 1-

With a temporary variable-

#include <stdio.h>

int main()

{

double a, b, temp;

printf("Now I am going to show you how to swapt two vaiables with a temp variable\n");

printf("What is the first variable?\n");

scanf("%lf", &a);

printf("What is the second variable?\n");

scanf("%lf", &b);

temp = a;

a = b;

b = temp;

printf("Value of first variable: %.2lf\n", a);

printf("Value of second variable: %.2lf\n", b);

return 0;

}

A screenshot of a computer

Description automatically generated

Without temporary variable –

#include <stdio.h>

int main()

{

double a, b;

printf("Now I am going to show you how to swapt two vaiables without a temp variable\n");

printf("What is the first variable?\n");

scanf("%lf", &a);

printf("What is the second variable?\n");

scanf("%lf", &b);

// Code to swap 'x' and 'y. Let x = 10 and y = 5

a = a + b; // x now becomes 15

b = a - b; // y becomes 10

a = a - b; // x becomes 5

printf("Value of first variable: %.2lf\n", a);

printf("Value of second variable: %.2lf\n", b);

return 0;

}

A screenshot of a computer

Description automatically generated

With pointers –

#include <stdio.h>

int main()

{

double a, b;

double temp;

printf("Now I am going to show you how to swap two vaiables without a temp variable\n");

printf("What is the first variable?\n");

scanf("%lf", &a);

printf("What is the second variable?\n");

scanf("%lf", &b);

double \* addressOfA = &a;

double \* addressOfB = &b;

temp = \* addressOfA;

\* addressOfA = \* addressOfB;

\* addressOfB = temp;

printf("Value of first variable: %.2lf\n", a);

printf("Value of second variable: %.2lf\n", b);

return 0;

}

A screenshot of a computer

Description automatically generated

Question – 2

#include <stdio.h>

int factorial(int);

int main()

{

int no;

int facto;

printf("Now I am going to show you how to find factorial of a no.\n");

printf("Please enter the no.\n");

scanf("%d", &no);

if(no<0)

{

printf("Number is negative ABORTING!");

}

else

{

facto = factorial(no);

printf("Result: %d", facto);

}

return 0;

}

int factorial(int no)

{

int result;

if(no == 0)

{

result = 1;

}

else

{

result = no \* factorial(no-1);

}

return result;

}

A screenshot of a computer

Description automatically generated

Question 3 –

#include <stdio.h>

int main()

{

int no;

int i, a, b;

printf("Now I am going to show you whether a no is prime or not.\n");

printf("Please enter the no.\n");

scanf("%d", &no);

if(no<=0)

{

printf("Please select a positive no.\n");

}

else

{

for(i=2; i<no; i++)

{

if(no%i == 0)

{

a = 1;

}

else

{

b = 1;

}

}

}

if (a==1)

{

printf("No is a not prime no\n");

}

else if(b==1)

{

printf("No is a prime no.\n");

}

return 0;

}

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Question 4 –

#include <stdio.h>

int main()

{

int no=2;

int i, a=0, b=0, total, c=0;

printf("How many prime numbers do you want?\n");

scanf("%d", &total);

while(c < total)

{

if(no == 2)

{

printf("No %d is a prime no.\n", 2);

c = c + 1;

}

for(i=2; i<no; i++)

{

if(no%i == 0)

{

a = 1;

}

else

{

b = 1;

}

}

if((b==1) && (a==0))

{

printf("No %d is a prime no.\n", no);

c = c+1;

}

a = 0;

b = 0;

no = no + 1;

}

return 0;

}

A screenshot of a computer

Description automatically generated

Question 5 –

#include <stdio.h>

int main()

{

int a, b, hcf=0, i, j;

printf("Now I am going to tell you HCF of two numbers.\n");

printf("Please enter the first no.\n");

scanf("%d", &a);

printf("Please enter the second no.\n");

scanf("%d", &b);

if(a>b)

{

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

{

if(a%i == 0 && b%i == 0)

{

hcf = i;

}

}

printf("HCF: %d", hcf);

}

else if(b>a)

{

for(j=1; j<=a; j++)

{

if(a%j == 0 && b%j == 0)

{

hcf = j;

}

}

printf("HCF: %d", hcf);

}

return 0;

}

A screenshot of a computer

Description automatically generated

Question 6 –

#include <stdio.h>

int main()

{

int a, b, i=1, no;

int t=1;

int lcm;

printf("Now I am going to show you how to find LCM of two given numbers/\n");

printf("Please enter the first number.\n");

scanf("%d", &a);

printf("Please enter the second number.\n");

scanf("%d", &b);

while(t != 0)

{

no = a\*i;

if(no%b == 0)

{

lcm = no;

t = 0;

}

i = i + 1;

}

printf("The LCM of given no is %d", lcm);

return 0;

}

A screenshot of a computer

Description automatically generated

Question 7 –

#include <stdio.h>

int fibonacci(int);

int main()

{

int no, value, i;

printf("Now I am going to write fibonacci series of n numbers.\n");

printf("Upto how many numbers do you want the fibonacci series?\n");

scanf("%d", &no);

if(no<0)

{

printf("Please enter a positive value. Aborting!\n");

}

else

{

for(i=0; i<no; i++)

{

value = fibonacci(i+1);

printf("%d fibonacci no: %d\n", i+1, value);

}

}

return 0;

}

int fibonacci(int no)

{

int result;

if(no == 1)

{

result = 0;

}

else if(no == 2)

{

result = 1;

}

else

{

result = fibonacci(no-1) + fibonacci(no-2);

}

return result;

}

A screenshot of a computer

Description automatically generated

Question 8 –

#include <stdio.h>

int main()

{

int no, digit, sum=0;

printf("Now I am going to show you a program to add digits of a number.\n");

printf("Please enter the no.\n");

scanf("%d", &no);

while(no>=1)

{

digit = no % 10;

printf("Digit: %d\n", digit);

sum = sum + digit;

no = no/10;

}

printf("Sum: %d", sum);

return 0;

}

A screenshot of a computer

Description automatically generated

Question 9 –

Factorial of a number by recursion –

#include <stdio.h>

int factorial(int);

int main()

{

int no;

int value;

printf("Now I am going to find factorial of a number by recursion.\n");

printf("What is the number you want factorial of ?\n");

scanf("%d", &no);

if(no<0)

{

printf("Please enter a positive number. Aborting!\n");

}

else

{

value = factorial(no);

printf("%d! = %d", no, value);

}

return 0;

}

int factorial(int no)

{

int result;

if(no == 0)

{

result = 1;

}

else if(no == 1)

{

result = 1;

}

else

{

result = no \* factorial(no-1);

}

return result;

}

A screenshot of a computer

Description automatically generated

Fibonacci series by recursion –

#include <stdio.h>

int fibonacci(int);

int main()

{

int no, value, i;

printf("Now I am going to write fibonacci series of n numbers.\n");

printf("Upto how many numbers do you want the fibonacci series?\n");

scanf("%d", &no);

if(no<0)

{

printf("Please enter a positive value. Aborting!\n");

}

else

{

for(i=0; i<no; i++)

{

value = fibonacci(i+1);

printf("%d fibonacci no: %d\n", i+1, value);

}

}

return 0;

}

int fibonacci(int no)

{

int result;

if(no == 1)

{

result = 0;

}

else if(no == 2)

{

result = 1;

}

else

{

result = fibonacci(no-1) + fibonacci(no-2);

}

return result;

}

A screenshot of a computer

Description automatically generated

Summation of n by recursion –

#include <stdio.h>

int summation(int);

int main()

{

int no, value;

printf("Now I am going to show you summation of first n natural numbers.\n");

printf("Upto which number do want the summation of ?\n");

scanf("%d", &no);

if(no<0)

{

printf("Please enter a positive no. Aborting!\n");

}

else

{

value = summation(no);

printf("Summation of first %d natural numbers = %d\n", no, value);

}

return 0;

}

int summation(no)

{

int result;

if(no==1)

{

result = 1;

}

else

{

result = no + summation(no-1);

}

return result;

}

Text

Description automatically generated with low confidence

Question 10 –

Return type int –

#include <stdio.h>

int sum(int, int); // Means return type of the function is int(integer) and has 2 parameters which are both int

int main()

{

int a, b, value;

printf("Now I am going to show you various return types like void, int, float, char etc.\n");

printf("Now I am going to add 2 numbers by help of a function called sum and use return type int.\n");

printf("Please enter the first no.\n");

scanf("%d", &a);

printf("Please enter the second no.\n");

scanf("%d", &b);

value = sum(a, b); // a and b here are the values of arguments for the function sum

printf("%d + %d = %d\n", a, b, value);

return 0;

}

int sum(int x, int y)

{

int result;

result = x + y;

return result;

}

A screenshot of a computer

Description automatically generated

Return type float –

#include <stdio.h>

float sum(float, float); // Means return type of the function is float and has 2 parameters which are both float

int main()

{

float a, b, value;

printf("Now I am going to show you various return types like void, int, float, char etc.\n");

printf("Now I am going to add 2 numbers by help of a function called sum and use return type float\n");

printf("Please enter the first no.\n");

scanf("%f", &a);

printf("Please enter the second no.\n");

scanf("%f", &b);

value = sum(a, b); // a and b here are the values of arguments for the function sum

printf("%f + %f = %f\n", a, b, value);

return 0;

}

float sum(float x, float y)

{

float result;

result = x + y;

return result;

}

A screenshot of a computer

Description automatically generated

Return type void –

#include <stdio.h>

void sum(float, float); // Means return type of the function is void and has 2 parameters which are both float

int main()

{

float a, b;

printf("Now I am going to show you various return types like void, int, float, char etc.\n");

printf("Now I am going to add 2 numbers by help of a function called sum and use return type void\n");

printf("Please enter the first no.\n");

scanf("%f", &a);

printf("Please enter the second no.\n");

scanf("%f", &b);

sum(a, b); // a and b here are the values of arguments for the function sum

return 0;

}

void sum(float x, float y)

{

float result;

result = x + y;

printf("%f + %f = %f\n", x, y, result);

}

A screenshot of a computer

Description automatically generated

Return type char –

#include <stdio.h>

char getcharacter(); // Means return type of the function is char and has no parameters

int main()

{

char value;

printf("Now I am going to show you various return types like void, int, float, char etc.\n");

printf("Now you are going to input a charcter by help of function called getcharacter and use return type char\n");

value = getcharacter();

printf("The character you enetered is %c\n", value);

return 0;

}

char getcharacter()

{

char result;

printf("Please enter a character.\n");

scanf("%c", &result);

return result;

}

A screenshot of a computer

Description automatically generated