1. **Write a 'C' program for checking whether number is Armstrong or not.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int num, a, b=0,temp;**

**printf ("enter the number :- ");**

**scanf ("%d",&num);**

**temp= num;**

**while (num>0)**

**{**

**a=num%10;**

**b=b+ a\*a\*a;**

**num= num /10;**

**}**

**if(temp==b)**

**printf("Entered number is armstrong");**

**else**

**printf("Entered number is not armstrong");**

**return 0;**

**}**

1. **Write a C Program to Find the Binary Value of Decimal Number using loop**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**long int decimalNumber, remainder, quotient;**

**int binaryNumber[100],i=1,j;**

**printf("Enter any decimal number: ");**

**scanf("%ld",&decimalNumber);**

**quotient = decimalNumber;**

**while(quotient!=0)**

**{**

**binaryNumber[i++]= quotient % 2;**

**quotient = quotient / 2;**

**}**

**printf("Equivalent binary value of decimal number %d: ",decimalNumber);**

**for(j = i -1 ;j> 0;j--)**

**{**

**printf("%d",binaryNumber[j]);**

**}**

**return 0;**

**}**

1. **Write a c program for entered string is palindrome or not.**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include<string.h>**

**int main()**

**{**

**int n;**

**char str1[10],str2[10];**

**printf("enter the string1 -");**

**scanf("%s",str1);**

**strcpy(str2,str1);**

**strrev(str1);**

**n=strcmp(str1,str2);**

**if(n==0)**

**{**

**printf("enter the string is palindrom");**

**}**

**else**

**{**

**printf("enter the string not palindrom");**

**}**

**return 0;**

**}**

1. **Write a ‘C’ program for entered number is palindrome or not.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int n, reverse = 0, temp;**

**printf("Enter a number to check if it is a palindrome or not\n");**

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

**temp = n;**

**while( temp != 0 )**

**{**

**reverse = reverse \* 10;**

**reverse = reverse + temp%10;**

**temp = temp/10;**

**}**

**if ( n == reverse )**

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

**else**

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

**}**

1. **Write a program for delete an element from an array**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int a[10],i,n,pos;**

**printf("enter the size of array:- ");**

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

**printf("enter the elements in an array:- \n");**

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

**{**

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

**}**

**printf("array elements are:-");**

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

**{**

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

**}**

**printf("enter the position where you wish to delete an element: - ");**

**scanf("%d",&pos);**

**if(pos>=n+1)**

**printf("Deletion not possible\n");**

**else**

**{**

**for(i=pos-1;i<n-1;i++)**

**a[i]=a[i+1];**

**printf("Resultant array is\n");**

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

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

**}**

**return 0;**

**}**

1. **Write a program for finding minimum number from array.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int a[10],i, n, location=1,min;**

**printf("enter the size of array:- ");**

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

**printf("enter the elements in an array:- \n");**

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

**{**

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

**}**

**printf("array elements are:-\n");**

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

**{**

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

**}**

**min=a[0];**

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

**{**

**if(a[i]<min)**

**{**

**min=a[i];**

**location=i+1;**

**}**

**}**

**printf("Minimum elements is present at location %d and its value is %d.\n",location,min);**

**return 0;**

**}**

1. **Transpose matrix**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int a[10][10],r,c,i,j,b[10][10];**

**printf("enter the dimension:-");**

**scanf("%d%d",&r,&c);**

**printf("\n enter the elements in array:- \n");**

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

**{**

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

**{**

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

**}**

**}**

**printf("matrix is:-\n");**

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

**{**

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

**{**

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

**}**

**printf("\n");**

**}**

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

**{**

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

**{**

**b[i][j]=a[j][i];**

**}**

**}**

**printf("Transpose matrix is:-\n");**

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

**{**

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

**{**

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

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**8) matrix multiplication**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int m, n, p, q, c, d, k, sum = 0;**

**int first[10][10], second[10][10], multiply[10][10];**

**printf("Enter the number of rows and columns of first matrix\n");**

**scanf("%d%d", &m, &n);**

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

**for ( c = 0 ; c < m ; c++ )**

**{**

**for ( d = 0 ; d < n ; d++ )**

**{**

**scanf("%d", &first[c][d]);**

**}**

**}**

**printf("Enter the number of rows and columns of second matrix\n");**

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

**if ( n != p )**

**printf("Matrices with entered orders can't be multiplied with each other.\n");**

**else**

**{**

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

**for ( c = 0 ; c < p ; c++ )**

**for ( d = 0 ; d < q ; d++ )**

**scanf("%d", &second[c][d]);**

**for ( c = 0 ; c < m ; c++ )**

**{**

**for ( d = 0 ; d < q ; d++ )**

**{**

**for ( k = 0 ; k < p ; k++ )**

**{**

**sum = sum + first[c][k]\*second[k][d];**

**}**

**multiply[c][d] = sum;**

**sum = 0;**

**}**

**}**

**printf("Product of entered matrices:-\n");**

**for ( c = 0 ; c < m ; c++ )**

**{**

**for ( d = 0 ; d < q ; d++ )**

**printf("%d\t", multiply[c][d]);**

**printf("\n");**

**}**

**}**

**}**

1. **Determinant of 3x3 matrix**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int a[3][3],i,j;**

**long determinant;**

**printf("enter 9 elements in array:- \n");**

**for(i=0;i<3;i++)**

**{**

**for(j=0;j<3;j++)**

**{**

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

**}**

**}**

**printf("Array elements are - \n");**

**for(i=0;i<3;i++)**

**{**

**for(j=0;j<3;j++)**

**{**

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

**}**

**printf("\n");**

**}**

**determinant=a[0][0]\*((a[1][1]\*a[2][2])-(a[1][2]\*a[2][1]))**

**-a[0][1]\*((a[1][0]\*a[2][2])-(a[1][2]\*a[2][0]))**

**+a[0][2]\*((a[1][0]\*a[2][2])-(a[1][2]\*a[2][0]));**

**printf("\n Determinant of 3x3 matrix : %ld",determinant);**

**return 0;**

**}**