C- Programming Assignment

Q.1 Perform addition, subtraction and multiplication operations on two matrices.

→

```
#include <stdio.h>

int main() {
int a[10][10], b[10][10], c[10][10];
int i,j,k,n;

Printf ("enter size of square matrices : ");
scanf ("%d", &n);

Printf ("Enter elements of first matrices : \n");
for (i=0; i<n; i++)
for (j=0 ; j<n ; j++)
scanf ("%d", & a[i][j];

Printf ("Enter elements of second matrix :\n");
for (i=0; i<n ; i++)
for (j=0; j<n ; j++)
scanf ("%d", & b[i][j]);

Printf ("\n subtraction of matrices : \n");
for (i=0 ; i<n ; i++){
for (j=0 ; j<n ; j++) {
c[i][j] = a[i][j] + b[i][j];

Printf ("%d", c[i][j]);
}
Printf ("\n");
}
```

vision

```
Printf ("\n subtraction of matrices : \n");
for (i=0, i<n , i++) {
for (j=0 ; j<n ; j++ ){
( [i][j] = a[i][j] - b[i][j];

Printf ("%d", ([i][j]);
}
Printf ("\n");
}

Printf ("\n multiplication of matrices : \n");
for (i=0; i<n ; i++) {
for (j=0; j<n ; j++) {
( [i][j] =0;
for (k=0; k<n ; k++)
([i] [j] + = a [i][k] * b [k][j];

Printf ("%d", ([i][j]);
}
Printf (" \n");
}
return 0;
}
```

→ Input:

Enter size of square matrices: 2

Enter elementeas of first matrix:

1 2

3 4

Enter elements of second matrix:

6 5

7 8

→ Output:

Addition of matrices:

6 8

10 12

Subtraction of matrices:

-4 -4

-4 -4

Multiplication of matrices:

19 22

43 50

Q.2 Sort all the elements of a 4x4 matrices and store the result in a single dimension array.

→

```
#include <stdio.h>

int main() {
int a[4][4], arr[16];
int i, j, k, temp;

Printf ("Enter elements of 4x4 matrix : \n");
for (i=0; i<4; i++)
for (j=0; j<4; j++)
arr[k++] = a[i][j];

for (i=0; i<16; i++) {
for (j=i+1; j<16; j++) {
if (arr[i] > arr[j]) {
 temp = arr[i];
  arr[i] = arr[j];
  arr[j] = temp;
}
```

```
Printf ("\n sorted elements in 1D array: \n");
for (i=0; i<16; i++)
Printf ("%d", arr[i]);

return 0;

}
```

→ Input:

1	9	5	2
7	4	8	6
3	12	10	11
15	14	13	16

→ output:

sorted elements in 1D array:

1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16

Q.3 Print the largest and smallest numbers from a 3x3 matrix using pointers:

→

```
#include <stdio.h>

int main () {
int a[3][3];
int i, j;
int *p;
```

```
int max, min;
Printf ("Enter 3x3 matrix elements: \n");
for (i=0; i<3; i++)
for (j=0; j<3; j++)
scanf ("%d", &a[i][j]);

P = &a[0][0];

max = min = *P;

for (i=0; i<9; i++) {
if (*(p+i) > max)
  max = *(p+i);
if (*p+i) <min)
 min = *(p+i);
}

Printf ("\n largest element = %d", max);
Printf ("\n smallest element = %d", min);

return 0;

}
```

→ Input:

1 5 9
3 2 8
4 6 7

→ output:

largest element = 9
smallest element = 1

vision

Q.4 Accept and print later on three book names using array of pointers.

→
```
# include <stdio.h>
int main () {
char * ptr [3];
int i;

Printf (" Enter names of 3 books : \n");
for (i=0; i<3; i++) {
Printf ( " Book %d : ", i+1);
scanf (" % [^\n]" book [i]);

Ptr [i] = book [i];

}

Printf ( "\n. you entered these boole names : \n");

for (i=0; i<3; i++)
Printf ("%s \n "z, Ptr [i]);

return 0;

}
```

→ Input:

Book 1 : The Alchemist
Book 2 : Harry Potter
Book 3 : Pride and Prejudice

→ Output:

You entered these book names:
The Alchemist
Harry Potter
Pride and Prejudice

Q.5 Write a program that takes a set of names of individuals and abbreviates the first, middle and other names except the last name by their first letter.

→
```
# include <stdio.h>
int main () {
 char name [100];
 int i , len;

Printf (" Enter full name : ");
 gets (name);

len = strlen (name);

Printf (" Abbreviated name : ");
Printf (" %.c", name [0]};

for (i=0; i<len ; i++) {
 if (name [i] == ' ' &&
 name [i+1] != '\0') {

int j . flag = 0;
 for (j= i+1 ; j<len ; j++)
  if (name [j] == ' ' )
flag = 1 ;
        if ( flag == 1)
        Printf (" %c ", name [i+1]);

        else {
        Printf (" %s ", & name [i+1]);
```

```
        break;
      }
    }
  }
  return 0;
}
```

→ Input :

Enter full name;
Avul Pakir jainulabdeen
Abdul Kalam

→ Output :

Abbreviated name : A . P . J . A . Kalam

Unit 3: Functions and Recursive Functions

Q.1 Write a functions power (a,b) to calculate the value of a raised to b

→

```
# include <stdio.h>

int power (int a ; int b) {
int i , result = 1;
for (i=1; i <= b; i++)
  result = result * a;
  result result ;
}

int main () {
int a, b;

Print ("Enter base number (a): ");
scanf ("%d", & a);
Printf (" Enter exponent (b): ");
Scant ("%d" ,& b);

Printf ("%d raised to %d %d \n",
    a,b, Power (a,b));

return 0;

}
```

vision

→ Input:

Enter base number (a) : 2

Enter exponent (b) : 5

→ Output

2 raised to 5 = 32

Q.2 Any year is entered through the keyboard. write a function to determine whether the year is a leap year or not.

```
→ #include <stdio.h>

int isLeap (int year) {
if ((year % 4 == 0 && year % 100
      != 0 ) || (year % 400 == 0))

  return 1;
  else
    return 0;
}
  int main () {
   int year;

Printf (" Enter a year: ");
 Scanf (" %d", & year);
```

```
if (isLeap (year))
  Printf ("%d is a leap year.\n", year);

else

Printf ("%d is not a leap year.\n", year);

return 0;

}
```

→ Input:

Enter a year : 2024

→ Output:

2024 is a leap year.

Q.3 Write a recursive function to calculate factorial of a number.

```
→ #include <stdio.h>

int factorial (int n) {
  if (n==0 || n==1)
    return 1;
  else
    return n * factorial (n-1);
  }
  int main () {
  int ~~main~~ num;
```

```
Printf ("Enter a number: ");
Scanf ("%d", &num);

Print ("Factorial of %d \n"); num , factorial (num));

return 0;
}
```

→ Input:

Enter a number: 5

→ Output:

Factorial of 5 = 120

Q.4 Write a function to swap two integers using call by value, show that the original values are not changed.

→

```
# include <stdio.h>

void swap (int a, int b) {
    int temp ;
    temp=a ;
    a = b ;
    b = temp ;
Printf ("\n Inside function after swapping : a = %d , b= %d,
                                    a, b);
}
```

```
int main () {
  int x, y;

Printf (" Enter two numbers : ");
 scanf ("%d %d ", &x , &y );

Printf (" \n Before calling function: x = %d , y = %d ", x, y );

 swap (x,y);

Printf (" \n After calling function:
   x = %d, y = %d \n", x, y);

return 0;
}
```

→ Input :

Enter two numbers : 5 10

→ Output :

Before calling function: x=5, y=10

Inside function after swapping : a=10 , b=5

After calling functions : x=5 , y=10

Q.5 Write a program that uses a function to update both the maximum and minimum values in an array using call by reference

```
→ #include <stdio.h>
  void findmaxmin(int arr[], int n, int *max, int *min){
  int i;
  *max = *min = arr[0];

  for(i=1; i<n; i++){
  if (arr[i] > *max)
   *max = arr[i];
   if (arr[i] < *min)
  *min = arr[i];
  }
}
  int main(){
  int arr[100], n, i;
  int max, min;

Printf("Enter number of elements:")
 scanf("%d", &n);
Printf("Enter %d elements: \n", n);
 for(i=0; i<n; i++)
 scanf("%d", &arr[i]);
Find maxmin(arr, n, &max, &min);

Printf("\n maximum value = %d", max);
Printf("\n minimum value = %d \n", min);
 return 0;
}
```

VISION

→ Input:

Enter number of elements : 5

Enter 5 elements:

10, 25, 5, 40, 15

→ Output:

maximum value = 40

minimum value = 5

Q.6 Write a program to implement a calculator using separate function for add, subtract, multiply and divide.

```
→ #include <stdio.h>

float add (float a, float b) {
    return a+b;}

float sub (float a, float b) {
    return a-b;}

float mul (float a, float b) { return a*b; }

float div (float a, float b) { return (b!=0)? a/b; 0; }

int main () {
   float x, y;
   int ch;
Printf (" 1. Add 2. Sub 3. Mul 4. Div \n Enter choice: ");
scanf ("%d", &ch);
```

vision

```
Printf ("Enter two numbers : ");
scanf ("%f %f", &x, &y);

switch (ch){
case 1: Printf ("Result = %.2f", add (x,y)); break;
case 2: Printf ("Result = %.2f", sub (x,y)); break;
case 3: Printf ("Result = %.2f", mul (x,y)); break;
case 4: ~~Printf (A~~ if (y! = 0) Printf ("Result = %.d2f",
                    divi (x,y));
else
Printf ("cannot divide by zero"); break;
default : Printf ("Invalid choice");
}
return 0;
}
```

→ Input :

1. Add 2. Sub 3. Mul 4. Div

Enter choice: 1

Enter two numbers : 5 3

→ Output

Result = 8

Q.7 All the above mentioned programs.

1) Print numbers 1 to N

```
# include <stdio.h>
void Printf (int n) { if (n==0)
  return;
Printf (n-1); Printf ("%d ", n); }
int main () {
  int n ;
scanf (" %d ", & n);
printf (n);
}
```

2) Sum of N natural numbers

```
# include <stdio.h>
int sum (int n) {
return (n==0) ?
0 : n + sum (n-1);
}
int main () {
int n;
scanf ("%d", & n);
Printf (" sum = %d ", sum (n)); }
```

3) Factorial

```
# include <stdio.h>
 int fact (int n) {
 return (n<=1) ?
 1: n * fact (n-1); }
 int main () {
```

```
int n;
scanf ("%d",&n);
Printf ("Fact=%d", fact (n)); }
```

4) Reverse a number

```
# include <stdio.h>
int rev (int n, int r) {
return (n==0)?
r: rev (n/10, r*10 + n%10); }
int main () {
int n;
scanf ("%d", &n);
printf ("Rev = %d", rev (n,0)); }
```

5) Fibonacci series

```
# include <stdio.h>
int fib (int n) {
return (n<=1)?
n: fib(n-1) + fib (n-2); }
int main () {
int n, i;
scanf ("%d",&n);
for (i=0; i<n; i++)
Printf ("%d", fib(i); }
```

6) GCD (Greatest common divisioner)

```
# include <stdio.h>
int gcd (int a , int b) {
return (b==0)?
a : gcd (b, a %.b); }
int main () {
int a,b;
scanf ("%.d %.d ", &a , &b );
printf ("gcd = %.d ", gcd (a,b); }
```

7) count digits

```
# include <stdio.h>
int count (int n) {
return (n==0)?
0 : 1+ count (n /10); }
int main () {
int n;
scanf ("%.d", &n);
printf ("digits = %.d", count (n)); }
```

8) Sum of Digits

```
# include <stdio.h>
int sumD (int n) { return (n==0)?;
(n %. 10)+ sumD (n) 10); }
int main () { int n; scanf ("%.d", &n);
Printf (" sum = %d ", sumD (n)); }
```

vision

a) Power (a^b)

```
# include <stdio.h>
int power (int a, int b) { return (b==0)?
1 : a * power (a,b-1); }
int main () { int a, b;
scanf ("%d%d", &a, &b);
printf ("%d ^ %d = %d", a, b, power(a,b)); }
```

Q.8 All the programs of loop using recursion.

1) Print numbers from 1 to N

```
# include <stdio.h>
 void print (int n){
 if (n==0)
 return;
 printf (n-1);
 printf ("%d", n);
 }
 int main () {
  int n
  printf ("Enter N: ");
  scanf ("%d", &n);
  printf (n);
  return 0;
 }
```

2) sum of first N natural numbers

```
# include <stdio.h>
int sum(int n){
 if (n==0)
 return 0;
else
 return n + sum(n-1);
}
 int main(){
 int n;
 Printf ("Enter N: ");
 scanf ("%d", &n);
 printf ("sum = %d", sum(n));
 reuturn 0;
}
```

3) Factorial of a number

```
# include <stdio.h>
 int fact (int n){
 if (n==0 || n==1)
  return 1;
else
 return n * fact(n-1);
}
 int main (){
 int n;
 Printf ("Enter a number : ");
 scanf ("%d", &n);
 Printf (" Factorial = %d"; fact (n)));
  return 0;
 }
```

4) Reverse a number

```
# include <stdio.h>
int rev (int n, int r){
if (n==0)
return r;
return rev (n/10, r*10 + n%10);
}
int main (){
int n;
printf (" Enter number : ");
scanf ("%d", &n);
printf (" Reversed = %d ", rev (n,0));
}
```

5) Fibonacci series

```
# include <stdio.h>
int fib(int n){
if (n==0) return 0;
if (n==1) return 1;
return fib(n-1)+ fib (n-2);
}
int main (){
int n,i;
printf ("Enter number of terms:");
scanf ("%d", &n);
for (i=0, i<n; i++)
printf ("%d] fib (i));
return 0;
}
```

6) GCD (Greatest common divisor)

```
# include <stdio.h>
int gcd (int a, int b){
 if (b==0)
  return a;
else
return gcd (b,a%b);
}
int main (){
int a,b;
Printf (" Enter two numbers: ");
scanf ("%d %d, &a, &b);
printf ("gcd = %d ", gcd (a,b));
 return 0;
}
```

7) count digits of a numbers.

```
# include <stdio.h>
int count (int n) {
if (n==0)
 return 0;
 return 1 + count (n/10);
}
int main () {
int n;
printf (" Enter numbers: ");
scanf ("%d ", &n );
printf ("Digits = %d ", count (n));
return 0;
}
```

vision

8) Sum of Digits

```
# include <stdio.h>
int sum digits (int n){
if (n==0)
return 0;
return (n%10) + sumdigits (n/10);
}
int main (){
int n;
printf ("Enter number: ");
scanf ("%d", &n);
printf ("sum of digits = %d",
sumdigits (n));
return 0;
```