

Name : Naeem khan

ID : 2012020105

Course Code : CSE-1214

Section : C

Batch : 53

problem 01 :

Based on the problem it is required to get the input of lower limit and upper limit. The program should display the all factorial of positive integer in between lower limit to upper limit as output.

The input variable should be : low, up. We need to calculate factorial of lower limit to upper limit. So the variable 'fact' holds the factorial to be displayed.

$$\text{factorial} = \text{factorial} * i$$

∴ Hence,

$$fact = 1$$

$$fact = fact * i$$

Input variables	processing variables	output variables	Necessary header files/functions
low, up (int)	i (int) fact (int)	fact (int)	stdio.h Scantf () printf () for submitted I/O

Now, The program for the Sine problem is as follows:-

```
* include <stdio.h>
int main ()
{
    int low, up, fact = 1 ;
    printf ("Enter the lower limit : ");
    scanf ("%d", &low);
    printf ("Enter the upper limit : ");
    scanf ("%d", &up);
    while (low <= up) {
        for (int i = 1; i <= low; i++)
            fact = fact * i;
        printf ("Factorial of %d : %d\n", low, fact);
        fact = 1;
        low++;
    }
    return 0;
}
```

problem 02 :

Based on the problem, it is required to get the input of a integer N. Like N is 9. The program should be display fibonacci series, like 0 1 1 3 5 8 13 21 as output. The input variable should be N. We need to above Series.

Fibonacci Series : $F_{n-1} + F_{n-2}$
With seed value

Hence,

$$F_0 = 0, F_1 = 1$$

$$\text{fib}_1 = 0, \text{fib}_2 = 1$$

$$\text{temp} = \text{fib}_1 + \text{fib}_2$$

$$\text{fib}_1 = \text{fib}_2$$

$$\text{fib}_2 = \text{temp}$$

Input variables	processing variables	output variables	necessary header files / function
N(int)	temp, fib1 fib2 (int)	fib1 (int)	stdio.h scanf () printf () for submitted i/o

2

```
#include <stdio.h>
int main()
{
    int N, fib1, fib2, temp;
    printf ("Enter a number : ");
    scanf ("%d", &N);
    fib1 = 0, fib2 = 1;
    printf ("Fibonacci Series : %d ", fib1);
    for (int i = 2; i <= N; i++) {
        temp = fib1 + fib2;
        fib1 = fib2;
        fib2 = temp;
        printf ("%d ", fib1);
    }
    return 0;
}
```

3pr: Based on the problem , it is required to get the input of two binary number . The program should display binary addition with out any carry as output. The input variable should be : p,q . We need to addition of binary number . So the variable 'r' holds the binary addition to be displayed .

Binary addition

p	q	r
<hr/>		
0	+ 0	= 0
1	+ 0	= 1
0	+ 1	= 1
1	+ 1	= 0

Hence

$$P = 10110011$$

$$Q = 10100101$$

$$\underline{R = 00010110}$$

Input variables	processing variables	output variable	Necessary header files / function
P, q { char } Stringz	len1, len2, temp (int) P[i], q[i] (char)	Z { char } Stringz	stdio.h scanf() printf() for submitted I/O string.h

3/

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

int main()
{
    char p[35];
    char q[35];
    char r[35];
    printf("Enter a first binary : ");
    scanf("%s", &p);
    printf("Enter a second binary : ");
    scanf("%s", &q);
```

```
int len1 = strlen(p);
int len2 = strlen(q);
int temp;
if (len1 >= len2)
    temp = len1;
else
    temp = len2;
for (int i=0; i<=temp; i++){
    if (p[i] == '0' && q[i] == '0')
        r[i] = '0';
    if (p[i] == '1' && q[i] == '0')
        r[i] = '1';
    if (p[i] == '0' && q[i] == '1')
        r[i] = '1';
    if (p[i] == '1' && q[i] == '1')
        r[i] = '1';
}
printf (" Addition of binary without any
carry : %sm", r);
return 0;
```

4 NO:

Based no the problem , it is required to get the input of the numbers of rows and columns and elements of matrix . The program should display given matrix will change its every row to every column (transpose matrix). The input variables should be : p, q, matrix . We need to transpose of this given matrix , so the variable a transpose matrix ,

Given matrix :

$$\text{matrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ b_{11} & b_{12} & b_{13} \\ c_{11} & c_{12} & c_{13} \end{bmatrix}$$

$$\text{transpose-matrix} = \begin{bmatrix} a_{11} & b_{11} & c_{11} \\ a_{12} & b_{12} & c_{12} \\ a_{13} & b_{13} & c_{13} \end{bmatrix}$$

Input variables	processing variables	output variables	Necessary header files/functions
p, q (int)	transpose - matrix (int)	transpose - matrix()	stdio.h
matrix (int)		scanf()	printf() for formatted I/O

* include <stdio.h>

int main ()

{ int matrix [10][10], p, q; transpose - matrix
[10][10];

printf ("Enter rows and columns : ");

scanf ("%d %d", &p, &q);

printf ("\nEnter elements of matrix :\n");

~~for (%d)~~

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

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

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

}

```
for (int i=0; i<p; i++)
    for (int j=0; j<q; j++) {
        transpose-matrix[j][i] = matrix[i][j];
    }
    cout << "In change of every row and column matrix : m";
    for (int i=0; i<q; i++)
        for (int j=0; j<p; j++) {
            cout << a%10 << ", transpose-matrix[i][j]);";
            if (j == p-1)
                cout << endl;
        }
    cout << "m";
}
return 0;
}
```

5 No: Based on the program, it is required to get the input of 5 float numbers. The program should display ~~not~~ maximum among this numbers (using ternary operator). The input variables a, b, c, d, e . We need to find maximum number. So the variable 'maximum' holds the maximum number to be displayed.

Hence:

$$\text{maximum} = (a > b \text{ ?? } a > c \text{ ?? } a > d \text{ ?? } a > e) ?$$

$$a : ((b > c \text{ ?? } b > d \text{ ?? } b > e) ?$$

$$b : (c > d \text{ ?? } c > e) ?$$

$$c : (d > e) ?$$

$$d : e);$$

Input variables	processing variable	output variables	Necessary header file / function
a, b, c, d, e (float)	maximum (float)	maximum (float)	stdio.h scanf() printf() for submitted I/O.

```

#include <stdio.h>
int main ()
{
    float a, b, c, d, e, maximum;
    printf ("Enter the float number : ");
    scanf ("%f, %f, %f, %f, %f", &a, &b, &c, &d, &e);
    maximum = (a > b && a > c && a > d && a > e);
    a: ((b > c && b > d && b > e));
    b: (c > d && c > e);
    c: (d > e);
    d: e);
    printf ("Maximum number : %0.2f m", maximum);
    return 0;
}

```

Q No :- Based on the problem, it is required to get the input number of students, student name, student id, student gpa. The program should display CGPA and GRADE (Flooring, ceiling) as output. The input variable should be : name, id, C, algo, eng, math. we need to calculate CGPA. So the variable, 'CGPA' holds the CGPA and Grade to be displayed.

$$CGPA = \frac{\text{Total CGPA on Sub}}{\text{Total Sub number}}$$

Input Variable	processing Variables	output Variable	necessary header files/function
n, id (int)	cgpa (float)	cgpa (float)	stdio stdio.h
name (char)		grade	scanf () for
c, algo, eng,			Submitted i/o
math (float)			

6/

```
# include < stdio.h >
struct Students {
    char name[100];
    int id;
    double c;
    double algo;
    double eng;
    double math;
};

int main()
{
    int n;
    printf("Number of Students : ");
    scanf("%d", &n);
    struct Students student[n];
    for (int i=0; i<n; i++) {
        printf("Student %d\n", i+1);
        printf("Student Name : ");
        scanf("%s", student[i].name);
        scanf("%s", student[i].name);
    }
}
```

```
    printf ("Student ID: ");
    scanf ("%d", &student[i].id);
    printf ("GPA of C: ");
    scanf ("%f", &student[i].c);
    printf ("GPA of Algo: ");
    scanf ("%f", &student[i].algo);
    printf ("GPA of English: ");
    scanf ("%f", &student[i].eng);
    printf ("GPA of math: ");
    scanf ("%f", &student[i].math));
}

for (int i=0; i<n; i++){
    printf ("Student %d : \n", i+1);
    double cgpa = (student[i].c + student[i].algo +
                    student[i].eng + student[i].math) / 4;
    printf ("CGPA = %f\n",
```

#

```
if (cgpa == 4){  
    printf ("GRADE : A+\n");  
}  
  
else if (cgpa == 3.75){  
    printf ("GRADE : A\n");  
}  
  
else if (cgpa == 3.50){  
    printf ("GRADE : A-\n");  
}  
else if (cgpa == 3.25){  
    printf ("GRADE : B+\n");  
}  
  
else if (cgpa == 3.00){  
    printf ("GRADE : B\n");  
}  
  
else if (cgpa == 2.75){  
    printf ("GRADE : B-\n");  
}  
  
else if (cgpa == 2.50){  
    printf ("GRADE : C+\n");  
}  
}
```

else if ($3.75 \leq \text{cgpa} < 4.00$) {

if ($3.75 \leq \text{cgpa} < 3.875$)

printff ("GRADE : A (Equivalence to 3.75 (using
Flooring))\n");

else printff ("GRADE : A+ (Equivalence to 4.00
) (using ceiling))\n");

else if ($3.50 \leq \text{cgpa} < 3.75$) {

if ($3.25 \leq \text{cgpa} < 3.375$)

printff ("GRADE : B+ (Equivalence to 3.25 (using
Flooring))\n");

else printff ("GRADE : A- (Equivalence to 3.50
(using ceiling))\n");

else if ($3.00 \leq \text{cgpa} < 3.25$) {

if ($3.00 \leq \text{cgpa} < 3.125$)

printff ("GRADE : B (Equivalence to 3.00
(using Flooring))\n");

else printff ("GRADE : B (Equivalence to
3.00 (using ceiling))\n");

else if ($2.75 \leq \text{cspa} < \text{cspa}(3.00)$) {

 if ($2.75 \leq \text{cspa} & \text{cspa} \leq 2.875$)

 print ("GRADE : B- (Equivalence to 2.75 (using
 Flooring))\n");

 else print ("GRADE : B (Equivalence to 3.00
 (using ceiling))\n");

else if ($2.50 \leq \text{cspa} & \text{cspa} < 2.75$) {

 if ($2.50 \leq \text{cspa} & \text{cspa} \leq 2.625$)

 print ("GRADE : C+ (Equivalence to 2.50
 (using Flooring))\n");

 else print ("GRADE : B- (Equivalence to 2.75
 (using ceiling))\n");

}

}

return 0;

7 No :- Based on the problem , it is required to get the input of P,R,T. The program should display simple ^{interest} output. The variable should be P,R,T. We need to calculate simple interest. So the variables 'I' holds the interest to be displayed.

Hence

$$I = \frac{P * T * R}{100}$$

Input variables	processing variables	output variables	necessary header files/ function
P, T (int) R (float)	I (float)	I (float)	stdio.h printf() for formatted i/o .

```
#include <stdio.h>
int main ()
{
    int P,T;
    float R,I;
    P=200, T=2, R=0.5;
    I=(P*T*R)/100;
    printf ("Interest : %.2f\n",I);
    return 0;
}
```