C assignments – 26-Sept-2022 to 30-Sept-2022

Name – Shreyas Raju Awankar

**All the questions are typed in comments of the code.**

**26-Sept-2022**

**Que.1**

// Write a program in C to accept two matrices and check wheather they are equal.

#include <stdio.h>

void main()

{

    int ra, rb, ca, cb;

    int i, j;

    int a[ra][ca];

    int b[rb][cb];

    int valueChecker = 1;

    printf("Enter how many rows you want in first matrix\n");

    scanf("%d", &ra);

    printf("Enter how many columns you want in first matrix\n");

    scanf("%d", &ca);

    printf("Enter how many rows you want in second matrix\n");

    scanf("%d", &rb);

    printf("Enter how many columns you want in second matrix\n");

    scanf("%d", &cb);

    if (ra == rb && ca == cb)

    {

        printf("Enter the element of the first matrix.\n");

        for (i = 0; i < ra; i++)

        {

            for (j = 0; j < ca; j++)

            {

                printf("a[%d][%d] = ?\n", i, j);

                scanf("%d", &a[i][j]);

            }

        }

        printf("\nEnter the elements of the second matrix.\n");

        for (i = 0; i < rb; i++)

        {

            for (j = 0; j < cb; j++)

            {

                printf("b[%d][%d] = ?\n", i, j);

                scanf("%d", &b[i][j]);

            }

        }

        for (i = 0; i < rb; i++)

        {

            for (j = 0; j < cb; j++)

            {

                if (a[i][j] != b[i][j])

                    valueChecker = 0;

            }

        }

        if (valueChecker == 1)

            printf("Two entered matrices are equal.");

        else

            printf("Two entered matrices are not equal as their elements are not equal");

    }

    else

        printf("Two entered matrices are not equal since their sizes are not same.");

}

// Write a program to convert a binary number into a decimal number.

#include <stdio.h>

void main(){

    int a,i,j,sum=0;

    printf("Enter your binary number\n");

    scanf("%d",&a);

    for (a, i=1; a>0; a/=10,i+=i ){

        if(a%10==1)

        sum+=i;

    }

    printf("The decimal of the entered binary number is %d\n", sum);

}

**Que.2**

// Write a program to convert a binary number into a decimal number.

#include <stdio.h>

int power(int a, int b){

    int c=1 ;

    while(b>0){

        c \*= 2;

        b--;

    }

    return c;

}

void main(){

    int binArray[16];

    int a,i,j,sum=0;

    printf("Enter your binary number strictly in 16 bit format.\n");

    scanf("%d",&a);

    for (i=15, j=0; i>=0, j<15; i--,j++){

        binArray[i]=a%10;

        if(a%10==1)

        sum+=power(2,j);

        a/=10;

    }

    printf("The decimal of the entered binary number is %d\n", sum);

}

**Que3.**

// Write a program to convert a binary number into a decimal number using math function.

#include <stdio.h>

#include <math.h>

void main()

{

    int binArray[16];

    int a, i, j, sum = 0;

    printf("Enter your binary number strictly in 16 bit format.\n");

    scanf("%d", &a);

    for (i = 15, j = 0; i >= 0, j < 15; i--, j++)

    {

        binArray[i] = a % 10;

        if (a % 10 == 1)

            sum += pow(2, j);

        a /= 10;

    }

    printf("The decimal of the entered binary number is %d\n", sum);

}

**Que.4**

// Write a program to reverse a 1D array.

#include <stdio.h>

void main()

{

    int e, i, j;

    printf("How many elements you want in your array?\n");

    scanf("%d", &e);

    int array[e];

    int reversed\_array[e];

    printf("Enter the array elements\n");

    for (i = 0 ; i < e; i++)

    {

        printf("array[%d] = ?\n",i);

        scanf("%d", &array[i]);

    }

    // Reversing the array.

    for (i = e-1, j = 0; i >= 0, j <= e-1; i--, j++)

    {

        reversed\_array[j] = array[i];

    }

    // Printing the reversed array.

    printf("The reversed array elements are\n");

    for (i = 0 ; i < e; i++)

    {

        printf("%d  ", reversed\_array[i]);

    }

}

**Que.5**

// Write a program to find the second largest element in an array.

#include <stdio.h>

void main()

{

    int e, i;

    printf("How many elements you want in your array?\n");

    scanf("%d", &e);

    int array[e];

    printf("Enter the array elements\n");

    for (i = 0; i < e; i++)

    {

        printf("array[%d] = ?\n", i);

        scanf("%d", &array[i]);

    }

    int max1 = array[0];

    for (i = 0; i < e; i++)

    {

        if (array[i] > max1)

        {

            max1 = array[i];

        }

    }

    printf("%d ", max1);

    int max2 = -2147483648;

    for (i = 0; i < e; i++)

    {

        if (array[i] >= max2 && array[i] < max1)

        // if (array[i] >= max2 )

        {

            max2 = array[i];

        }

    }

    printf("The second largest element in your array is %d ", max2);

}

**Que.6**

// Write a program to find the second smallest element in an array.

#include <stdio.h>

void main()

{

    int e, i;

    printf("How many elements you want in your array?\n");

    scanf("%d", &e);

    int array[e];

    printf("Enter the array elements\n");

    for (i = 0; i < e; i++)

    {

        printf("array[%d] = ?\n", i);

        scanf("%d", &array[i]);

    }

    int min1 = array[0];

    for (i = 1; i < e; i++)

    {

        if (array[i] < min1)

        {

            min1 = array[i];

        }

    }

    // printf("%d ", min1);

    int min2 = 2147483647;

    for (i = 2; i < e; i++)

    {

        if (array[i] <= min2 && array[i] > min1)

        {

            min2 = array[i];

        }

    }

    printf("The second smallest element in your array is %d ", min2);

}

**27-Sept-2022**

**Que.1**

// Write a program to make a calculator using user defined functions.

#include <stdio.h>

#include <stdlib.h>

float adder(float x, float y)

{

    float d = x + y;

    return d;

}

float sub(float x, float y)

{

    float d = x - y;

    return d;

}

float devider(float x, float y)

{

    float d = x / y;

    return d;

}

void main()

{

    char ch = 'y';

    do{

    float a, b;

    int choice;

    printf("Enter two numbers\n");

    scanf("%f%f", &a, &b);

    printf("Please enter the operation choice: \n 1 -> add \n 2 -> substract \n 3 -> devide \n 4 -> exit the program\n");

    scanf("%d", &choice);

    switch (choice)

    {

    case 1:

        printf("The addition of two numbers is %.2f\n", adder(a, b));

        break;

    case 2:

        printf("The substraction of two numbers is %.2f\n", sub(a, b));

        break;

    case 3:

        printf("The division of two numbers is %.2f\n", devider(a, b));

        break;

    case 4:

        exit(0);

    }

    printf("Do you want to continue?[y/n]\n");

    scanf("%s",&ch);

    }while (ch != 'n');

}

**Que.2**

// Write a menu driven program

// choice 1 : Check for perfect number.

// choice 1 : Check for palindrome number.

// choice 1 : Check for Armstrong number.

// Using functions.

// This program may not work in vs code for Armstrong number in VS code but it worked on other online compilers

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int perfect(int a)

{

    int sum = 0;

    for (int i = 1; i <= a / 2; i++)

    {

        if (a % i == 0)

            sum += i;

    }

    if (sum == a)

        return 1;

    return 0;

}

int Armstrong(int a)

{

    int b = a, c, sum = 0, count = 0;

    c = a;

    while (c > 0)

    {

        count++;

        c /= 10;

    }

    while (a > 0)

    {

        sum += pow(a % 10, count);

        a /= 10;

    }

    if (sum == b)

        return 1;

    return 0;

}

int palindrome(int a)

{

    int reversed = 0, r, b = a;

    while (a > 0)

    {

        r = a % 10;

        reversed = reversed \* 10 + r;

        a /= 10;

    }

    if (reversed == b)

        return 1;

    return 0;

}

void main()

{

    char option = 'y';

    do

    {

        int x, choice;

        printf("Enter a number\n");

        scanf("%d", &x);

    ABC:

        printf("Enter your choice\n 1 -> To check for a perfect number \n 2 -> To check for an Armstrong number \n 3 -> To check for a palindrome number.\n");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            if (perfect(x) == 0)

                printf("The given number is not a perfect number.\n");

            else

                printf("The given number is a perfect number.\n");

            break;

        case 2:

            if (Armstrong(x) == 1)

                printf("The given number is an Armstrong number.\n");

            else

                printf("The given number is not an Armstrong number.\n");

            break;

        case 3:

            if (palindrome(x) == 0)

                printf("The given number is not a palindrome number.\n");

            else

                printf("The given number is a palindrome number.\n");

            break;

        case 4:

            exit(0);

        default:

            printf("Please enter the correct choice out of 1,2,3,4.\n");

            goto ABC;

        }

        printf("Do you want to continue?[y/n]\nAny thing except 'n' will be treated as 'y' \n");

        scanf(" %c", &option);

    } while (option != 'n');

}

**Que.3**

// Write a program to create a function called prime and send any value from main and check if the number is prime or not.

#include <stdio.h>

int prime(int a){

    for (int i =2 ; i<= a/2; i++){

        if (a%i==0)

        return 0;

    }

    return 1;

}

void main(){

    int k, l;

    printf("Enter your number\n");

    scanf("%d", &k);

    l = prime(k);

    if (l==1)

    printf("Entered number %d is prime\n", k);

    else

    printf("Entered number %d is not prime\n", k);

}

**28-Sept-2022**

**Que.1**

/\* Write a program to pass any function using call by address and perform square of all the parameters in alled function and print the result in calling function. \*/

#include <stdio.h>

void squar(int \*x, int \*y,int \*z){

    \*x \*= \*x;

    \*y \*= \*y;

    \*z \*= \*z;

}

void main(){

    int a,b,c;

    printf(“Enter your numbers\n”);

    scanf(“%d%d%d”, &a,&b,&c);

    squar(&a,&b,&c);

    printf(“The squares are %d %d %d”,a, b,c);

}

**Que.2**

// Write a program to pass the elements of an array to a function and check how many numbers are prime.

#include <stdio.h>

int prime(int a)

{

    for (int i = 2; i <= a / 2; i++)

    {

        if (a % i == 0)

            return 0;

    }

    return 1;

}

void main()

{

    int a, count = 0;

    printf("How many array elements do you want?\n");

    scanf("%d", &a);

    int arr[a];

    for (int i = 0; i < a; i++)

    {

        printf("a[%d]=? \n", i);

        scanf("%d", &arr[i]);

        if (prime(arr[i]) == 1)

            count++;

    }

    printf("There are %d prime elements in your array", count);

}

**Que.3**

/\* Write a program to pass an entire array in a function and square every element at that array in called function and display that array in both the functions. \*/

#include <stdio.h>

void squar(int array[], int a)

{

    printf("\nArray elements in called functions are\n");

    for (int i = 0; i < a; i++)

    {

        array[i] \*= array[i];

        printf("%d ", array[i]);

    }

}

#include <stdio.h>

void main()

{

    int a;

    printf("How many elements you want in your array?\n");

    scanf("%d",&a);

    int array[a];

    for (int i = 0; i < a; i++)

    {

        printf("array[%d] = ?\n", i);

        scanf("%d", &array[i]);

    }

    squar(array, a);

    printf("\nThe array elements in calling function are\n");

    for (int i = 0; i < a; i++)

    {

        printf("%d ", array[i]);

    }

}

**Que.4**

/\* Write a program to pass an entire array in a function and square every element at that array in called function and display that array in both the functions. \*/

#include <stdio.h>

void squar(int array[], int a)

{

    printf("\nArray elements in called functions are\n");

    for (int i = 0; i < a; i++)

    {

        array[i] \*= array[i];

        printf("%d ", array[i]);

    }

}

#include <stdio.h>

void main()

{

    int a;

    printf("How many elements you want in your array?\n");

    scanf("%d",&a);

    int array[a];

    for (int i = 0; i < a; i++)

    {

        printf("array[%d] = ?\n", i);

        scanf("%d", &array[i]);

    }

    printf("The array elements in calling function are\n");

    for (int i = 0; i < a; i++)

    {

        printf("%d ", array[i]);

    }

    squar(array, a);

}

**Que.5**

// Write a program to pass only even index elements of an array to a function.

#include <stdio.h>

void evenPrinter(int a, int i)

{

    printf("\narray [%d] = %d \n", i, a);

}

void main()

{

    int a, i;

    printf("How many elements you want in your array?\n");

    scanf("%d", &a);

    int array[a];

    for (i = 0; i < a; i++)

    {

        printf("array[%d] = ?\n", i);

        scanf("%d", &array[i]);

    }

    for (i = 0; i < a; i++)

    {

        if (i % 2 == 0)

            evenPrinter(array[i], i);

    }

}

**29-Sept-2022**

**Que.1**

// Write a program in C to convert a decimal number to binary using recursion.

#include <stdio.h>

int dec(int x){

    int c;

    c = x%2;

    if (x>1)

    dec(x/2);

    printf("%d", c);

}

void main(){

    int a,f;

    printf("Enter a decimal number\n");

    scanf("%d",&a);

    dec(a);

}

**Que.2**

// Write a c program to check two given integers and return the value whichever value is nearest to 13 without going over. Return 0 if both numbers go over.

#include <stdio.h>

int nearest(int x, int y)

{

    if (x >= 13 && y >= 13)

    {

        return 0;

    }

    else if (x < 13 && y > 13)

    {

        int z = x, r = y, counter1 = 0, counter2 = 0;

        while (z <= 13)

        {

            counter1++;

            z++;

        }

        while (r >= 13)

        {

            counter2++;

            r--;

        }

        if (counter1 < counter2)

            return x;

        else if (counter1 == counter2)

            return 2;

        else

            return y;

    }

    else if (x > 13 && y < 13)

    {

        int z = x, r = y, counter1 = 0, counter2 = 0;

        while (z >= 13)

        {

            counter1++;

            z--;

        }

        while (r <= 13)

        {

            counter2++;

            r++;

        }

        if (counter1 < counter2)

            return x;

        else if (counter1 == counter2)

            return 2;

        else

            return y;

    }

}

void main()

{

    int a, b, near;

    printf("Enter two numbers\n");

    scanf("%d%d", &a, &b);

    near = nearest(a, b);

    if(nearest(a,b)!=0 && nearest(a,b)!=2)

    printf("The nearest number to 13 is %d ", near);

    else if (nearest(a,b)==2)

    printf("Both the numbers share the same distance from 13");

    else

    printf("Both the numbers go over 13");

}

**Que.3**

// Write a program in C to count the digit in a given number using recursion.

#include <stdio.h>

int counter(int x, int y){

    int c = x%10;

    if (x==0)

    return 0;

    else if (c == y)

    return 1+counter(x/10,y);

    else

    return counter(x/10,y);

}

void main()

{

    int a,b, count;

    printf("Enter your number\n");

    scanf("%d", &a);

    printf("Enter your digit\n");

    scanf("%d", &b);

    count = counter(a,b);

    printf("The digit occured %d times",count);

}

**Que.4**

// Write a program in C to convert a decimal number to binary using recursion.

#include <stdio.h>

int dec(int x){

    int c;

    c = x%2;

    if (x>1)

    dec(x/2);

    printf("%d", c);

}

void main(){

    int a,f;

    printf("Enter a decimal number\n");

    scanf("%d",&a);

    dec(a);

}

**Que.5**

// Write a c program to check two given integers and return the value whichever value is nearest to 13 without going over. Return 0 if both numbers go over.

#include <stdio.h>

int nearest(int x, int y)

{

    if (x >= 13 && y >= 13)

    {

        return 0;

    }

    else if (x < 13 && y > 13)

    {

        if (13 - x < y - 13)

            return x;

        else if (13 - x == y - 13)

            return 2;

        else

            return y;

    }

    else if (x > 13 && y < 13)

    {

        if (x - 13 < 13 - y)

            return x;

        else if (x - 13 == 13 - y)

            return 2;

        else

            return y;

    }

    else

    {

        if (13 - x < 13 - y)

            return x;

        else if (x - 13 == 13 - y)

            return 2;

        else

            return y;

    }

}

void main()

{

    int a, b, near;

    printf("Enter two numbers\n");

    scanf("%d%d", &a, &b);

    near = nearest(a, b);

    if (nearest(a, b) != 0 && nearest(a, b) != 2)

        printf("The nearest number to 13 is %d ", near);

    else if (nearest(a, b) == 2)

        printf("Both the numbers share the same distance from 13");

    else

        printf("Both the numbers go over 13");

}

**Que.6**

// WAP for power function using recursion..

#include <stdio.h>

int power(int a, int b)

{

    if (b == 0)

        return 1;

    else

        return a \* power(a,b - 1);

}

void main()

{

    int a, b, pow;

    printf("Enter a number to make power of\n");

    scanf("%d", &a);

    printf("Enter the value which you want to power the number by\n");

    scanf("%d", &b);

    pow = power(a,b);

    printf("The power of %d by %d is %d", a, b, pow);

}

**Que.7**

//Write a program to check if a triple is present in an array of integers or not. If the vale appears three times in a row in an array is called triple.

#include <stdio.h>

void main(){

    int a,i,c=0;

    printf("How many elements you want in your array?\n");

    scanf("%d",&a);

    int array[a+2];

    for(i=0; i<a; i++){

        printf("array[%d]=?\n", i);

        scanf("%d",&array[i]);

    }

    for(i=0; i<a; i++){

        if(array[i] == array[i+1] && array[i] == array[i+2]){

        c=1;

        printf("%d is present for 3 times contiguously.", array[i]);

        }

    }

    if (c==0)

    printf("No element repeated for 3 times contiguously.");

}