Problem 01: A C program to take **two integers(i1, i2)** and **two float values (f1,f2)** as input from the user and print the following:

1. **Average of floating numbers.**

2. **Bitwise-OR and Bitwise-AND of integers numbers**.

3. **Left-shift i1 by 2 places and Right-shift i2 by 1 place and print out the result**. (Note that left-shift multiplies the number by 2 and right-shift divides it by 2).

#include<stdio.h>

int main()

{

int i1,i2,bitwise\_OR,bitwise\_AND;

float f1,f2,avg;

printf("Enter i1 and i2: ");

scanf("%d %d",&i1,&i2);

printf("Enter f1 and f2: ");

scanf("%f %f",&f1,&f2);

avg = (f1 + f2) / 2;

printf("Average of f1 and f2 is: %.2f\n",avg);

bitwise\_OR = i1 | i2;

bitwise\_AND = i1 & i2;

printf("Bitwise-OR of i1 and i2 is: %d\n",bitwise\_OR);

printf("Bitwise-AND of i1 and i2 is: %d\n",bitwise\_AND);

int left\_shift\_i1 = i1 << 2;

int right\_shift\_i2 = i2 >> 1;

printf("i1 shifted left by two places: %d\n",left\_shift\_i1);

printf("i2 shifted right by one places: %d\n",right\_shift\_i2);

return 0;

}

Problem 02: Sum of three digit decimal integer:

#include<stdio.h>

int main()

{

int n,rem,sum=0;

printf("Enter a three-digit integer: ");

scanf("%d",&n);

while(n!=0)

{

rem=n%10;

sum=sum+rem;

n=n/10;

}

printf("Sum of the digits of the numbers is: %d",sum);

getch();

}

Problem 03: Number Palindrome:

#include<stdio.h>

int main()

{

int n,rem,sum=0;

int temp;

printf("Enter an integer:");

scanf("%d",&n);

temp=n;

while(temp!=0)

{

rem=temp%10;

sum=sum\*10+rem;

temp= temp/10;

}

if(sum==n)

printf("Yes, it's a palindrome");

else

printf("No, it's not a palindrome");

return 0;

}

Problem 04: String Palindrome:

#include <stdio.h>

#include <string.h>

int main()

{

char s1[20];

int i, length;

int flag = 0;

printf("Enter a string:");

scanf("%s", s1);

length = strlen(s1);

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

{

if (s1[i] != s1[length- i-1])

{

flag = 1;

break;

}

else

flag=0;

}

if (flag==0)

printf("%s is a palindrome", s1);

else

printf("%s is not a palindrome", s1);

return 0;

}

Problem 05: First N prime numbers:

#include<stdio.h>

int main()

{

int n,i,j,flag=0;

printf("Enter N: ");

scanf("%d",&n);

printf("1st %d Prime Numbers are: ",n);

for(i=2; i!=0 && n!=0; i++)

{

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

{

if(i%j==0)

{

flag=1;

break;

}

else

flag=0;

}

if(flag==0)

{

printf("%d ",i);

n--;

}

}

getch();

}

Problem 06: Full Pyramid of Numbers:

1  
2 3 2  
3 4 5 4 3  
4 5 6 7 6 5 4  
5 6 7 8 9 8 7 6 5

#include <stdio.h>

int main()

{

int row,col,m,n;

printf("Enter N:");

scanf("%d",&n);

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

{

for(col=1; col<=n-row; col++)

{

printf("\t");

}

for(col=1,m=row; col<row; col++,m++)

{

printf("\t%d",m);

}

for(col=1; col<=row; col++,m--)

{

printf("\t%d",m);

}

printf("\n");

}

return 0;

}

Problem 07: Pascal's Triangle with N rows:

1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
1 5 10 10 5 1

#include<stdio.h>

int main()

{

int n,row,col;

int count=1;

printf("Enter N:");

scanf("%d",&n);

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

{

for(col=1; col<=n-row; col++)

{

printf(" ");

}

for(col=0; col<=row; col++)

{

if(row==0 || col==0)

count=1;

else

count=count\*(row-col+1)/col;

printf(" %d",count);

}

printf("\n");

}

return 0;

}

Problem 08: Pyramid of '\*' with N rows:

\* \* \* \* \* \* \* \*  
\* \* \* \* \* \* \*  
\* \* \* \* \*  
\* \* \*  
\*

#include<stdio.h>

int main()

{

int row,col,n;

printf("Enter N:");

scanf("%d",&n);

for(row=n; row>=1; row--)

{

for(col=1; col<=n-row; col++)

{

printf(" ");

}

for(col=1; col<=2\*row-1; col++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

Problem 09: Add/Subtract of two matrix:(using user defined function)

#include <stdio.h>

// Function declaration

void getMatrixElements(int r, int c, int m[][c]);

void AddMatrices(int r, int c, int m1[][c], int m2[][c], int result[][c]);

void SubMatrices(int r, int c, int m1[][c], int m2[][c], int result[][c]);

void display(int r, int c, int m[][c]);

int main()

{

int row, col;

// take the number of rows and columns from the user

printf("Enter the number of rows and columns of the matrices: ");

scanf("%d %d", &row, &col);

int A[row][col], B[row][col], add[row][col], dif[row][col];

// take the elements of the two matrices

printf("\nEnter the %d elements of the matrix A: ", row\*col);

getMatrixElements(row, col, A);

printf("\nEnter the %d elements of the matrix B: ", row\*col);

getMatrixElements(row,col,B);

// operations

AddMatrices(row, col, A, B, add);

SubMatrices(row, col, A, B, dif);

// Display the matrices and the result of addition and subtraction

printf("\nThe first matrix is: \n");

display(row, col, A);

printf("\nThe second matrix is: \n");

display(row, col, B);

printf("\nThe sum of the two entered matrices is: \n");

display(row, col, add);

printf("\nThe difference (subtraction) of the two entered matrices is: \n");

display(row, col, dif);

return 0;

}

// Function to take matrix elements input from the user

void getMatrixElements(int r, int c, int m[][c])

{

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

for (int j = 0; j < c; j++)

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

}

// Function to add two matrices

void AddMatrices(int r, int c, int m1[][c], int m2[][c], int result[][c])

{

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

for (int j = 0; j < c; j++)

result[i][j] = m1[i][j] + m2[i][j];

}

// Function to subtract two matrices

void SubMatrices(int r, int c, int m1[][c], int m2[][c], int result[][c])

{

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

for (int j = 0; j < c; j++)

result[i][j] = m1[i][j] - m2[i][j];

}

// Function to display a matrix

void display(int r, int c, int m[][c])

{

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

{

for (int j = 0; j < c; j++)

{

printf("%d ", m[i][j]);

}

printf("\n");

}

}

Problem 10: Multiply two matrices: (using user defined function)

#include<stdio.h>

// Function declaration

void getMatrixElements(int r,int c,int A[][c]);

void display(int r,int c, int A[][c]);

void multiplyMatrices(int r1,int c2,int c1,int A[][c1],int B[][c2],int multi[][c2]);

int main()

{

int r2,c2,r1,c1;

/\*taking rows & cols of first & second matrix until accurate\*/

printf("Instruction: to multiply column of 1st matrix must be equal to row of second matrix");

do

{

printf("\nEnter the row & col of first matrix: ");

scanf("%d %d",&r1,&c1);

printf("enter the rows & cols of second matrix: ");

scanf("%d %d",&r2,&c2);

if(c1!=r2)

{

printf("\n\aError!! it can not be multiplied.\n");

printf("Please read instruction and input again\n");

}

}

while(c1!=r2);

int a[r1][c1],b[r2][c2],multiplication[r1][c2];

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

getMatrixElements(r1,c1,a);

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

getMatrixElements(r2,c2,b);

multiplyMatrices(r1,c2,c1,a,b,multiplication);

printf("\n\nmatrix A= \n");

display(r1,c1,a);

printf("\n\nmatrix B= \n");

display(r2,c2,b);

printf("\n\nthe multiplication is = \n");

display(r1,c2,multiplication);

return 0;

}

//taking the elements of matrices

void getMatrixElements(int r,int c,int A[][c])

{

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

for(int j=0; j<c; j++)scanf("%d",&A[i][j]);

}

//printing matrices

void display(int r,int c, int A[][c])

{

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

{

printf("\t");

for(int j=0; j<c; j++)

{

printf("%d ",A[i][j]);

}

printf("\n");

}

}

// multiplication

void multiplyMatrices(int r1,int c2,int c1,int A[][c1],int B[][c2],int multi[][c2])

{

for(int i=0; i<r1; i++)

{

for(int j=0; j<c2; j++)

{

multi[i][j]=0;

for(int k=0; k<c1; k++)

multi[i][j]+=A[i][k]\*B[k][j];

}

}

}

Problem 11: A C Program to find factorial of a given number using  
recursion.

Problem 12: A C Program to find the Fibonacci Series consisting of n  
numbers using recursive function.