**1: Program to Check Vowel or consonant**

1. #include <stdio.h>
2. int main()
3. {
4. char c;
5. int isLowercaseVowel, isUppercaseVowel;
6. printf("Enter an alphabet: ");
7. scanf("%c",&c);
8. // evaluates to 1 (true) if c is a lowercase vowel
9. isLowercaseVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
10. // evaluates to 1 (true) if c is an uppercase vowel
11. isUppercaseVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
12. // evaluates to 1 (true) if either isLowercaseVowel or isUppercaseVowel is true
13. if (isLowercaseVowel || isUppercaseVowel)
14. printf("%c is a vowel.", c);
15. else
16. printf("%c is a consonant.", c);
17. return 0;
18. }

**2. C Program to Find ASCII Value of a Character**

1. #include <stdio.h>
2. int main()
3. {
4. char c;
5. printf("Enter a character: ");
6. // Reads character input from the user
7. scanf("%c", &c);
9. // %d displays the integer value of a character
10. // %c displays the actual character
11. printf("ASCII value of %c = %d", c, c);
12. return 0;
13. }

**3. C Program to check Leap Year**

#include <stdio.h>

int main()

{

int year;

printf("Enter a year: ");

scanf("%d",&year);

if(year%4 == 0)

{

if( year%100 == 0)

{

// year is divisible by 400, hence the year is a leap year

if ( year%400 == 0)

printf("%d is a leap year.", year);

else

printf("%d is not a leap year.", year);

}

else

printf("%d is a leap year.", year );

}

else

printf("%d is not a leap year.", year);

return 0;

}

**Output:**

Enter a year: 1900

1900 is not a leap year.

# 4. C Program to Find Factorial of a Number

1. #include <stdio.h>
2. int main()
3. {
4. int n, i;
5. unsigned long long factorial = 1;
6. printf("Enter an integer: ");
7. scanf("%d",&n);
8. // show error if the user enters a negative integer
9. if (n < 0)
10. printf("Error! Factorial of a negative number doesn't exist.");
11. else
12. {
13. for(i=1; i<=n; ++i)
14. {
15. factorial \*= i; // factorial = factorial\*i;
16. }
17. printf("Factorial of %d = %llu", n, factorial);
18. }
19. return 0;
20. }

## 5. Multiplication Table Up to 10

1. #include <stdio.h>
2. int main()
3. {
4. int n, i;
5. printf("Enter an integer: ");
6. scanf("%d",&n);
7. for(i=1; i<=10; ++i)
8. {
9. printf("%d \* %d = %d \n", n, i, n\*i);
10. }
12. return 0;
13. }

## 6. Fibonacci Series up to n number of terms

1. #include <stdio.h>
2. int main()
3. {
4. int i, n, t1 = 0, t2 = 1, nextTerm;
5. printf("Enter the number of terms: ");
6. scanf("%d", &n);
7. printf("Fibonacci Series: ");
8. for (i = 1; i <= n; ++i)
9. {
10. printf("%d, ", t1);
11. nextTerm = t1 + t2;
12. t1 = t2;
13. t2 = nextTerm;
14. }
15. return 0;
16. }

## 7. Fibonacci Sequence Up to a Certain Number

1. #include <stdio.h>
2. int main()
3. {
4. int t1 = 0, t2 = 1, nextTerm = 0, n;
5. printf("Enter a positive number: ");
6. scanf("%d", &n);
7. // displays the first two terms which is always 0 and 1
8. printf("Fibonacci Series: %d, %d, ", t1, t2);
9. nextTerm = t1 + t2;
10. while(nextTerm <= n)
11. {
12. printf("%d, ",nextTerm);
13. t1 = t2;
14. t2 = nextTerm;
15. nextTerm = t1 + t2;
16. }
18. return 0;
19. }

## 8. GCD Using while loop and if...else Statement

1. #include <stdio.h>
2. int main()
3. {
4. int n1, n2;
6. printf("Enter two positive integers: ");
7. scanf("%d %d",&n1,&n2);
8. while(n1!=n2)
9. {
10. if(n1 > n2)
11. n1 -= n2;
12. else
13. n2 -= n1;
14. }
15. printf("GCD = %d",n1);
16. return 0;
17. }

## 9. LCM using while Loop and if Statement

1. #include <stdio.h>
2. int main()
3. {
4. int n1, n2, minMultiple;
5. printf("Enter two positive integers: ");
6. scanf("%d %d", &n1, &n2);
7. // maximum number between n1 and n2 is stored in minMultiple
8. minMultiple = (n1>n2) ? n1 : n2;
9. // Always true
10. while(1)
11. {
12. if( minMultiple%n1==0 && minMultiple%n2==0 )
13. {
14. printf("The LCM of %d and %d is %d.", n1, n2,minMultiple);
15. break;
16. }
17. ++minMultiple;
18. }
19. return 0;
20. }

## 10. Reverse an Integer

1. #include <stdio.h>
2. int main()
3. {
4. int n, reversedNumber = 0, remainder;
5. printf("Enter an integer: ");
6. scanf("%d", &n);
7. while(n != 0)
8. {
9. remainder = n%10;
10. reversedNumber = reversedNumber\*10 + remainder;
11. n /= 10;
12. }
13. printf("Reversed Number = %d", reversedNumber);
14. return 0;
15. }

## 11. Program to Check Palindrome

1. #include <stdio.h>
2. int main()
3. {
4. int n, reversedInteger = 0, remainder, originalInteger;
5. printf("Enter an integer: ");
6. scanf("%d", &n);
7. originalInteger = n;
8. // reversed integer is stored in variable
9. while( n!=0 )
10. {
11. remainder = n%10;
12. reversedInteger = reversedInteger\*10 + remainder;
13. n /= 10;
14. }
15. // palindrome if orignalInteger and reversedInteger are equal
16. if (originalInteger == reversedInteger)
17. printf("%d is a palindrome.", originalInteger);
18. else
19. printf("%d is not a palindrome.", originalInteger);
21. return 0;
22. }

## 12. Check Armstrong Number

#include <stdio.h>

#include <math.h>

int main()

{

int number, originalNumber, remainder, result = 0, n = 0 ;

printf("Enter an integer: ");

scanf("%d", &number);

originalNumber = number;

while (originalNumber != 0)

{

originalNumber /= 10;

++n;

}

originalNumber = number;

while (originalNumber != 0)

{

remainder = originalNumber%10;

result += pow(remainder, n);

originalNumber /= 10;

}

if(result == number)

printf("%d is an Armstrong number.", number);

else

printf("%d is not an Armstrong number.", number);

return 0;

}

## 13. Factors of a Positive Integer

1. #include <stdio.h>
2. int main()
3. {
4. int number, i;
5. printf("Enter a positive integer: ");
6. scanf("%d",&number);
7. printf("Factors of %d are: ", number);
8. for(i=1; i <= number; ++i)
9. {
10. if (number%i == 0)
11. {
12. printf("%d ",i);
13. }
14. }
15. return 0;
16. }

### 14. Program to print full pyramid using \*

\*

\* \* \*

\* \* \* \* \*

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

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

**Source Code**

1. #include <stdio.h>
2. int main()
3. {
4. int i, space, rows, k=0;
5. printf("Enter number of rows: ");
6. scanf("%d",&rows);
7. for(i=1; i<=rows; ++i, k=0)
8. {
9. for(space=1; space<=rows-i; ++space)
10. {
11. printf(" ");
12. }
13. while(k != 2\*i-1)
14. {
15. printf("\* ");
16. ++k;
17. }
18. printf("\n");
19. }
21. return 0;
22. }

### 15. Program to print pyramid using 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

**Source Code**

1. #include <stdio.h>
2. int main()
3. {
4. int i, space, rows, k=0, count = 0, count1 = 0;
5. printf("Enter number of rows: ");
6. scanf("%d",&rows);
7. for(i=1; i<=rows; ++i)
8. {
9. for(space=1; space <= rows-i; ++space)
10. {
11. printf(" ");
12. ++count;
13. }
14. while(k != 2\*i-1)
15. {
16. if (count <= rows-1)
17. {
18. printf("%d ", i+k);
19. ++count;
20. }
21. else
22. {
23. ++count1;
24. printf("%d ", (i+k-2\*count1));
25. }
26. ++k;
27. }
28. c

### 16. Print Pascal's triangle

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

**Source Code**

1. #include <stdio.h>
2. int main()
3. {
4. int rows, coef = 1, space, i, j;
5. printf("Enter number of rows: ");
6. scanf("%d",&rows);
7. for(i=0; i<rows; i++)
8. {
9. for(space=1; space <= rows-i; space++)
10. printf(" ");
11. for(j=0; j <= i; j++)
12. {
13. if (j==0 || i==0)
14. coef = 1;
15. else
16. coef = coef\*(i-j+1)/j;
17. printf("%4d", coef);
18. }
19. printf("\n");
20. }
21. return 0;
22. }

## 17. Program to Add Two Matrices

1. #include <stdio.h>
2. int main(){
3. int r, c, a[100][100], b[100][100], sum[100][100], i, j;
4. printf("Enter number of rows (between 1 and 100): ");
5. scanf("%d", &r);
6. printf("Enter number of columns (between 1 and 100): ");
7. scanf("%d", &c);
8. printf("\nEnter elements of 1st matrix:\n");
9. for(i=0; i<r; ++i)
10. for(j=0; j<c; ++j)
11. {
12. printf("Enter element a%d%d: ",i+1,j+1);
13. scanf("%d",&a[i][j]);
14. }
15. printf("Enter elements of 2nd matrix:\n");
16. for(i=0; i<r; ++i)
17. for(j=0; j<c; ++j)
18. {
19. printf("Enter element a%d%d: ",i+1, j+1);
20. scanf("%d", &b[i][j]);
21. }
22. // Adding Two matrices
23. for(i=0;i<r;++i)
24. for(j=0;j<c;++j)
25. {
26. sum[i][j]=a[i][j]+b[i][j];
27. }
28. // Displaying the result
29. printf("\nSum of two matrices: \n");
30. for(i=0;i<r;++i)
31. for(j=0;j<c;++j)
32. {
33. printf("%d ",sum[i][j]);
34. if(j==c-1)
35. {
36. printf("\n\n");
37. }
38. }
40. return 0;
41. }

## 18. Program to Find Transpose of a Matrix

1. #include <stdio.h>
2. int main()
3. {
4. int a[10][10], transpose[10][10], r, c, i, j;
5. printf("Enter rows and columns of matrix: ");
6. scanf("%d %d", &r, &c);
7. // Storing elements of the matrix
8. printf("\nEnter elements of matrix:\n");
9. for(i=0; i<r; ++i)
10. for(j=0; j<c; ++j)
11. {
12. printf("Enter element a%d%d: ",i+1, j+1);
13. scanf("%d", &a[i][j]);
14. }
15. // Displaying the matrix a[][] \*/
16. printf("\nEntered Matrix: \n");
17. for(i=0; i<r; ++i)
18. for(j=0; j<c; ++j)
19. {
20. printf("%d ", a[i][j]);
21. if (j == c-1)
22. printf("\n\n");
23. }
24. // Finding the transpose of matrix a
25. for(i=0; i<r; ++i)
26. for(j=0; j<c; ++j)
27. {
28. transpose[j][i] = a[i][j];
29. }
30. // Displaying the transpose of matrix a
31. printf("\nTranspose of Matrix:\n");
32. for(i=0; i<c; ++i)
33. for(j=0; j<r; ++j)
34. {
35. printf("%d ",transpose[i][j]);
36. if(j==r-1)
37. printf("\n\n");
38. }
39. return 0;
40. }

### 19. Multiply Matrices by Passing it to a Function

#include <stdio.h>

void enterData(int firstMatrix[][10], int secondMatrix[][10], int rowFirst, int columnFirst, int rowSecond, int columnSecond);

void multiplyMatrices(int firstMatrix[][10], int secondMatrix[][10], int multResult[][10], int rowFirst, int columnFirst, int rowSecond, int columnSecond);

void display(int mult[][10], int rowFirst, int columnSecond);

**int main()**

**{**

int firstMatrix[10][10], secondMatrix[10][10], mult[10][10], rowFirst, columnFirst, rowSecond, columnSecond, i, j, k;

printf("Enter rows and column for first matrix: ");

scanf("%d %d", &rowFirst, &columnFirst);

printf("Enter rows and column for second matrix: ");

scanf("%d %d", &rowSecond, &columnSecond);

// If colum of first matrix in not equal to row of second matrix, asking user to enter the size of matrix again.

while (columnFirst != rowSecond)

{

printf("Error! column of first matrix not equal to row of second.\n");

printf("Enter rows and column for first matrix: ");

scanf("%d%d", &rowFirst, &columnFirst);

printf("Enter rows and column for second matrix: ");

scanf("%d%d", &rowSecond, &columnSecond);

}

// Function to take matrices data

enterData(firstMatrix, secondMatrix, rowFirst, columnFirst, rowSecond, columnSecond);

// Function to multiply two matrices.

multiplyMatrices(firstMatrix, secondMatrix, mult, rowFirst, columnFirst, rowSecond, columnSecond);

// Function to display resultant matrix after multiplication.

display(mult, rowFirst, columnSecond);

return 0;

}

void enterData(int firstMatrix[][10], int secondMatrix[][10], int rowFirst, int columnFirst, int rowSecond, int columnSecond)

{

int i, j;

printf("\nEnter elements of matrix 1:\n");

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

{

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

{

printf("Enter elements a%d%d: ", i + 1, j + 1);

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

}

}

printf("\nEnter elements of matrix 2:\n");

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

{

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

{

printf("Enter elements b%d%d: ", i + 1, j + 1);

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

}

}

}

void multiplyMatrices(int firstMatrix[][10], int secondMatrix[][10], int mult[][10], int rowFirst, int columnFirst, int rowSecond, int columnSecond)

{

int i, j, k;

// Initializing elements of matrix mult to 0.

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

{

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

{

mult[i][j] = 0;

}

}

// Multiplying matrix firstMatrix and secondMatrix and storing in array mult.

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

{

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

{

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

{

mult[i][j] += firstMatrix[i][k] \* secondMatrix[k][j];

}

}

}

}

void display(int mult[][10], int rowFirst, int columnSecond)

{

int i, j;

printf("\nOutput Matrix:\n");

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

{

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

{

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

if(j == columnSecond - 1)

printf("\n\n");

}

}

}

**Output**

Enter rows and column for first matrix: 3

2

Enter rows and column for second matrix: 3

2

Error! column of first matrix not equal to row of second.

Enter rows and column for first matrix: 2

3

Enter rows and column for second matrix: 3

2

Enter elements of matrix 1:

Enter elements a11: 3

Enter elements a12: -2

Enter elements a13: 5

Enter elements a21: 3

Enter elements a22: 0

Enter elements a23: 4

Enter elements of matrix 2:

Enter elements b11: 2

Enter elements b12: 3

Enter elements b21: -9

Enter elements b22: 0

Enter elements b31: 0

Enter elements b32: 4

Output Matrix:

24 29

6 25

## 20. Calculate Length of String without Using strlen() Function

1. #include <stdio.h>
2. int main()
3. {
4. char s[1000];
5. int i;
6. printf("Enter a string: ");
7. scanf("%s", s);
8. for(i = 0; s[i] != '\0'; ++i);
9. printf("Length of string: %d", i);
10. return 0;
11. }

**Output**

Enter a string: Programiz

Length of string: 9

## 21. Find the Frequency of Characters

1. #include <stdio.h>
2. int main()
3. {
4. char str[1000], ch;
5. int i, frequency = 0;
6. printf("Enter a string: ");
7. gets(str);
8. printf("Enter a character to find the frequency: ");
9. scanf("%c",&ch);
10. for(i = 0; str[i] != '\0'; ++i)
11. {
12. if(ch == str[i])
13. ++frequency;
14. }
15. printf("Frequency of %c = %d", ch, frequency);
16. return 0;
17. }

## 22. Program to count vowels, consonants etc.

1. #include <stdio.h>
2. int main()
3. {
4. char line[150];
5. int i, vowels, consonants, digits, spaces;
6. vowels = consonants = digits = spaces = 0;
7. printf("Enter a line of string: ");
8. scanf("%[^\n]", line);
9. for(i=0; line[i]!='\0'; ++i)
10. {
11. if(line[i]=='a' || line[i]=='e' || line[i]=='i' ||
12. line[i]=='o' || line[i]=='u' || line[i]=='A' ||
13. line[i]=='E' || line[i]=='I' || line[i]=='O' ||
14. line[i]=='U')
15. {
16. ++vowels;
17. }
18. else if((line[i]>='a'&& line[i]<='z') || (line[i]>='A'&& line[i]<='Z'))
19. {
20. ++consonants;
21. }
22. else if(line[i]>='0' && line[i]<='9')
23. {
24. ++digits;
25. }
26. else if (line[i]==' ')
27. {
28. ++spaces;
29. }
30. }
31. printf("Vowels: %d",vowels);
32. printf("\nConsonants: %d",consonants);
33. printf("\nDigits: %d",digits);
34. printf("\nWhite spaces: %d", spaces);
35. return 0;
36. }

## 23. Concatenate Two Strings Without Using strcat()

1. #include <stdio.h>
2. int main()
3. {
4. char s1[100], s2[100], i, j;
5. printf("Enter first string: ");
6. scanf("%s", s1);
7. printf("Enter second string: ");
8. scanf("%s", s2);
9. // calculate the length of string s1
10. // and store it in i
11. for(i = 0; s1[i] != '\0'; ++i);
12. for(j = 0; s2[j] != '\0'; ++j, ++i)
13. {
14. s1[i] = s2[j];
15. }
16. s1[i] = '\0';
17. printf("After concatenation: %s", s1);
18. return 0;
19. }

## 24. Copy String Manually Without Using strcpy()

1. #include <stdio.h>
2. int main()
3. {
4. char s1[100], s2[100], i;
5. printf("Enter string s1: ");
6. scanf("%s",s1);
7. for(i = 0; s1[i] != '\0'; ++i)
8. {
9. s2[i] = s1[i];
10. }
11. s2[i] = '\0';
12. printf("String s2: %s", s2);
13. return 0;
14. }

## 25. Store Information and Display it Using Structure

1. #include <stdio.h>
2. struct student
3. {
4. char name[50];
5. int roll;
6. float marks;
7. } s;
8. int main()
9. {
10. printf("Enter information:\n");
11. printf("Enter name: ");
12. scanf("%s", s.name);
13. printf("Enter roll number: ");
14. scanf("%d", &s.roll);
15. printf("Enter marks: ");
16. scanf("%f", &s.marks);
17. printf("Displaying Information:\n");
18. printf("Name: ");
19. puts(s.name);
20. printf("Roll number: %d\n",s.roll);
21. printf("Marks: %.1f\n", s.marks);
22. return 0;
23. }

**26. C program to read name and marks of n number of students from and store them in a file. If the file previously exits, add the information to the file.**

1. #include <stdio.h>
2. int main()
3. {
4. char name[50];
5. int marks, i, num;
6. printf("Enter number of students: ");
7. scanf("%d", &num);
8. FILE \*fptr;
9. fptr = (fopen("C:\\student.txt", "a"));
10. if(fptr == NULL)
11. {
12. printf("Error!");
13. exit(1);
14. }
15. for(i = 0; i < num; ++i)
16. {
17. printf("For student%d\nEnter name: ", i+1);
18. scanf("%s", name);
19. printf("Enter marks: ");
20. scanf("%d", &marks);
21. fprintf(fptr,"\nName: %s \nMarks=%d \n", name, marks);
22. }
23. fclose(fptr);
24. return 0;
25. }

# 27. C Program to Copy File into Another File

This C Program copies a file into another file.

Here is source code of the C Program to copy a file into another file. The C program is successfully compiled and run on a Linux system. The program output is also shown below.

1. /\*
2. \* C Program to Copy a File into Another File
3. \*/
4. #include <stdio.h>
6. void main(int argc,char \*\*argv)
7. {
8. FILE \*fp1, \*fp2;
9. char ch;
10. int pos;
12. if ((fp1 = fopen(argv[1],"r")) == NULL)
13. {
14. printf("\nFile cannot be opened");
15. return;
16. }
17. else
18. {
19. printf("\nFile opened for copy...\n ");
20. }
21. fp2 = fopen(argv[2], "w");
22. fseek(fp1, 0L, SEEK\_END); // file pointer at end of file
23. pos = ftell(fp1);
24. fseek(fp1, 0L, SEEK\_SET); // file pointer set at start
25. while (pos--)
26. {
27. ch = fgetc(fp1); // copying file character by character
28. fputc(ch, fp2);
29. }
30. fcloseall();
31. }