

Q1.

Simulation of a Simple Calculator.

Program :

```
# include <stdio.h>
# include <stdlib.h>
int main()
{
    int a, b, res ;
    char op ;
    printf("\n Enter a simple arithmetic expression");
    scanf("%d %c %d", &a, &op, &b);
    switch (op)
    {
        case '+':
            res = a+b ;
            break ;
        case '-':
            res = a-b ;
            break ;
        case '*':
            res = a*b ;
            break ;
    }
}
```

D	D	M	M	Y	Y	Y	Y

case '/':

if (b != 0)

res = a/b;

else

{

printf("division by zero is not possible");

exit(0);

}

break;

case '%':

res = a % b;

break;

default:

~~printf("illegal operator");~~

~~exit(0);~~

}

printf("\n %d %c %d = %d", a, op, b, res);

return(0);

};

D	D	M	M	Y	Y	Y	Y
11	11	2023					

Q2. Compute the roots of quadratic equation by accepting the coefficients. Print appropriate message.

Program :

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main ()
{
    float a,b,c,d,root1,root2,real,imag;
    printf("Enter the three coefficients : \n");
    scanf("%f %f %f", &a, &b, &c);
    if (a == 0)
    {
        root printf("Invalid coefficient");
        exit(0);
    }
    d = b * b - 4 * a * c;
    if (d > 0)
    {
        root1 = (-b + (sqrt(d))) / (2.0 * a);
        root2 = (-b - (sqrt(d))) / (2.0 * a);
    }
}
```

printf ("The roots are real and distinct.\n");

printf ("root1 = %f\n root2 = %f\n", root1, root2);

}

else if (d == 0)

{

root1 = root2 = -b / (2.0 * a);

printf ("The roots are real and equal....\n");

printf ("root1 = %f\n root2 = %f\n", root1, root2);

}

else

{

real = -b / (2.0 * a);

imag = sqrt (fabs (a)) / (2.0 * a);

printf ("The roots are complex and imaginary....\n");

printf ("root1 = %f + i %f\n root2 = %f - i %f",

real, imag, real, imag);

}

return 0;

{

D	D	M	M	Y	Y	Y	Y
1	1	1	1	2	0	2	3

03. An electricity board charges the following rates for the use of electricity : for the first 200 units 80 paise per unit : for the next 100 units 90 paise per unit : beyond 300 units Rs 1 per unit. All users are charged a minimum of Rs 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and printout the charges.

Program :

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
char name[20] ;
```

```
int units ;
```

```
float charges = 0 ;
```

```
printf ("\n enter the name of the user : ") ;
```

```
gets(name) ;
```

```
printf ("\n enter number of units consumed : ") ;
```

```
scanf ("%d", &units) ;
```

D	D	M	M	Y	Y	Y	Y

if (units <= 200)

}

charges = units * 0.80 ;

}

else if (units <= 300)

{

charges = 200 * 0.80 + (units - 200) * 0.90 ;

}

else

{

charges = 200 * 0.80 + 100 * 0.90 + (units - 300) * 1.00 ;

}

charges = charges + 100 ;

if (charges > 400)

~~charges = charges + 0.15 * charges ;~~

~~printf ("%s has to pay rupees %f", name,
charges);~~

return 0 ;

}

D	D	M	M	Y	Y	Y	Y
1	1	1	1	2	0	2	3

04. Write a C program to display the following by reading the number of rows as input,

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

 n^{th} row

Program :

#include <stdio.h>

int main ()

{

int i, j, row ;

printf("Enter number of rows : ");

scanf ("%d", &row) ;

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

{

for (j=1 ; j<=row-i ; j++)

{

printf(" ") ;

{

for ($j=1$; $j < i$; $j++$)

{

 printf ("%d", j);

}

for ($j = i - 1$; $j >= 1$; $j--$)

{

 printf ("%d", j);

}

 printf ("\n");

}

return 0;

}



D	D	M	M	Y	Y	Y	Y
1	5	1	1	2	0	2	3

05

Implement Binary Search on Integers

Program :

```
# include < stdio.h >
# include < string.h >
int main ()
{
```

```
int a[10], key ;
```

```
int n, i, low, high, mid, found = 0 ;
```

```
printf("enter the number of numbers to read \n");
```

```
scanf("%d", &n);
```

```
printf("enter the numbers in ascending order \n");
```

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

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

```
printf("enter the number to search \n");
```

```
scanf("%d", &key);
```

```
low = 0 ;
```

```
high = n-1 ;
```

```
while (low <= high && found != 1)
```

```
{
```

```
mid = (low + high) / 2 ;
```

```
if (a[mid] == key)
```

```
found = 1 ;
```

D D M M Y Y Y Y

```
else if (a[mid] < key)
    low = mid + 1 ;
else
    high = mid - 1 ;
}
if (found == 1)
    printf ("\n number found at position :
            %d", mid + 1) ;
else
    printf ("\n number not found") ;
return 0 ;
}
```

✓

D	D	M	M	Y	Y	Y	Y
1	5	1	1	2	0	2	3

06. Sort the given set of N numbers using Bubble sort

Program :

```
#include <stdio.h>
int main ()
{
    int n, i, j, a[10], temp;
    printf ("\n enter the number of elements");
    scanf ("%d", &n);
    printf ("Enter the array elements\n");
    for (i=0 ; i<n ; i++)
    {
        scanf ("%d", &a[i]);
    }
    printf ("Original elements are \n");
    for (i=0 ; i<n ; i++)
    {
        printf ("%d \t", a[i]);
    }
    for (i=0 ; i<n-1 ; i++)
    {
        for (j=0 ; j<n-1-i ; j++)
    }
```

{

if ($a[j] > a[j+1]$)

{

temp = $a[j]$;

$a[j] = a[j+1]$;

$a[j+1] = \text{temp}$;

}

}

}

printf("The sorted elements are \n");

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

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

return 0;

}

U

D	D	M	M	Y	Y	Y	Y
18		11		20	2	3	

Q.7.

Write a C program for matrix multiplication.

Program :

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a[10][10], b[10][10], c[10][10];
    int i, j, k, m, n, p, q;
    printf("Enter the order of the matrix A \n");
    scanf("%d %d", &m, &n);
    printf("Enter the order of the matrix B \n");
    scanf("%d %d", &p, &q);
    if (n != p)
    {
        printf("\n Multiplication not possible");
        exit(0);
    }
    else
    {
        printf("Enter the elements of matrix A \n");
    }
}

```

for ($i=0$; $i < m$; $i++$)

{

for ($j=0$; $j < n$; $j++$)

{

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

}

}

printf("Enter the elements of matrix B.\n");

for ($i=0$; $i < p$; $i++$)

{

for ($j=0$; $j < q$; $j++$)

{

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

}

}

printf("\n Matrix A \n");

for ($i=0$; $i < m$; $i++$)

{

printf("\n");

for ($j=0$; $j < n$; $j++$)

{

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

}

}

D	D	M	M	Y	Y	Y	Y

printf ("In Matrix B \n");

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

printf ("\n");

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

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

}

}

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

{

for (j=0 ; j<q ; j++)

{

c[i][j] = 0 ;

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

c[i][j] = c[i][j] + a[i][k] * b[k][j] ;

{

printf ("In Product of A and B matrices :

MATRIX C \n");

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

{

printf ("\n");

for (j = 0; j < 10; j++)

{

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

}

}

}

}

1/1/12

D	D	M	M	Y	Y	Y	Y
1	1	2		2	0	2	3

08.

Write a function to implement string operations such as string length, compare and concatenate. Convince the parameter passing technique.

```
#include <stdio.h>
int length (char s[10]);
void concat (char s1[10], char s2[10]);
void compare (char s1[10], char s2[10]);
int main ()
{
    char s1[10], s2[10];
    int len1, len2;
    printf ("Enter the string 1 : ");
    gets (s1);
    printf ("Enter the string 2 : ");
    gets (s2);
    len1 = length (s1);
    len2 = length (s2);
    printf ("length of s1 = %d \n", len1);
    printf ("length of s2 = %d \n", len2);
    compare (s1, s2);
    concat (s1, s2);
    return 0;
}
```

```
int length (char s[10]) {
    int i = 0;
    while (s[i] != '\0')
        i++;
    return i;
}
```

```
void compare (char s1[10], char s2[10]) {
    int i = 0;
    while (s1[i] == s2[i] && s1[i] != '\0' &&
           s2[i] != '\0')
        i++;
    if (s1[i] == '\0' && s2[i] == '\0')
        printf ("Strings are same \n");
    else
        printf ("Strings are not same \n");
}
```

void concat (char s1[10], char s2[10])

{

char s3[20] :

int i = 0, j = 0 ;

while (s1[i] != '\0')

{

s3[i] = s1[i] ;

i++ ;

}

while (s2[j] != '\0')

{

s3[i+j] = s2[j] ;

j++ ;

}

~~s3[i+j] = '\0' ;~~

printf (" concatenated string is %s", s3);

}

D	D	M	M	Y	Y	Y	Y
1	2	1	2	2	0	2	3

09.

Compute $\sin(x)$ using Taylor series approximation. Compare your result with the built-in library function. Print both the results with appropriate inferences.

```

#include <stdio.h>
#include <math.h>
#define PI 3.142
int main()
{
    int i, degree;
    float x, sum=0, term, nume, deno;
    printf("Enter the value of degree : ");
    scanf("%d", &degree);
    x = degree * (PI/180);
    nume = x;
    deno = 1;
    i = 2;
    do {
        term = nume/deno;
        nume = -nume * x * x;
        deno = deno * i * (i+1);
        sum = sum + term;
    } while (i < 10);
    printf("Sum = %f", sum);
}

```

D D M M Y Y Y Y

i = i + 2 ;

3 while (fabs (term) >= 0.00001) ;

printf ("Sine of %d is %f", degree, sum) ;

printf ("In The Sine function of %d is %f
using library function", degree, sin(x)) ;

return 0 ;

3.

D	D	M	M	Y	Y	Y	Y
1	2	1	2	2	0	2	3

10. Develop a program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of N real numbers.

```

#include <stdio.h>
#include <math.h>
int main()
{
    float a[10], *ptr, mean, std, sum = 0;
    float sumstd = 0;
    int n, i;
    printf ("\n Enter the number of elements");
    scanf ("%d", &n);
    printf ("\n Enter the array elements");
    for (i=0; i<n; i++)
        scanf ("%f", &a[i]);
    ptr = a;
    for (i=0; i<n; i++)
    {
        sum = sum + *ptr;
        ptr++;
    }
    }
```

mean = sum / n ;

ptr = a ;

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

sumstd = sumstd + pow((*ptr - mean), 2);

ptr ++ ;

}

std = sqrt (sumstd / n) ;

printf (" sum = %f \n" , sum) ;

printf (" Mean = %f \n" , mean) ;

printf (" Standard Deviation = %f \n" , std) ;

return 0 ;

}

b2
18/23