***C Programming Questions***

***Weekly Assignments***

***Name: Gitansh gupta***

***Section: AU-02***

***Roll no.:74***

***Week 1***

Q. 1 Write a program to accept height and base of triangle and

calculate area of Triangle

Note: area =(h\*b)/2

#include<stdio.h>

intmain()

{

inth,b,area;

printf("Enter the height:");

scanf("%d"

,&h);

printf("Enter the base:");

scanf("%d"

,&b);

area=(h\*b)/2;

printf("Area of

triangle:%d"

,area);

return0;

}

Q. 2 Write a program to accept radius of circle and calculate area of

circle

Note: area =pi \* r2

#include<stdio.h>

intmain()

{

intr,area;

printf("Enter the radius of

circle:");

scanf("%d"

,&r);

area=3.14\*r\*r;

printf("Area of circle:%d"

,area);

return0;

}

Q. 3 Write a program to find the lowest marks of three students

using conditional operator.

#include<stdio.h>

voidmain() {

inta, b, c;

printf("Enter the marks of three

students: ");

scanf("%d%d%d"

, &a, &b, &c);

(a<b) ? ((a<c) ?printf("Lowest

marks: %d"

, a) :printf("Lowest marks:

%d"

, c)) : ((b<c) ?printf("Lowest

marks: %d"

, b) :printf("Lowest marks:

%d"

, c));

}

Q. 4 Write a program to Calculate Compound Interest.

# include<stdio.h>

# include<math.h>

intmain()

{

floatp,t,r,amt,ci;

printf("Enter the principal

amount:");

scanf("%f"

,&p);

printf("Enter the annual rate:");

scanf("%f"

,&r);

printf("Enter the annual time:");

scanf("%f"

,&t);

amt=p\*pow((1+r/100),t);

printf("amount is %.2f"

,amt);

ci=amt-p;

printf("\ncompound interest is

%.2f"

,ci);

return0;

}

Q. 5 Write a program to Calculate Cube of a Number.

# include<stdio.h>

# include<math.h>

intmain()

{

intn,cube;

printf("enter the number:");

scanf("%d"

,&n);

cube=pow(n,3);

printf("Cube of given number is

%d"

,cube);

return0;

}

***Week – 2***

Q. 1 Write a program to interchange two values by using Assignment

Operator.

#include<stdio.h>

intmain()

{

inta,b,temp=0;

printf ("enter the value of a:");

scanf("%d"

,&a);

printf("enter the value of b:");

scanf("%d"

,&b);

temp=a;

a=b;

b=temp;

printf("value of a after

swaping:%d"

,a);

printf("\nvalue of b after

swaping:%d"

,b);

return0;

}

Q. 2 Write a program to interchange two values by using Arithmetic

Operator.

# include<stdio.h>

intmain()

{

inta,b;

printf("enter the value for a:");

scanf("%d"

,&a);

printf("enter the value for b:");

scanf("%d"

,&b);

a=a+b;

b=a-b;

a=a-b;

printf("after swaping value of

a:%d"

,a);

printf("\nafterswaping value of

b:%d"

,b);

return0;

}

Q. 3 Write a program to interchange two values by using Bitwise

Operator.

# include<stdio.h>

intmain()

{

inta,b;

printf ("enter the value of a:");

scanf("%d"

,&a);

printf("enter the value of b:");

scanf("%d"

,&b);

a=a^b;

b=a^b;

a=b^a;

printf("value of a after

swapping:%d"

,a);

printf("\nvalue of b after

swapping:%d"

,b);

return0;

}

Q. 4 Write a program to find the size of all data types (Int, Float,

Char, Double, Long Double, Short Int etc.).

# include<stdio.h>

intmain()

{

printf("the size of

int:%lu"

,sizeof(int));

printf("\nthe size of

float:%lu"

,sizeof(float));

printf("\nthe size of

char:%lu"

,sizeof(char));

printf("\nthe size of

double:%lu"

,sizeof(double));

printf("\nthe size of long

double:%lu"

,sizeof(longdouble));

printf("\nthe size of short

int:%lu"

,sizeof(shortint));

return0;

}

Q. 5 Write a program to find out whether input number is even or

odd without using arithmetic operators.

#include<stdio.h>

intmain() {

intnum;

printf("Enter an integer: ");

scanf("%d"

, &num);

(num&1) ?printf("%d is odd.",

num) :printf("%d is even.", num);

return0;

}

***Week – 3***

Q. 1 Write a C program to check whether a given number is even or

odd.

# include<stdio.h>

intmain()

{

intn;

printf("enter the number:");

scanf("%d"

,&n);

if(n%2!=0)

{

printf("Odd!!!");

}

else

{

printf("Even!!!");

}

}

Q. 2 Write a C program to check whether a given number is positive

or negative.

# include<stdio.h>

intmain()

{

intn;

printf("enter the number:");

scanf("%d"

,&n);

if(n>0)

{

printf("positive!!!");

}

elseif(n<0)

{

printf("negative!!!");

}

else{

printf("zero");

}

}

Q. 3 Write a C program to find whether a given year is a leap year or

not.

# include<stdio.h>

intmain()

{

intn;

printf("enter the year:");

scanf("%d"

,&n);

if(n%4!=0)

{

printf("not a leap year!!!");

}

else

{

printf("leap year!!!");

}

return0;

}

Q. 4 Write a C program to find the largest of three numbers.

#include<stdio.h>

intmain()

{

intm1,m2,m3;

printf("enter the first

number:");

scanf("%d"

,&m1);

printf("enter the second

number:");

scanf("%d"

,&m2);

printf("enter the third

number:");

scanf("%d"

,&m3);

if(m1>m2&&m1>m3)

{

printf("first number is the

largest no.");

}

elseif(m2>m1&&m2>m3)

{

printf("second number is the

largest no.");

}

else

{

printf("third number is the

largest no.");

}

return0;

}

Q. 5 Write a C program to read temperature in centigrade and

display a suitable message according to the temperature state

below: a. Temp < 0 then Freezing weather b. Temp 0-10 then Very

Cold weather c. Temp 10-20 then Cold weather d. Temp 20-30 then

Normal in Temp e. Temp 30-40 then Its Hot f. Temp >=40 then Its

Very Hot

# include<stdio.h>

intmain()

{

inttemp;

printf("enter the temperature in

centigrade:");

scanf("%d"

,&temp);

if(temp<=0)

{

printf("freezing weather");

}

elseif(temp<=10)

{

printf("very cold weather");

}

elseif(temp<=20)

{

printf("cold weather");

}

elseif(temp<=30)

{

printf("normal temperature");

}

elseif(temp<=40)

{

printf("its hot");

}

elseif(temp>40){

printf("very hot");

}

return0;

}

Q. 6 Write a C program to read any digit and display it in the word.

# include<stdio.h>

intmain()

{

intn;

printf("enter the digit to be

displayed:");

scanf("%d"

,&n);

switch (n)

{

case0: printf("Zero");

break;

case1: printf("one");

break;

case2: printf("Two");

break;

case3: printf("Three");

break;

case4:printf("Four");

break;

case5:printf("Five");

break;

case6:printf("Six");

break;

case7:printf("Seven");

break;

case8:printf("Eight");

break;

case9:printf("nine");

break;

default:printf("nit a digit");

break;

}

}

Q. 7 Write a C program to create a Simple Calculator using a switch

case.

#include<stdio.h>

intmain() {

charoperator;

doublefirst, second;

printf("Enter an operator (+, -,

\*, /): ");

scanf("%c"

, &operator);

printf("Enter two operands: ");

scanf("%lf%lf"

, &first, &second);

switch (operator) {

case'+':

printf("%.1lf + %.1lf =

%.1lf"

, first, second, first+second);

break;

case'-':

printf("%.1lf - %.1lf =

%.1lf"

, first, second, first-second);

break;

case'\*':

printf("%.1lf \* %.1lf =

%.1lf"

, first, second, first\*second);

break;

case'/':

printf("%.1lf / %.1lf =

%.1lf"

, first, second, first/second);

break;

default:

printf("Error! operator

is not correct");

}

return0;

}

Q. 8 Write a C program using C Switch…Case to Calculate the Area of

Rectangle/ Circle/ Triangle

#include<stdio.h>

intmain() {

intchoice;

floatbase, height, radius,

length, breadth, area;

printf("Switch Case in C Program

to Calculate Area of

Rectangle/Circle/Triangle\n");

printf("1. Calculate the area of

a circle\n");

printf("2. Calculate the area of

a rectangle\n");

printf("3. Calculate the area of

a triangle\n");

printf("Enter your choice (1, 2,

or 3): ");

scanf("%d"

, &choice);

switch (choice) {

case1:

printf("Enter the radius

of the circle: ");

scanf("%f"

, &radius);

area=3.14159\*radius\*radiu

s;

printf("The area of the

circle is: %f\n"

, area);

break;

case2:

printf("Enter the length

and breadth of the rectangle: ");

scanf("%f%f"

, &length,

&breadth);

area=length\*breadth;

printf("The area of the

rectangle is: %f\n"

, area);

break;

case3:

printf("Enter the base

and height of the triangle: ");

scanf("%f%f"

, &base,

&height);

area=0.5\*base\*height;

printf("The area of the

triangle is: %f\n"

, area);

break;

default:

printf("Invalid

choice\n");

break;

}

return0;

}

H.O.T.S Questions

Q. 9 Write a C program to calculate the sum and average of positive

numbers. If the user enters a negative number, the sum and average

are displayed.

# include<stdio.h>

intmain()

{

intsum,avr,n,n1,n2;

printf("enter the number:");

scanf("%d"

,n);

printf("enter the first

number:");

scanf("%d"

,&n1);

printf("enter the second

number:");

scanf("%d"

,&n2);

if(n>0)

{

sum+=i;

avr=sum/i;

printf("sum is%d"

,sum);

printf("average is

%d"

,avr);

}

else

{

printf("sum is%d"

,sum);

printf("average is

%d"

,avr);

}

return0;

}

Q. 10 Write a C program to design a digital clock.

#include<stdio.h>

#include<time.h>

intmain() {

while (1) {

time\_tcurrentTime=time(NULL);

structtm\*tm=localtime(&curren

tTime);

printf("%02d:%02d:%02d\n"

,

tm->tm\_hour, tm->tm\_min, tm->tm\_sec);

sleep(1);

}

return0;

}

Q. 11 Write a C program to find the sum of digits of a number until a

single digit is occurred

#include<stdio.h>

intmain() {

intnumber, sum;

printf("Enter a number: ");

scanf("%d"

, &number);

while (number>9) {

sum=0;

while (number!=0) {

sum+=number%10;

number/=10;

}

number=sum;

}

printf("The sum of digits until a

single digit is occurred: %d\n"

,

number);

return0;

}

***Week – 4***

Q. 1 Write a C program to print multiplication table of a number.

# include<stdio.h>

intmain()

{

intn,mul;

printf("enter the number:");

scanf("%d"

,&n);

for(inti=1;i<=10;++i)

{

mul=n\*i;

printf("%d\*%d=%d\n"

,n,i,mul);

}

return0;

}

Q. 2 Write a C program to calculate factorial of a number

# include<stdio.h>

intmain()

{

intn,i,fac=1;

printf("enter the number:");

scanf("%d"

,&n);

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

{

fac=fac\*i;

}

printf("factorial of %d is

%d"

,n,fac);

return0;

}

Q. 3 Write a C program to check whether a number is palindrome or

not.

#include<stdio.h>

intmain()

{

intn,i,r,rev=0;

printf("enter the number:");

scanf("%d"

,&n);

i=n;

while(i!=0)

{

r=i%10;

rev=rev\*10+r;

i=i/10;

}

if(rev==n)

{

printf("palindrome");

}

else{

printf("not palindrome");

}

return0;

}

Q. 4 Write a C program to count frequency of digits in a given

number.

#include<stdio.h>

intmain() {

intnum, digit, count;

printf("Enter a number: ");

scanf("%d"

, &num);

printf("Enter a digit to count:

");

scanf("%d"

, &digit);

count=0;

while (num>0) {

if (num%10==digit) {

count++;

}

num/=10;

}

printf("Frequency of digit %d in

the given number is %d"

, digit,

count);

return0;

}

Q. 5 Write a C program to find HCF(GCD) AND LCM of two numbers

#include<stdio.h>

intmain() {

intnum1, num2, i, gcd, lcm;

printf("Enter two numbers: ");

scanf("%d%d"

, &num1, &num2);

for (i=1; i<=num1&&i<=num2; ++i)

{

if (num1%i==0&&num2%i==0) {

gcd=i;

}

}

lcm= (num1\*num2) /gcd;

printf("HCF(GCD) of %d and %d is

%d\n"

, num1, num2, gcd);

printf("LCM of %d and %d is %d"

,

num1, num2, lcm);

return0;

}

Q. 6 Write a C program to print all prime numbers between 1 to n.

#include<stdio.h>

intmain() {

inti, j, n, flag;

printf("Enter a number: ");

scanf("%d"

, &n);

printf("Prime numbers between 1

and %d are: ", n);

for (i=2; i<=n; ++i) {

flag=0;

for (j=2; j<=i/2; ++j) {

if (i%j==0) {

flag=1;

break;

}

}

if (flag==0)

printf("%d "

, i);

}

return0;

}

Q. 7 Write a C program to print Fibonacci series up to n terms.

#include<stdio.h>

intmain() {

inti, n, t1=0, t2=1, nextTerm;

printf("Enter the number of

terms: ");

scanf("%d"

, &n);

printf("Fibonacci Series: ");

for (i=1; i<=n; ++i) {

printf("%d, ", t1);

nextTerm=t1+t2;

t1=t2;

t2=nextTerm;

}

return0;

}

Q. 8 Write a C program to print Armstrong numbers from 1 to n AND

check a given number is Armstrong numbers or not.

#include<stdio.h>

intmain()

{

intn,r,d,sum=0;

printf("enter the number:");

scanf("%d"

,&n);

d=n;

while(d!=0){

r=d%10;

sum+=r\*r\*r;

d=d/10;}

if(sum==n){

printf("armstrong number");

}

else{

printf("not armstrong number");

}

return0;

}

H.O.T.S Questions

Q. 9 Write a C program to print all Perfect numbers between 1 to n

AND Check a given number is Perfect numbers or not.

#include<stdio.h>

intmain() {

intnumber, sum;

printf("Enter a number: ");

scanf("%d", &number);

for (inti=1; i<=number; i++) {

sum=0;

for (intj=1; j<i; j++) {

if (i%j==0) {

sum+=j;

}

}

if (sum==i) {

printf("%d is a perfect number.\n", i);

}

}

printf("Perfect numbers between 1 and %d are: ", number);

for (inti=1; i<=number; i++) {

sum=0;

for (intj=1; j<i; j++) {

if (i%j==0) {

sum+=j;

}

}

if (sum==i) {

printf("%d ", i);

}

}

printf("\n");

return0;

}

Q. 10 Write a C program to print all Strong Numbers between 1 to n.

#include <stdio.h>

int main() {

int n;

printf("Enter the value of n: ");

scanf("%d", &n);

printf("Strong numbers between 1 and %d are:\n", n);

for (int i = 1; i<= n; i++) {

int originalNum = i;

int sum = 0;

int num = i;

while (num> 0) {

int digit = num % 10;

int factorial = 1;

for (int j = 1; j <= digit; j++) {

factorial \*= j;

}

sum += factorial;

num /= 10;

}

if (sum == originalNum) {

printf("%d\n", originalNum);

}

}

return 0;

}

***Week 5***

1.(a):

#include <stdio.h>

int main() {

int rows = 4;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= 5; j++) {

printf("\*");

}

printf("\n");

}

return 0;

}

(b):

#include <stdio.h>

int main() {

int rows = 5;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= rows; j++) {

printf("%d", j);

}

printf("\n");

}

return 0;

}

(c):

#include <stdio.h>

int main() {

int rows = 4;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= i; j++) {

printf("%d", j);

}

printf("\n");

}

return 0;

}

(d):

#include <stdio.h>

int main() {

int rows = 4;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= i; j++) {

printf("%d", i);

}

printf("\n");

}

return 0;

}

(e):

#include <stdio.h>

int main() {

int rows = 4;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= i; j++) {

printf("\*");

}

printf("\n");

}

return 0;

}

(f):

#include <stdio.h>

int main() {

int rows = 4;

for (int i = 0; i< rows; i++) {

for (int j = 0; j < rows - i - 1; j++) {

printf(" ");

}

for (int k = 0; k <= i; k++) {

printf("%c", 'A' + k);

}

printf("\n");

}

return 0;

}

(g):

#include <stdio.h>

int main() {

int rows = 4;

int counter = 1;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= i; j++) {

printf("%d", counter);

counter++;

}

printf("\n");

}

return 0;

}

(h):

#include <stdio.h>

int main() {

int rows = 5;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= i; j++) {

printf("%d", j % 2);

}

printf("\n");

}

return 0;

}

(i):

#include <stdio.h>

int main() {

int rows = 5;

for (int i = 5; i>= 1; i--) {

for (int j = 5; j >= i; j--) {

printf("%d", j);

}

printf("\n");

}

return 0;

}

(j):

#include <stdio.h>

int main() {

int rows = 5;

for (int i = 1; i<= rows; i++) {

for (int j = 5; j >= i; j--) {

printf("%d", j);

}

printf("\n");

}

return 0;

}

(k):

#include <stdio.h>

int main() {

int rows = 5;

int cols = 5;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= cols; j++) {

if (i == 1 || i == rows || j == 1 || j == cols) {

printf("\*");

} else {

printf(" ");

}

}

printf("\n");

}

return 0;

}

(L):

#include <stdio.h>

int main() {

int rows = 4;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= rows - i; j++) {

printf(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

printf("\*");

}

printf("\n");

}

return 0;

}

(m):

#include <stdio.h>

int main() {

int rows = 4;

for (int i = 1; i<= rows; i++) {

for (int j = 1; j <= rows - i; j++) {

printf(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

printf("\*");

}

printf("\n");

}

for (int i = rows - 1; i>= 1; i--) {

for (int j = 1; j <= rows - i; j++) {

printf(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

printf("\*");

}

printf("\n");

}

return 0;

}

(n):

#include <stdio.h>

int main() {

int i, j, k;

for (i = 3; i>= 0; i--) {

for (k = 0; k <i; k++) {

printf(" ");

}

for (j = 0; j <= 3 - i; j++) {

printf("%d", 7 - (i \* 2) + j);

}

printf("\n");

}

return 0;

}

***Week 6***

1.Write a menu driven program to insert and delete

elements of kth position to an array of size N.

#include <stdio.h>

int main() {

int N, choice, k, i;

printf("Enter the size of the array: ");

scanf("%d", &N);

int arr[N];

for (i = 0; i< N; i++) {

printf("Enter element at position %d: ", i + 1);

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

}

while (1) {

printf("\nMenu:\n");

printf("1. Insert element at kth position\n");

printf("2. Delete element at kth position\n");

printf("3. Display array\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter the position (1 to %d) to insert element: ", N + 1);

scanf("%d", &k);

if (k < 1 || k > N + 1) {

printf("Invalid position. Position should be between 1 and

%d.\n", N + 1);

} else {

printf("Enter the element to insert: ");

int newElement;

scanf("%d", &newElement);

for (i = N - 1; i>= k - 1; i--) {

arr[i + 1] = arr[i];

}

arr[k - 1] = newElement;

N++;

printf("Element inserted successfully.\n");

}

break;

case 2:

printf("Enter the position (1 to %d) to delete element: ", N);

scanf("%d", &k);

if (k < 1 || k > N) {

printf("Invalid position. Position should be between 1 and

%d.\n", N);

} else {

for (i = k - 1; i< N - 1; i++) {

arr[i] = arr[i + 1];

}

N--;

printf("Element deleted successfully.\n");

}

break;

case 3:

printf("Array elements: ");

for (i = 0; i< N; i++) {

printf("%d ", arr[i]);

}

printf("\n");

break;

case 4:

printf("Exiting the program.\n");

return 0;

default:

printf("Invalid choice. Please enter a valid option.\n");

}

}

return 0;

}

2.Write the program to print the biggest and

smallest element in an array.

#include <stdio.h>

int main() {

int N, i;

printf("Enter the size of the array: ");

scanf("%d", &N);

int arr[N];

for (i = 0; i< N; i++) {

printf("Enter element at position %d: ", i + 1);

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

}

int largest = arr[0];

int smallest = arr[0];

for (i = 1; i< N; i++) {

if (arr[i] > largest) {

largest = arr[i];

}

if (arr[i] < smallest) {

smallest = arr[i];

}

}

printf("The largest element in the array is: %d\n", largest);

printf("The smallest element in the array is: %d\n", smallest);

return 0;

}

3.Write the program to print the sum and average of

an array.

#include <stdio.h>

int main() {

int N, i;

printf("Enter the size of the array: ");

scanf("%d", &N);

int arr[N];

for (i = 0; i< N; i++) {

printf("Enter element at position %d: ", i + 1);

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

}

int sum = 0;

float average;

for (i = 0; i< N; i++) {

sum += arr[i];

}

average = (float)sum / N;

printf("The sum of the elements in the array is: %d\n", sum);

printf("The average of the elements in the array is: %.2f\n",

average);

return 0;

}

4.Write the program to sort an array using bubble

sort.

#include <stdio.h>

int main() {

int N, i, j, temp;

printf("Enter the size of the array: ");

scanf("%d", &N);

int arr[N];

for (i = 0; i< N; i++) {

printf("Enter element at position %d: ", i + 1);

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

}

for (i = 0; i< N - 1; i++) {

for (j = 0; j < N - i - 1; j++) {

if (arr[j] >arr[j + 1]) {

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

printf("Sorted array: ");

for (i = 0; i< N; i++) {

printf("%d ", arr[i]);

}

printf("\n");

return 0;

}

5.Write the program to search an element using linear

search as well as binary search.

#include <stdio.h>

int main() {

int N, i, element;

printf("Enter the size of the array: ");

scanf("%d", &N);

int arr[N];

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

for (i = 0; i< N; i++) {

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

}

printf("\nEnter the element to search using linear search: ");

scanf("%d", &element);

int linearIndex = -1;

for (i = 0; i< N; i++) {

if (arr[i] == element) {

linearIndex = i;

break;

}

}

if (linearIndex != -1) {

printf("Element %d found at position %d using linear search.\n",

element, linearIndex + 1);

} else {

printf("Element %d not found in the array using linear search.\n",

element);

}

printf("\nEnter the element to search using binary search: ");

scanf("%d", &element);

int low = 0, high = N - 1, mid, binaryIndex = -1;

while (low <= high) {

mid = (low + high) / 2;

if (arr[mid] == element) {

binaryIndex = mid;

break;

} else if (arr[mid] < element) {

low = mid + 1;

} else {

high = mid - 1;

}

}

if (binaryIndex != -1) {

printf("Element %d found at position %d using binary search.\n",

element, binaryIndex + 1);

} else {

printf("Element %d not found in the array using binary search.\n",

element);

}

return 0;

}

6.Take an array of 20 integer inputs from user and

print the following:

a. number of positive numbers

b. number of negative numbers

c. number of odd numbers

d. number of even numbers e. number of 0.

#include <stdio.h>

int main() {

int arr[20];

int positiveCount = 0, negativeCount = 0, oddCount = 0, evenCount

= 0, zeroCount = 0;

printf("Enter 20 integers:\n");

for (int i = 0; i< 20; i++) {

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

}

for (int i = 0; i< 20; i++) {

if (arr[i] > 0) {

positiveCount++;

} else if (arr[i] < 0) {

negativeCount++;

} else {

zeroCount++;

}

if (arr[i] % 2 == 0) {

evenCount++;

} else {

oddCount++;

}

}

printf("\na. Number of positive numbers: %d\n", positiveCount);

printf("b. Number of negative numbers: %d\n", negativeCount);

printf("c. Number of odd numbers: %d\n", oddCount);

printf("d. Number of even numbers: %d\n", evenCount);

printf("e. Number of zeros: %d\n", zeroCount);

return 0;

}

7.Take an array of 10 elements. Split it into middle and

store the elements in two different arrays.

#include <stdio.h>

int main(){

int initialArray[10];

int firstHalf[5], secondHalf[5];

printf("Enter 10 integers:\n");

for (int i = 0; i< 10; i++) {

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

}

for (int i = 0; i< 5; i++) {

firstHalf[i] = initialArray[i];

secondHalf[i] = initialArray[i + 5];

}

printf("\nINITIAL array: ");

for (int i = 0; i< 10; i++) {

printf("%d, ", initialArray[i]);

}

printf("\n");

printf("After splitting:\n");

printf("First Half: ");

for (int i = 0; i< 5; i++) {

printf("%d, ", firstHalf[i]);

}

printf("\n");

printf("Second Half: ");

for (int i = 0; i< 5; i++) {

printf("%d, ", secondHalf[i]);

}

printf("\n");

return 0;

}

8.Write the program to count frequency of each

element in an array.

#include <stdio.h>

int main() {

int N;

printf("Enter the size of the array: ");

scanf("%d", &N);

int arr[N];

printf("Enter %d integers:\n", N);

for (int i = 0; i< N; i++) {

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

}

int frequency[N];

for (int i = 0; i< N; i++) {

frequency[i] = 0;

}

for (int i = 0; i< N; i++) {

if (frequency[i] == -1) {

continue;

}

for (int j = i + 1; j < N; j++) {

if (arr[i] == arr[j]) {

frequency[j] = -1;

frequency[i]++;

}

}

}

printf("\nFrequency of each element:\n");

for (int i = 0; i< N; i++) {

if (frequency[i] != -1) {

printf("%d occurs %d times.\n", arr[i], frequency[i] + 1);

}

}

return 0;

}

***WEEK-7***

Question 1

#include<stdio.h>

#defineMAX\_ROWS3

#defineMAX\_COLS3

voidprintRowMajor(int matrix[MAX\_ROWS][MAX\_COLS]) {

printf("Row Major Order:\n");

for (inti=0; i<MAX\_ROWS; ++i) {

for (int j =0; j <MAX\_COLS; ++j) {

printf("%d\t", matrix[i][j]);

}

printf("\n");

}

}

voidprintColumnMajor(int

matrix[MAX\_ROWS][MAX\_COLS]) {

printf("\nColumn Major Order:\n");

for (int j =0; j <MAX\_COLS; ++j) {

for (inti=0; i<MAX\_ROWS; ++i) {

printf("%d\t", matrix[i][j]);

}

printf("\n");

}

}

intmain() {

intmatrix[MAX\_ROWS][MAX\_COLS] = {{1, 2, 3},

{4, 5, 6},

{7, 8, 9}};

printRowMajor(matrix);

printColumnMajor(matrix);

return0;

}

Question 2

#include<stdio.h>

#defineMAX\_ROWS3

#defineMAX\_COLS3

intcalculateMatrixSum(int

matrix[MAX\_ROWS][MAX\_COLS]) {

int sum =0;

for (inti=0; i<MAX\_ROWS; ++i) {

for (int j =0; j <MAX\_COLS; ++j)

{

sum += matrix[i][j];

}

}

returnsum;

}

intmain() {

intmatrix[MAX\_ROWS][MAX\_COLS] = {{1,

2, 3},

{4,

5, 6},

{7,

8, 9}};

int sum =calculateMatrixSum(matrix);

printf("Sum of the matrix: %d\n"

,

sum);

return0;

}

Question 3

#include<stdio.h>

#defineROWS3

#defineCOLS3

voidaddMatrices(intmat1[ROWS][COLS],

intmat2[ROWS][COLS],

intresult[ROWS][COLS]) {

for (inti=0; i<ROWS; ++i) {

for (int j =0; j <COLS; ++j) {

result[i][j] =mat1[i][j]

+mat2[i][j];

}

}

}

voidmultiplyMatrices(intmat1[ROWS][COLS],

intmat2[ROWS][COLS],

intresult[ROWS][COLS]) {

for (inti=0; i<ROWS; ++i) {

for (int j =0; j <COLS; ++j) {

result[i][j] =0;

for (int k =0; k <COLS; ++k)

{

result[i][j] +=mat1[i][k]

\*mat2[k][j];

}

}

}

}

voiddisplayMatrix(intmatrix[ROWS][COLS])

{

for (inti=0; i<ROWS; ++i) {

for (int j =0; j <COLS; ++j) {

printf("%d\t"

, matrix[i][j]);

}

printf("\n");

}

printf("\n");

}

intmain() {

int matrix1[ROWS][COLS] = {{1, 2, 3},

{4, 5, 6},

{7, 8,

9}};

int matrix2[ROWS][COLS] = {{9, 8, 7},

{6, 5, 4},

{3, 2,

1}};

intsumMatrix[ROWS][COLS];

intproductMatrix[ROWS][COLS];

addMatrices(matrix1, matrix2,

sumMatrix);

multiplyMatrices(matrix1, matrix2,

productMatrix);

printf("Matrix 1:\n");

displayMatrix(matrix1);

printf("Matrix 2:\n");

displayMatrix(matrix2);

printf("Sum of Matrices:\n");

displayMatrix(sumMatrix);

printf("Product of Matrices:\n");

displayMatrix(productMatrix);

return0;

}

Question 4

#include<stdio.h>

#defineSIZE3

voidprintSumDiagonal(intmatrix[SIZE][SIZE]) {

int sum =0;

for (inti=0; i<SIZE; ++i) {

sum +=matrix[i][i];

}

printf("Sum of diagonal elements: %d\n", sum);

}

voidprintUpperTriangular(intmatrix[SIZE][SIZE]) {

printf("Upper triangular matrix:\n");

for (inti=0; i<SIZE; ++i) {

for (int j =0; j <SIZE; ++j) {

if (i<= j) {

printf("%d\t", matrix[i][j]);

} else {

printf("0\t");

}

}

printf("\n");

}

}

voidprintLowerTriangular(intmatrix[SIZE][SIZE]) {

printf("Lower triangular matrix:\n");

for (inti=0; i<SIZE; ++i) {

for (int j =0; j <SIZE; ++j) {

if (i>= j) {

printf("%d\t", matrix[i][j]);

} else {

printf("0\t");

}

}

printf("\n");

}

}

intmain() {

intmatrix[SIZE][SIZE] = {{1, 2, 3},

{4, 5, 6},

{7, 8, 9}};

printSumDiagonal(matrix);

printUpperTriangular(matrix);

printLowerTriangular(matrix);

return0;

}

Question 5

#include<stdio.h>

#defineROWS3

#defineCOLS3

voidfindFrequency(intmatrix[ROWS][COLS]) {

intoddCount=0, evenCount=0;

for (inti=0; i<ROWS; ++i) {

for (int j =0; j <COLS; ++j) {

if (matrix[i][j] %2==0) {

evenCount++;

} else {

oddCount++;

}

}

}

printf("Frequency of odd elements: %d\n",

oddCount);

printf("Frequency of even elements: %d\n",

evenCount);

}

intmain() {

intmatrix[ROWS][COLS] = {{1, 2, 3},

{4, 5, 6},

{7, 8, 9}};

findFrequency(matrix);

return0;

}

Question 6

#include<stdio.h>

#defineROWS3

#defineCOLS3

voidfindRowSum(int matrix[ROWS][COLS]) {

printf("Sum of each row:\n");

for (inti=0; i<ROWS; ++i) {

introwSum=0;

for (int j =0; j <COLS; ++j) {

rowSum+= matrix[i][j];

}

printf("Row %d: %d\n", i+1, rowSum);

}

}

voidfindColumnSum(int matrix[ROWS][COLS]) {

printf("\nSum of each column:\n");

for (int j =0; j <COLS; ++j) {

intcolSum=0;

for (inti=0; i<ROWS; ++i) {

colSum+= matrix[i][j];

}

printf("Column %d: %d\n", j +1, colSum);

}

}

intmain() {

intmatrix[ROWS][COLS] = {{1, 2, 3},

{4, 5, 6},

{7, 8, 9}};

findRowSum(matrix);

findColumnSum(matrix);

return0;

}

Question 7

#include<stdio.h>

intmain() {

// Initialize a 3x3 matrix

intmatrix[3][3] = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

// Print the initialized matrix

printf("Initialized 3x3 Matrix:\n");

for (inti=0; i<3; ++i) {

for (int j =0; j <3; ++j) {

printf("%d\t", matrix[i][j]);

}

printf("\n");

}

return0;

}

Question 8

#include<stdio.h>

#defineSIZE3

voidcheckSpecialMatrix(int matrix[SIZE][SIZE]) {

intisDiagonal=1, isUpperTriangular=1,

isLowerTriangular=1;

for (inti=0; i<SIZE; ++i) {

for (int j =0; j <SIZE; ++j) {

if (i!= j && matrix[i][j] !=0) {

isDiagonal=0;

}

if (i> j && matrix[i][j] !=0) {

isUpperTriangular=0;

}

if (i< j && matrix[i][j] !=0) {

isLowerTriangular=0;

}

}

}

if (isDiagonal) {

printf("The matrix is a diagonal

matrix.\n");

} elseif (isUpperTriangular) {

printf("The matrix is an upper triangular

matrix.\n");

} elseif (isLowerTriangular) {

printf("The matrix is a lower triangular

matrix.\n");

} else {

printf("The matrix is not a special

matrix.\n");

}

}

intmain() {

intmatrix[SIZE][SIZE];

printf("Enter the elements of the %dx%d

matrix:\n", SIZE, SIZE);

for (inti=0; i<SIZE; ++i) {

for (int j =0; j <SIZE; ++j) {

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

}

}

checkSpecialMatrix(matrix);

return0;

}

Question 9

#include<stdio.h>

#defineROWS3

#defineCOLS3

intisSparseMatrix(intmatrix[ROWS][COLS]) {

intzeroCount=0, nonZeroCount=0;

for (inti=0; i<ROWS; ++i) {

for (int j =0; j <COLS; ++j) {

if (matrix[i][j] ==0) {

zeroCount++;

} else {

nonZeroCount++;

}

}

}

if (zeroCount> (ROWS\*COLS) /2) {

return1;

} else {

return0;

}

}

voidmain() {

intmatrix[ROWS][COLS];

inti, j;

printf("Enter the elements of the %dx%d

matrix:\n", ROWS, COLS);

for (i=0; i<ROWS; ++i) {

for (j =0; j <COLS; ++j) {

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

}

}

if (isSparseMatrix(matrix)) {

printf("The matrix is a sparse matrix.\n");

} else {

printf("The matrix is not a sparse

matrix.\n");

}

}

***Week-8***

Question1

#include<stdio.h>

intmain() {

int number =10;

int\*ptr=&number;

printf("Value of number: %d\n", number);

printf("Value pointed to by ptr: %d\n", \*ptr);

\*ptr=20;

printf("Updated value of number: %d\n", number);

doubledoubleNumber=3.14;

double\*doublePtr=&doubleNumber;

printf("Value of doubleNumber: %lf\n", doubleNumber);

printf("Value pointed to by doublePtr: %lf\n", \*doublePtr);

return0;

}

Question 2

#include<stdio.h>

voidaddNumbers(int\*num1, int\*num2, int\*sum) {

\*sum =\*num1 +\*num2;

}

intmain() {

int number1, number2, result;

printf("Enter first number: ");

scanf("%d", &number1);

printf("Enter second number: ");

scanf("%d", &number2);

addNumbers(&number1, &number2, &result);

printf("Sum of %d and %d is: %d\n", number1, number2, result);

return0;

}

Question-3

#include<stdio.h>

voidswapNumbers(int\*num1, int\*num2) {

int temp =\*num1;

\*num1 =\*num2;

\*num2 =temp;

}

intmain() {

int number1, number2;

printf("Enter first number: ");

scanf("%d", &number1);

printf("Enter second number: ");

scanf("%d", &number2);

printf("Before swapping: \n");

printf("First number: %d\n", number1);

printf("Second number: %d\n", number2);

swapNumbers(&number1, &number2);

printf("After swapping: \n");

printf("First number: %d\n", number1);

printf("Second number: %d\n", number2);

return0;

}

Question 4

#include<stdio.h>

voidinputArray(int\*arr, intsize) {

printf("Enter %d elements:\n", size);

for (inti=0; i< size; ++i) {

scanf("%d", arr+i);}

}

voidprintArray(int\*arr, intsize) {

printf("Array elements are:\n");

for (inti=0; i< size; ++i) {

printf("%d ", \*(arr+i));

}

printf("\n");

}

intmain() {

intsize;

printf("Enter the size of the array: ");

scanf("%d", &size);

int array[size];

inputArray(array, size);

printArray(array, size);

return0;

}

Question-5

#include<stdio.h>

voidcopyArray(int\*source, int\*destination, intsize) {

for (inti=0; i<size; ++i) {

\*(destination+i) =\*(source+i);

}

}

voidprintArray(int\*arr, intsize) {

printf("Array elements are:\n");

for (inti=0; i<size; ++i) {

printf("%d ", \*(arr+i));

}

printf("\n");

}

intmain() {

intsize;

printf("Enter the size of the array: ");

scanf("%d", &size);

intsourceArray[size];

intdestinationArray[size];

printf("Enter %d elements for the source array:\n", size);

for (inti=0; i< size; ++i) {

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

}

copyArray(sourceArray, destinationArray, size);

printf("\nSource Array:\n");

printArray(sourceArray, size);

printf("\nDestination Array (copied from source array):\n");

printArray(destinationArray, size);

return0;

}

Question-6

#include<stdio.h>

voidswapArrays(int\*arr1, int\*arr2, intsize) {

int temp[size];

for (inti=0; i< size; ++i) {

temp[i] =\*(arr1 +i);

}

for (inti=0; i< size; ++i) {

\*(arr1 +i) =\*(arr2 +i);

}

for (inti=0; i< size; ++i) {

\*(arr2 +i) = temp[i];

}

}

voidprintArray(int\*arr, intsize) {

printf("Array elements are:\n");

for (inti=0; i< size; ++i) {

printf("%d ", \*(arr+i));

}

printf("\n");

}

intmain() {

intsize;

printf("Enter the size of the arrays: ");

scanf("%d", &size);

int array1[size];

int array2[size];

printf("Enter %d elements for the first array:\n", size);

for (inti=0; i< size; ++i) {

scanf("%d", &array1[i]);

}

printf("Enter %d elements for the second array:\n", size);

for (inti=0; i< size; ++i) {

scanf("%d", &array2[i]);

}

printf("\nArrays before swapping:\n");

printf("Array 1:\n");

printArray(array1, size);

printf("Array 2:\n");

printArray(array2, size);

swapArrays(array1, array2, size);

printf("\nArrays after swapping:\n");

printf("Array 1 (swapped):\n");

printArray(array1, size);

printf("Array 2 (swapped):\n");

printArray(array2, size);

return0;}

Question-7

#include<stdio.h>

voidreverseArray(int\*arr, intsize) {

int\*start =arr;

int\*end =arr+ size -1;

while (start < end) {

int temp =\*start;

\*start =\*end;

\*end =temp;

start++;

end--;

}

}

voidprintArray(int\*arr, intsize) {

printf("Array elements are:\n");

for (inti=0; i< size; ++i) {

printf("%d ", \*(arr+i));

}

printf("\n");

}

intmain() {

intsize;

printf("Enter the size of the array: ");

scanf("%d", &size);

int array[size];

printf("Enter %d elements for the array:\n", size);

for (inti=0; i< size; ++i) {

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

}

printf("\nOriginal Array:\n");

printArray(array, size);s

reverseArray(array, size);

printf("\nArray after reversing:\n");

printArray(array, size);

return0;

}

Question 8

#include<stdio.h>

voidaddMatrices(int\*mat1, int\*mat2, int\*result, introws, intcols) {

for (inti=0; i< rows; ++i) {

for (int j =0; j < cols; ++j) {

\*(result +i\* cols + j) =\*(mat1 +i\* cols + j) +\*(mat2 +i\* cols +

j);

}

}

}

voidprintMatrix(int\*mat, introws, intcols) {

printf("Matrix elements are:\n");

for (inti=0; i< rows; ++i) {

for (int j =0; j < cols; ++j) {

printf("%d ", \*(mat +i\* cols + j));

}

printf("\n");

}

}

intmain() {

int rows, cols;

printf("Enter the number of rows: ");

scanf("%d", &rows);

printf("Enter the number of columns: ");

scanf("%d", &cols);

int matrix1[rows][cols];

int matrix2[rows][cols];

intresultMatrix[rows][cols];

printf("Enter elements for the first matrix:\n");

for (inti=0; i< rows; ++i) {

for (int j =0; j < cols; ++j) {

scanf("%d", &matrix1[i][j]);

}

}

printf("Enter elements for the second matrix:\n");

for (inti=0; i< rows; ++i) {

for (int j =0; j < cols; ++j) {

scanf("%d", &matrix2[i][j]);

}

}

addMatrices(&matrix1[0][0], &matrix2[0][0], &resultMatrix[0][0], rows,

cols);

printf("\nMatrix 1:\n");

printMatrix(&matrix1[0][0], rows, cols);

printf("\nMatrix 2:\n");

printMatrix(&matrix2[0][0], rows, cols);

printf("\nResult Matrix (Matrix 1 + Matrix 2):\n");

printMatrix(&resultMatrix[0][0], rows, cols);

return0;

}

Question 9

#include<stdio.h>

voidmultiplyMatrices(int\*mat1, int\*mat2, int\*result, introws1, intcols1,

intcols2) {

for (inti=0; i< rows1; ++i) {

for (int j =0; j < cols2; ++j) {

\*(result +i\* cols2 + j) =0;

for (int k =0; k < cols1; ++k) {

\*(result +i\* cols2 + j) +=\*(mat1 +i\* cols1 + k) \*\*(mat2 + k \*

cols2 + j);

}

}

}

}

voidprintMatrix(int\*mat, introws, intcols) {

printf("Matrix elements are:\n");

for (inti=0; i< rows; ++i) {

for (int j =0; j < cols; ++j) {

printf("%d ", \*(mat +i\* cols + j));

}

printf("\n");

}

}

intmain() {

int rows1, cols1, rows2, cols2;

printf("Enter the number of rows for matrix 1: ");

scanf("%d", &rows1);

printf("Enter the number of columns for matrix 1: ");

scanf("%d", &cols1);

printf("Enter the number of rows for matrix 2: ");

scanf("%d", &rows2);

printf("Enter the number of columns for matrix 2: ");

scanf("%d", &cols2);

if (cols1 != rows2) {

printf("Error: The number of columns in matrix 1 must be equal to the

number of rows in matrix 2 for multiplication.\n");

return1;

}

int matrix1[rows1][cols1];

int matrix2[rows2][cols2];

intresultMatrix[rows1][cols2];

printf("Enter elements for matrix 1:\n");

for (inti=0; i< rows1; ++i) {

for (int j =0; j < cols1; ++j) {

scanf("%d", &matrix1[i][j]);

}

}

printf("Enter elements for matrix 2:\n");

for (inti=0; i< rows2; ++i) {

for (int j =0; j < cols2; ++j) {

scanf("%d", &matrix2[i][j]);

}

}

multiplyMatrices(&matrix1[0][0], &matrix2[0][0], &resultMatrix[0][0],

rows1, cols1, cols2);

printf("\nMatrix 1:\n");

printMatrix(&matrix1[0][0], rows1, cols1);

printf("\nMatrix 2:\n");

printMatrix(&matrix2[0][0], rows2, cols2);

printf("\nResult Matrix (Matrix 1 \* Matrix 2):\n");

printMatrix(&resultMatrix[0][0], rows1, cols2);

return0;

}

***Week 9***

Question 1

#include<stdio.h>

intmain() {

charmainString[100], string[50];

inti, j, found;

printf("Enter the main string: ");

gets(mainString);

printf("Enter the substring to search: ");

gets(string);

for (i=0; mainString[i] !='\0'; ++i) {

found =1;

for (j =0; string[j] !='\0'; ++j) {

if (mainString[i+ j] != string[j]) {

found =0;

break;

}

}

if (found) {

printf("string found at position %d.\n", i);

return0;

}

}

printf("string not found in the main string.\n");

return0;

}

Question 2

#include<stdio.h>

#include<string.h>

#defineMAX\_SIZE100

voidreverseWords(char sentence[MAX\_SIZE]);

intmain() {

charsentence[MAX\_SIZE];

printf("Enter a sentence: ");

gets(sentence);

reverseWords(sentence);

printf("Reversed sentence: %s\n", sentence);

return0;

}

voidreverseWords(charsentence[MAX\_SIZE]) {

int start, end, length;

length =strlen(sentence);

for (start =0, end = length -1; start < end; ++start, --end) {

char temp =sentence[start];

sentence[start] =sentence[end];

sentence[end] =temp;

}

start =0;

for (end =0; end <= length; ++end) {

if (sentence[end] ==' '||sentence[end] =='\0') {

intwordStart, wordEnd;

wordStart=start;

wordEnd= end -1;

while (wordStart<wordEnd) {

char temp =sentence[wordStart];

sentence[wordStart] =sentence[wordEnd];

sentence[wordEnd] =temp;

++wordStart;

--wordEnd;

}

start = end +1;

}

}

}

Question 3

#include<stdio.h>

intmain() {

charinputString[1000];

int vowels =0, consonants =0, digits =0, spaces =0, other =0;

printf("Enter a string: ");

gets(inputString);

for (inti=0; inputString[i] !='\0'; ++i) {

charcurrentChar=inputString[i];

if ((currentChar>='a'&&currentChar<='z') ||

(currentChar>='A'&&currentChar<='Z')) {

if

(currentChar=='a'||currentChar=='e'||currentChar=='i'||currentChar=='o'||curre

ntChar=='u'||

currentChar=='A'||currentChar=='E'||currentChar=='I'||currentC

har=='O'||currentChar=='U') {

++vowels;

} else {

++consonants;

}

} elseif (currentChar>='0'&&currentChar<='9') {

++digits;

} elseif (currentChar==' '||currentChar=='\t'||currentChar=='\n') {

++spaces;

} else {

++other;

}

}

printf("Vowels: %d\n", vowels);

printf("Consonants: %d\n", consonants);

printf("Digits: %d\n", digits);

printf("Spaces: %d\n", spaces);

printf("Other characters: %d\n", other);

return0;

}

Question 4

#include<stdio.h>

intmain() {

charinputString[1000];

printf("Enter a string: ");

gets(inputString);

printf("Separated characters: ");

for (inti=0; inputString[i] !='\0'; ++i) {

printf("%c ", inputString[i]);

}

return0;

}

Question 5

#include<stdio.h>

#include<string.h>

#defineMAX\_SIZE100

intmain() {

charfirstString[MAX\_SIZE], secondString[MAX\_SIZE];

printf("Enter the first string: ");

gets(firstString);

printf("Enter the second string: ");

gets(secondString);

strcat(firstString, " ");

strcat(firstString, secondString);

printf("Concatenated string: %s\n", firstString);

return0;

}

Question 6

#include<stdio.h>

#include<string.h>

#defineMAX\_SIZE100

intmain() {

charinputString[MAX\_SIZE];

printf("Enter a string: ");

gets(inputString);

for (inti=0; i<strlen(inputString); ++i) {

if (islower(inputString[i])) {

inputString[i] =toupper(inputString[i]);

} elseif (isupper(inputString[i])) {

inputString[i] =tolower(inputString[i]);

}

}

printf("Toggled case string: %s\n", inputString);

return0;

}

Question 7

#include<stdio.h>

#defineMAX\_SIZE100

intareIdentical(char str1[MAX\_SIZE], char str2[MAX\_SIZE]);

intmain() {

charfirstString[MAX\_SIZE], secondString[MAX\_SIZE];

printf("Enter the first string: ");

gets(firstString);

printf("Enter the second string: ");

gets(secondString);

if (areIdentical(firstString, secondString)) {

printf("Identical\n");

} else {

printf("Not Identical\n");

}

return0;

}

intareIdentical(charstr1[MAX\_SIZE], charstr2[MAX\_SIZE]) {

inti=0;

while (str1[i] !='\0'&&str2[i] !='\0') {

if (str1[i] !=str2[i]) {

return0;

}

++i;

}

if (str1[i] !=str2[i]) {

return0;

}

return1;

}

Question 8

#include<stdio.h>

#include<string.h>

#defineMAX\_STUDENTS100

#defineMAX\_NAME\_LENGTH50

voidswap(chara[], charb[]) {

chartemp[MAX\_NAME\_LENGTH];

strcpy(temp, a);

strcpy(a, b);

strcpy(b, temp);

}

voidbubbleSort(charnames[][MAX\_NAME\_LENGTH], intn) {

for (inti=0; i<n-1; ++i) {

for (int j =0; j <n-i-1; ++j) {

if (strcmp(names[j], names[j +1]) >0) {

swap(names[j], names[j +1]);

}

}

}

}

intmain() {

intnumStudents;

printf("Enter the number of students: ");

scanf("%d", &numStudents);

if (numStudents<=0||numStudents>MAX\_STUDENTS) {

printf("Invalid number of students.

Exiting.\n");

return1;

}

charstudentNames[MAX\_STUDENTS][MAX\_NAME\_LENGTH];

for (inti=0; i<numStudents; ++i) {

printf("Enter the name of student %d: ", i+1);

scanf("%s", studentNames[i]);

}

bubbleSort(studentNames, numStudents);

printf("\nSorted List of Student Names:\n");

for (inti=0; i<numStudents; ++i) {

printf("%s\n", studentNames[i]);

}

return0;

}

9.Write a C program to multiply two matrix using pointers.

#include <stdio.h>

#include <stdlib.h>

int main() {

int n, m, p;

printf("Enter the number of rows in the first matrix: ");

scanf("%d", &n);

printf("Enter the number of columns in the first matrix (and rows in the second

matrix): ");

scanf("%d", &m);

printf("Enter the number of columns in the second matrix: ");

scanf("%d", &p);

int \*A = (int\*)malloc(n \* m \* sizeof(int));

int \*B = (int\*)malloc(m \* p \* sizeof(int));

int \*C = (int\*)malloc(n \* p \* sizeof(int));

if (!A || !B || !C) {

printf("Error: Memory allocation failed.\n");

exit(1);

}

printf("Enter elements of the first matrix:\n");

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

for (int j = 0; j < m; ++j) {

printf("Enter element [%d][%d]: ", i + 1, j + 1);

scanf("%d", A + i \* m + j);

}

}

printf("Enter elements of the second matrix:\n");

for (int i = 0; i< m; ++i) {

for (int j = 0; j < p; ++j) {

printf("Enter element [%d][%d]: ", i + 1, j + 1);

scanf("%d", B + i \* p + j);

}

}

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

for (int j = 0; j < p; ++j) {

int sum = 0;

for (int k = 0; k < m; ++k) {

sum += \*(A + i \* m + k) \* \*(B + k \* p + j);

}

\*(C + i \* p + j) = sum;

}

}

printf("Resultant matrix:\n");

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

for (int j = 0; j < p; ++j) {

printf("%d ", \*(C + i \* p + j));

}

printf("\n");

}

free(A);

free(B);

free(C);

return 0;

}

***Week 10***

1. Write a C program to find length of string using pointers.

#include <stdio.h>

int strlen(const char \*str) {

int l = 0;

while (\*str != '\0') {

l++;

str++;

}

return l;

}

int main(){

char a[100];

printf("Enter a string: ");

scanf("%s",a);

int l= strlen(a);

printf("Length of the string: %d\n",l);

return 0;

}

2. Write a C program to copy one string to another using pointer.

#include <stdio.h>

void copyString(char \*dest, const char \*src) {

while ((\*dest++ = \*src++) != '\0');

}

int main() {

char str[100],newstr[100];

printf("Enter the source string: ");

scanf("%s",str);

copyString(newstr,str);

printf("Copied string: %s\n",newstr);

return 0;

}

3.Write a C program to concatenate two strings using pointers

#include <stdio.h>

void concatenateStrings(char \*dest, const char

\*src) {

while (\*dest != '\0') {

dest++;

}

while ((\*dest++ = \*src++) != '\0');

}

int main() {

char firststr[100], secondstr[100];

printf("Enter the first string: ");

scanf("%s", firststr);

printf("Enter the second string: ");

scanf("%s", secondstr);

concatenateStrings(firststr, secondstr);

printf("Concatenated string: %s\n",firststr);

return 0;

}

4.Write a C program to compare two strings using pointers.

#include <stdio.h>

int cmpstr(const char \*str1, const char \*str2) {

while (\*str1 != '\0' && \*str2 != '\0') {

if (\*str1 != \*str2) {

return 0;

}

str1++;

str2++;

}

return (\*str1 == '\0' && \*str2 == '\0');

}

int main() {

char firstStr[100], secondStr[100];

printf("Enter the first string: ");

scanf("%s", firstStr);

printf("Enter the second string: ");

scanf("%s", secondStr);

if (cmpstr(firstStr, secondStr)) {

printf("The strings are equal.\n");

} else {

printf("The strings are not equal.\n");

}

return 0;

}

5. WAP to find largest among three numbers using pointer.

#include <stdio.h>

int findLargest(int \*n1, int \*n2, int \*n3) {

int l = \*n1;

if (\*n2 > l) {

l = \*n2;

}

if (\*n3 > l) {

l = \*n3;

}

return l;

}

int main() {

int n1,n2,n3;

printf("Enter the first number: ");

scanf("%d", &n1);

printf("Enter the second number: ");

scanf("%d", &n2);

printf("Enter the third number: ");

scanf("%d", &n3);

int l = findLargest(&n1, &n2, &n3);

printf("The largest number is: %d\n", l);

return 0;

}

6. WAP to find largest among three numbers using pointer.

#include <stdio.h>

int findLargest(int \*n1, int \*n2, int \*n3) {

int l = \*n1;

if (\*n2 > l) {

l = \*n2;

}

if (\*n3 > l) {

l = \*n3;

}

return l;

}

int main() {

int n1,n2,n3;

printf("Enter the first number: ");

scanf("%d", &n1);

printf("Enter the second number: ");

scanf("%d", &n2);

printf("Enter the third number: ");

scanf("%d", &n3);

int l = findLargest(&n1, &n2, &n3);

printf("The largest number is: %d\n", l);

return 0;

}

7. WAP to find factorial of a number using pointer.

#include <stdio.h>

long longFact(int \*n) {

long long f = 1;

for (int i = 1; i<= \*n; i++) {

f \*= i;

}

return f;

}

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

long long f = Fact(&n);

printf("Factorial of %d is: %lld\n", n, f);

return 0;

}

8.Write a program to print largest even number present in an array using

pointer to an array.

#include <stdio.h>

int findLargestEven(int \*a, int s) {

int lEven = -1;

for (int i = 0; i< s; i++) {

if (a[i] % 2 == 0 && a[i] >lEven) {

lEven = a[i];

}

}

return lEven;

}

int main() {

int s;

printf("Enter the size of the array: ");

scanf("%d", &s);

int n[s];

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

for (int i = 0; i< s; i++) {

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

}

int lEven = findLargestEven(n, s);

if (lEven != -1) {

printf("The largest even number is: %d\n", lEven);

} else {

printf("No even numbers found in the array.\n");

}

return 0;

}

9.WAP to find sum of elements of an array using array of pointer.

#include <stdio.h>

int findArraySum(int \*a[], int s) {

int sum = 0;

for (int i = 0; i< s; i++) {

sum += \*a[i];

}

return sum;

}

int main() {

int s;

printf("Enter the size of the array: ");

scanf("%d", &s);

int n[s];

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

for (int i = 0; i< s; i++) {

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

}

int \*ps[s];

for (int i = 0; i< s; i++) {

ps[i] = &n[i];

}

int sum = findArraySum(ps, s);

printf("Sum of elements in the array: %d\n", sum);

return 0;

}

10. WAP to compute simple interest using pointers.

#include <stdio.h>

float CSI(float \*p, float \*r, float \*t) {

return (\*p \* \*r \* \*t) / 100.0;

}

int main() {

float p, r, t;

printf("Enter principal amount: ");

scanf("%f", &p);

printf("Enter rate of interest: ");

scanf("%f", &r);

printf("Enter time in years: ");

scanf("%f", &t);

float i = CSI(&p, &r, &t);

printf("Simple Interest: %.2f\n", i);

return 0;

}

11.Write a program to print largest even number present in an array using

pointer to an array.

#include <stdio.h>

int findLargestEven(int \*a, int s) {

int lEven = -1;

for (int i = 0; i< s; i++) {

if (a[i] % 2 == 0 && a[i] >lEven) {

lEven = a[i];

}

}

return lEven;

}

int main() {

int s;

printf("Enter the size of the array: ");

scanf("%d", &s);

int n[s];

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

for (int i = 0; i< s; i++) {

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

}

int lEven = findLargestEven(n, s);

if (lEven != -1) {

printf("The largest even number is: %d\n", lEven);

} else {

printf("No even numbers found in the array.\n");

}

return 0;

}

***Week 11***

1.Write a C function to return the maximum of three

integers.

#include <stdio.h>

int findMaximum(int num1, int num2, int num3) {

int max = num1;

if (num2 > max) {

max = num2;

}

if (num3 > max) {

max = num3;

}

return max;

}

int main() {

int num1, num2, num3;

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

printf("Enter the third number: ");

scanf("%d", &num3);

int maximum = findMaximum(num1, num2, num3);

printf("The maximum number is: %d\n", maximum);

return 0;

}

2.Write a C function to check if a given number is prime or

not.

#include <stdio.h>

int isPrime(int n) {

if (n <= 1) {

return 0;

}

for (int i = 2; i \* i<= n; i++) {

if (n % i == 0) {

return 0;

}

}

return 1;

}

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

if (isPrime(n)) {

printf("%d is a prime number.\n", n);

} else {

printf("%d is not a prime number.\n", n);

}

return 0;

}

3.Write a C function to compute the factorial of a nonnegative integer.

#include <stdio.h>

unsigned long longfactorial(int n) {

if (n < 0) {

return 0;

}

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

return 1;

}

unsigned long long r = 1;

for (int i = 2; i<= n; i++) {

r \*= i;

}

return r;

}

int main() {

int n;

printf("Enter a non-negative integer: ");

scanf("%d", &n);

unsigned long long r = factorial(n);

printf("The factorial of %d is: %llu\n", n, r);

return 0;

}

4.Write a C function to swap the values of two integers in

actual arguments.

#include <stdio.h>

void swapIntegers(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int num1, num2;

printf("Enter the first integer: ");

scanf("%d", &num1);

printf("Enter the second integer: ");

scanf("%d", &num2);

swapIntegers(&num1, &num2);

printf("After swapping:\n");

printf("First integer: %d\n", num1);

printf("Second integer: %d\n", num2);

return 0;

}

5.Write a C function to compute the sum and average of an

array of integers.

#include <stdio.h>

void computeSumAndAverage(int \*arr, int size, int \*sum, float \*average) {

\*sum = 0;

for (int i = 0; i< size; i++) {

\*sum += \*(arr + i);

}

\*average = (float)(\*sum) / size;

}

int main() {

int size;

printf("Enter the size of the array: ");

scanf("%d", &size);

int numbers[size];

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

for (int i = 0; i< size; i++) {

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

}

int sum;

float average;

computeSumAndAverage(numbers, size, &sum, &average);

printf("Sum of the array elements: %d\n", sum);

printf("Average of the array elements: %.2f\n", average);

return 0;

}

6.Write a C function to find the GCD (Greatest Common

Divisor) of two nonnegative integers using Euclid's algorithm.

#include <stdio.h>

int findGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

int main() {

int num1, num2;

printf("Enter the first non-negative integer: ");

scanf("%d", &num1);

printf("Enter the second non-negative integer: ");

scanf("%d", &num2);

int gcd = findGCD(num1, num2);

printf("The GCD of %d and %d is: %d\n", num1, num2, gcd);

return 0;

}

7.Write a C function to check if a given string is a valid

palindrome, considering only alphanumeric characters and

ignoring cases.

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int isPalindrome(const char \*str) {

int length = strlen(str);

int start = 0;

int end = length - 1;

while (start < end) {

while (!isalnum(str[start]) && start < end) {

start++;

}

while (!isalnum(str[end]) && start < end) {

end--;

}

char char1 = tolower(str[start]);

char char2 = tolower(str[end]);

if (char1 != char2) {

return 0;

}

start++;

end--;

}

return 1;

}

int main() {

char input[100];

printf("Enter a string: ");

fgets(input, sizeof(input), stdin);

input[strcspn(input, "\n")] = '\0';

if (isPalindrome(input)) {

printf("The string is a valid palindrome.\n");

} else {

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

}

return 0;

}

8.Write a C function to calculate the sum and difference of

two complex numbers.

#include <stdio.h>

typedef struct {

float real;

float imaginary;

} ComplexNumber;

void addComplex(ComplexNumber num1, ComplexNumber num2,

ComplexNumber \*result) {

result->real = num1.real + num2.real;

result->imaginary = num1.imaginary + num2.imaginary;

}

void subtractComplex(ComplexNumber num1, ComplexNumber num2,

ComplexNumber \*result) {

result->real = num1.real - num2.real;

result->imaginary = num1.imaginary - num2.imaginary;

}

int main() {

ComplexNumber complex1, complex2, sum, difference;

printf("Enter the real part of the first complex number: ");

scanf("%f", &complex1.real);

printf("Enter the imaginary part of the first complex number: ");

scanf("%f", &complex1.imaginary);

printf("Enter the real part of the second complex number: ");

scanf("%f", &complex2.real);

printf("Enter the imaginary part of the second complex number: ");

scanf("%f", &complex2.imaginary);

addComplex(complex1, complex2, &sum);

subtractComplex(complex1, complex2, &difference);

printf("Sum: %.2f + %.2fi\n", sum.real, sum.imaginary);

printf("Difference: %.2f + %.2fi\n", difference.real, difference.imaginary);

return 0;

}

9.Write a C function to find the second largest and second

smallest elements in an array of integers.

#include <stdio.h>

void findSecondLargestAndSmallest(int arr[], int size, int \*secondLargest, int

\*secondSmallest) {

if (size < 2) {

printf("Array should have at least two elements.\n");

return;

}

\*secondLargest = (arr[0] >arr[1]) ? arr[0] : arr[1];

\*secondSmallest = (arr[0] <arr[1]) ? arr[0] : arr[1];

for (int i = 2; i< size; i++) {

if (arr[i] > \*secondLargest) {

\*secondLargest = arr[i];

} else if (arr[i] < \*secondSmallest) {

\*secondSmallest = arr[i];

}

}

}

int main() {

int size;

printf("Enter the size of the array: ");

scanf("%d", &size);

if (size <= 0) {

printf("Array size should be greater than 0.\n");

return 1;

}

int numbers[size];

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

for (int i = 0; i< size; i++) {

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

}

int secondLargest, secondSmallest;

findSecondLargestAndSmallest(numbers, size, &secondLargest,

&secondSmallest);

printf("Second Largest Element: %d\n", secondLargest);

printf("Second Smallest Element: %d\n", secondSmallest);

return 0;

}

10.Write a C function to find the number of occurrences of

each unique element in an array.

#include <stdio.h>

void countOccurrences(int arr[], int size) {

int frequency[size];

for (int i = 0; i< size; i++) {

frequency[i] = 0;

}

for (int i = 0; i< size; i++) {

int currentElement = arr[i];

int isEncountered = 0;

for (int j = 0; j <i; j++) {

if (arr[j] == currentElement) {

isEncountered = 1;

break;

}

}

if (!isEncountered) {

int count = 1;

for (int j = i + 1; j < size; j++) {

if (arr[j] == currentElement) {

count++;

}

}

printf("Element %d occurs %d times\n", currentElement, count);

}

}

}

int main() {

int size;

printf("Enter the size of the array: ");

scanf("%d", &size);

if (size <= 0) {

printf("Array size should be greater than 0.\n");

return 1;

}

int numbers[size];

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

for (int i = 0; i< size; i++) {

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

}

countOccurrences(numbers, size);

return 0;