**1. Write a program which reads in 10 integers from the user and stores them in an array. Find the largest value in the array and print it.**

#include<stdio.h>

int main() {

int a[10],max=0;

printf("Enter array ");

for(int i=0;i<10;i++)

{

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

if(a[i]>max)

max=a[i];

}

printf("%d",max);

return 0;

}

**2. Modify the last program to use a preprocessor constant for the size of the array and in the test condition of the loop which processes the array.**

#include <stdio.h>

#define size 10

int main(){

int a[size];

int large;

for(int i=0;i<10;i++){

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

}

large=a[0];

for(int i=1;i<10;i++){

if (a[i]>large){

large=a[i];

}

}

printf("%d",large);

return 0;}

**3. Modify the last program to find mean of n numbers using arrays.**

#include <stdio.h>

#define size 10

int main(){

int a[size];

int sum,mean;

for(int i=0;i<10;i++){

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

}

for(int i=0;i<10;i++){

sum+=a[i];

}

mean=sum/10;

printf("%d",mean);

return 0;}

**4. Write a program to interchange the largest and the smallest number in the array.**

#include<stdio.h>

int main() {

int a[10],max=0,c,min=9,d;

printf("Enter array ");

for(int i=0;i<10;i++)

{

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

if(a[i]>max)

{

max=a[i];

c=i;

}

if(a[i]<min)

{

min=a[i];

d=i;

}

}

a[d]=max;

a[c]=min;

for (int i=0;i<10;i++)

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

return 0;

}

**5. Write a program to find the second biggest number using an array of n numbers.**

#include <stdio.h>

int main(){

int n;

scanf("%d",&n);

int i,j,a,arr[n];

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

scanf("%d",&arr[i]);}

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

for(j=i+1;j<n;j++){

if(arr[i]>arr[j]){

a=arr[i];

arr[i]=arr[j];

arr[j]=a;

}

}

}

printf("%d\n", arr[n-2]);

return 0;

}

**6. Write a program to find whether the array of integers contain a duplicate number. If it’s there print the position of duplicate numbers.**

#include<stdio.h>

int main() {

int a[10];

printf("Enter array ");

for(int i=0;i<10;i++)

{

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

}

printf("The indexes of duplicates are ");

for(int i=0;i<10;i++)

{

for(int j=i+1;j<10;j++)

{

if(a[i]==a[j])

printf("%d %d\n",i,j);

}

}

return 0;

}

**7. Write a program which can store 10 integers in an array. Fill the array with “random” numbers using the library functions rand() instead of reading them from the user. Find the largest element in**

**the array and print it out.**

**Each time rand() is called it returns a “random” integer. Use the mod operator ( % ) to get a value in the desired range. For example:**

**int result; result = rand()**

**% 1000;**

**will assign a random value in the range 0 – 999 to the variable result. Make sure your program contains the line:**

**#include <stdlib.h>**

**to include information about the rand() function.**

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[10],max=0;

for(int i=0;i<10;i++)

{

a[i]=rand()%1000;

}

printf("The largest value is ");

for(int i=0;i<10;i++)

{

if(a[i]>max)

{

max=a[i];

}

}

printf("%d",max);

return 0;

}

**8. Modify the last program so that instead of finding the largest element in the array, the program sorts the elements of the array into ascending order.**

#include <stdio.h>

int main(){

int n;

scanf("%d",&n);

int i,j,a,arr[n];

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

a[i]=rand()%1000;

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

for(j=i+1;j<n;j++){

if(arr[i]>arr[j]){

a=arr[i];

arr[i]=arr[j];

arr[j]=a;

}

}

}

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

printf("%d",arr[i]);}

return 0;

}

**9. Write a program to find out whether a particular element is in the integer array using Linear search.**

#include <stdio.h>

int main(){

int a[10];

int search,i;

for(int i=0;i<10;i++){

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

}

printf("Enter value to search for ");

scanf("%d",&search);

for(i=0;i<10;i++){

if (a[i]==search){

break;

}

}

if(i!=10)

printf("%d",i);

else

printf("Not found");

return 0;}

**10. Write a program to find out whether a particular element is in the integer array using Binary search.**

#include <stdio.h>

int bin\_search(int a[],int beg,int end,int val)

{

int mid;

if(end>=beg)

{

mid=(beg+end)/2;

if(a[mid]==val)

{

return mid+1;

}

else if(a[mid]<val)

{

return bin\_search(a,mid+1,end,val);

}

else

{

return bin\_search(a,beg,mid-1,val);

}

}

return -1;

}

int main(){

int a[10];

int val,f;

for(int i=0;i<10;i++){

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

}

printf("Enter value to search for ");

scanf("%d",&val);

f=bin\_search(a,0,9,val);

if(f==-1){

printf("Not found");}

else{

printf("%d position",f);}

return 0;

}

**11. Write a program to sort an array of elements using Bubble sort.**

#include <stdio.h>

int main(){

int n;

scanf("%d",&n);

int arr[n];

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

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

for (int step = 0; step < n - 1; ++step)

{

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

{

if (arr[i] > arr[i + 1])

{

int temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

}

}

}

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

printf("%d",arr[i]);}

return 0;

}

**12. Write a program to sort an array of elements using Selection sort.**

#include <stdio.h>

int main(){

int n;

scanf("%d",&n);

int i,j,a,arr[n];

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

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

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

for(j=i+1;j<n;j++){

if(arr[i]>arr[j]){

a=arr[i];

arr[i]=arr[j];

arr[j]=a;

}

}

}

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

printf("%d",arr[i]);}

return 0;

}

**13. Read a value k from the user and using k left rotation and right rotation depending on the user choice Left or Right, print the rotated value. [Use function for the rotations]**

#include <stdio.h>

void leftRotatebyOne(int arr[], int n)

{

int temp = arr[0], i;

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

arr[i] = arr[i + 1];

arr[i] = temp;

}

void leftRotate(int arr[], int k, int n)

{

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

leftRotatebyOne(arr, n);

}

void rightRotatebyOne(int arr[], int n)

{

int temp = arr[n - 1], i;

for (i = n - 1; i > 0; i--)

arr[i] = arr[i - 1];

arr[0] = temp;

}

void rightRotate(int arr[], int k, int n)

{

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

rightRotatebyOne(arr, n);

}

void printArray(int arr[], int n)

{

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

printf("%d ", arr[i]);

}

int main()

{

int n, k;

char choice[5];

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array: ");

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

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

printf("Enter the value of k: ");

scanf("%d", &k);

printf("Enter left or right for rotation direction: ");

scanf("%s", choice);

if (strcmp(choice, "left") == 0)

leftRotate(arr, k, n);

else if (strcmp(choice, "right") == 0)

rightRotate(arr, k, n);

else

printf("Invalid choice\n");

printf("Array after rotation: ");

printArray(arr, n);

return 0;

}

**14. Read an array of size n and a variable k from user. find all the pairs of elements in the array which yields a sum as k.**

#include <stdio.h>

int main() {

int n, k;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d elements: ", n);

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

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

}

printf("Enter the value of k: ");

scanf("%d", &k);

printf("Pairs with sum equal to %d:\n", k);

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

for (int j = i + 1; j < n; j++) {

if (arr[i] + arr[j] == k) {

printf("(%d, %d)\n", arr[i], arr[j]);

}

}

}

return 0;

}

2-D Array

15. Write a program to read and display a matrix.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3];

printf("Enter matrix elements ");

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

{

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

{

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

}

}

printf("The matrix is ");

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

{

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

{

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

}

printf("\n");

}

return 0;

}

16. Write a program to add two matrices.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],b[3][3],c[3][3];

printf("Enter matrix elements ");

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

{

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

{

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

}

}

printf("Enter elements of second matrix ");

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

{

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

{

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

}

}

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

{

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

{

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

}

}

printf("The sum of two matrix is \n");

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

{

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

{

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

}

printf("\n");

}

return 0;

}

17. Write a program to find the transpose of a matrix.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],b[3][3];

printf("Enter matrix elements ");

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

{

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

{

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

}

}

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

{

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

{

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

}

}

printf("The transpose of the matrix is \n");

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

{

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

{

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

}

printf("\n");

}

return 0;

}

18. Write a program to find the sum of all the elements in a 2D array.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],b=0;

printf("Enter matrix elements ");

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

{

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

{

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

}

}

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

{

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

{

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

}

}

printf("The sum of all elements in the matrix is %d",b);

return 0;

}

19. Write a program to find the sum of the elements in each row of a 2D array and print it.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],b[3];

printf("Enter matrix elements ");

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

{

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

{

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

}

}

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

{

b[i]=0;

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

{

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

}

}

printf("The sum of each row of the matrix is \n");

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

{

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

}

return 0;

}

20. Write a program to fill a square matrix with value 0 on the diagonal, 1 on the upper right triangle and -1 on the lower left triangle.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],f=0;

printf("Enter matrix elements ");

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

{

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

{

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

}

}

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

{

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

{

if(i<j)

{

if(a[i][j]==0)

continue;

else

f=1;

}

}

}

if(f==0)

printf("1");

else

printf("-1");

return 0;

}

21. Write a program to multiply two matrices.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],b[3][3],c[3][3];

printf("Enter matrix elements ");

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

{

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

{

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

}

}

printf("Enter 2nd matrix elements ");

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

{

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

{

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

}

}

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

{

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

{

c[i][j]=0;

for (int x = 0; x < 3; x++)

{

c[i][j]=c[i][j]+a[i][x]\*b[x][j];

}

}

}

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

{

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

{

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

}

printf("\n");

}

return 0;

}

22. Write a program to find out whether a particular element is in the 2D integer array and print its row and column value using call by reference.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],l;

printf("Enter matrix elements ");

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

{

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

{

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

}

}

printf("Enter the element ");

scanf("%d",&l);

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

{

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

{

if(a[i][j]==l)

{

printf(" %d %d \n",i,j);

}

}

}

return 0;

}

23. Write a program to interchange any two Rows & Columns in the given Matrix.

#include<stdio.h>

#include<stdlib.h>

int main() {

int a[3][3],temp[3],c,d,g;

printf("Enter matrix elements ");

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

{

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

{

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

}

}

printf("Enter 1 for row exchange and 0 for column exchange ");

scanf("%d",&c);

printf("Enter the indexes to exchange ");

scanf("%d%d",&d,&g);

if(c==1)

{

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

{

temp[i]=a[d][i];

a[d][i]=a[g][i];

a[g][i]=temp[i];

}

}

else

{

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

{

temp[i]=a[i][d];

a[i][d]=a[i][g];

a[i][g]=temp[i];

}

}

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

{

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

{

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

}

printf("\n");

}

return 0;

}

24. Write a program to Sort Rows of the Matrix in Ascending & Columns in Descending Order.

Sample Output

Enter the order of the matrix

3 3

Enter co-efficients of the matrix

3 7 9

2 4 8

5 2 6

The given matrix is

3 7 9

2 4 8

5 2 6

After arranging rows in ascending order

3 7 9

2 4 8

2 5 6

After arranging the columns in descending order

5 7 9

3 4 8

2 2 6

#include<stdio.h>

#define MAXROWS 100

#define MAXCOLS 100

/\* Function to sort rows and columns of a 2D array

in ascending and descending order respectively \*/

void SortRowsCols(int mat[][MAXCOLS], int m, int n)

{

// sorting columns in descending order

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

for (int j = 0; j < n - 1 - i; j++) {

for (int k = 0; k < m; k++) {

/\* If left element is greater than right element,

swap them \*/

if (mat[k][j] < mat[k][j + 1]) {

int temp = mat[k][j];

mat[k][j] = mat[k][j + 1];

mat[k][j + 1] = temp;

}

}

}

}

// sorting rows in ascending order

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

for (int j = 0; j < m - 1 - i; j++) {

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

/\* If left element is greater than right element,

swap them \*/

if (mat[j][k] > mat[j + 1][k]) {

int temp = mat[j][k];

mat[j][k] = mat[j + 1][k];

mat[j + 1][k] = temp;

}

}

}

}

}

int main()

{

int m, n;

int mat[MAXROWS][MAXCOLS];

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

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

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

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

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

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

// function call

SortRowsCols(mat, m, n);

printf("\nMatrix after sorting : \n");

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

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

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

printf("\n");

}

return 0;

}

25. Write a program to do the Sum of the Main & Opposite Diagonal Elements of a MxN Matrix.

Sample Output

Enter the order of the matix

2 2

Enter the co-efficients of the matrix

40 30

38 90

The given matrix is

40 30

38 90

The sum of the main diagonal elements is = 130

The sum of the off diagonal elements is = 68

#include<stdio.h>

int main()

{

int n, m;

printf("Enter the size of the matrix\n");

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

int arr[m][n];

int i, j;

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

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

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

}

}

int maindiagonalsum = 0;

int oppdiagonalsum = 0;

// Main Diagonal

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

maindiagonalsum += arr[i][i];

// Opposite Diagonal

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

oppdiagonalsum += arr[i][m-i-1];

printf("Sum of Main Diagonal elements: %d\n", maindiagonalsum);

printf("Sum of Opposite Diagonal elements: %d", oppdiagonalsum);

return 0;

}

Strings

Reading Strings

26. If we declare a string by writing char str[50]; Then str can be read from the user by using three

ways:

1. Using scanf() function

2. Using gets() function

3. Using getchar() function repeatedly

Write program to read a string in the above three ways.

Writing strings

#include <stdio.h>

//Left Rotate

int main() {

int a[5],c,g,h;

printf("Enter array ");

for(int i=0;i<4;i++)

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

printf("Enter number of rotations ");

scanf("%d",&g);

for (int f=0;f<g;f++)

{

printf("Enter selection 0 for left and 1 for right rotation ");

scanf("%d",&h);

if(h==0)

{

c=a[0];

a[4]=c;

printf("New array ");

for(int i=1;i<5;i++)

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

}

//Rotate Right

else if(h==1)

{

c=a[4];

a[0]=c;

printf("New array ");

for(int i=0;i<4;i++)

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

}

else

printf("Wrong input");

}

return 0;

}

27. The string can be displayed on screen using three ways:

1. Using printf() function

2. Using puts() function

3. Using putchar()function repeatedly.

Modify the above program to display the string that you read.

#include <stdio.h>

#include<string.h>

int main()

{

char str[100];

printf("Enter a string: \n");

gets(str);

// Using printf() function

printf("The entered string using printf() is: %s\n",str);

// Using puts() function

puts("The entered string using puts() is:");

puts(str);

// Using putchar()function repeatedly.

int i;

printf("The entered string using putchar() is :");

for (i=0; i<strlen(str); i++)

putchar(str[i]);

return 0;

}

28. Run the following program and analyze the result.

#include<stdio.h> main()

{

char str = “Hello”; printf(“\n

%s”,str); printf(“\n

%s”,&str);

printf(“\n%s”,&str[2]);

}

Output:

Hello

Hello

llo

29. Run the following program and analyze the result. It’s about the use of width and

precision specifications along with %s .

#include<stdio.h> main()

{

char str[] = “Introduction to C”;

printf(“\n |%s|”,str); printf(“\n

|%20s|”,str); printf(“\n |%-

20s|”,str); printf(“\n |%.4s|”,str);

printf(“\n |%20.4s|”,str);

printf(“\n |%-20.4s|”,str);

}

Output:

|Introduction to C|

| Introduction to C|

|Introduction to C |

|Intr|

| Intr|

|Intr |

30. Write a program to find the length of a string. Also use strlen( ) to do the same.

#include <stdio.h>

#include <string.h>

int main()

{

char str[100];

// Input string from user

printf("Enter any string: ");

scanf("%s", str);

// Find string length without using library function

int length = 0;

while (str[length] != '\0')

length++;

printf("Length of the given string %s is %d.\n", str, length);

// Also find length using library function

printf("Length of the given string %s is %ld.\n",

str, strlen(str));

return 0;

}

31. Write a program to copy on string to another without using any string library functions. Do the

same operation using strcpy() function in string.h)

Without using any library functions:

#include <stdio.h>

#include <string.h>

int main()

{

char str1[100], str2[100];

int i;

printf("Enter a string: ");

gets(str1);

for (i = 0; str1[i] != '\0'; ++i)

str2[i] = str1[i];

str2[i] = '\0';

printf("Copied String: %s", str2);

return 0;

}

Using strcpy() function:

#include <stdio.h>

#include <string.h>

int main()

{

char str1[100], str2[100];

printf("Enter a string: ");

gets(str1);

strcpy(str2, str1);

printf("Copied string: %s", str2);

return 0;

}

32. Write a program to convert characters of a string to upper case.

Note: Recall that ASCII code for A-Z varies from 65 to 91 and the ASCII code for a-z

ranges from 97 to 123

#include <stdio.h>

#include <string.h>

void convertToUpperCase(char \*str)

{

int i = 0;

while (str[i] != '\0') {

if (str[i] >= 'a' && str[i] <= 'z') {

str[i] = str[i] - 32;

}

i++;

}

}

// Driver code

int main()

{

char str[100];

scanf("%d",&str);

convertToUpperCase(str);

printf("%s", str);

return 0;

}

33. Write a program to concatenate two strings. (Do the same operation using the string library

function strcat() and analyze the behavior; you should include string .h)

#include<stdio.h>

#include<string.h>

int main()

{

char str1[100], str2[50];

printf("\nEnter the first string :");

scanf("%s",str1);

printf("\nEnter the second string :");

scanf("%s",str2);

// Concatenating both the string

strcat(str1, str2);

printf("\nString obtained on concatenation is %s", str1);

return 0;

}

#include <stdio.h>

#include <string.h>

void concat(char \*dest, const char \*src) {

int i, j;

for (i = 0; dest[i] != '\0'; i++);

for (j = 0; src[j] != '\0'; j++) {

dest[i+j] = src[j];

}

dest[i+j] = '\0';

}

int main() {

char str1[100];

char str2[100];

scanf("%s",str1);

scanf("%s",str2);

concat(str1, str2);

printf("Concatenated string: %s\n", str1);

return 0;

}

34. Write a program to compare two strings. (Do the same operation using the string library function strcmp() and analyze the behavior; you should include string .h)

#include <stdio.h>

#include <string.h>

int main()

{

char str1[20];

char str2[20];

printf("Enter first string: ");

scanf("%s",str1);

printf("Enter second string: ");

scanf("%s",str2);

int result;

result = strcmp(str1, str2);

if(result == 0)

{

printf("Strings are equal.\n");

}

else if(result > 0)

{

printf("First string is greater than the second.\n");

}

else

{

printf("Second string is greater than the first.\n");

}

return 0;

}

35. Write a program to check whether the entered string is a palindrome or not.

#include <stdio.h>

#include <string.h>

int main()

{

char string[25], revstring[25] = {'\0'};

int i, length = 0, flag = 0;

printf("Enter a string \n");

scanf(“%s”,string);

/\* keep going through each character of the string till its end \*/

for (i = 0; string[i] != '\0'; i++)

{

length++;

}

for (i = length - 1; i >= 0 ; i--)

{

revstring[length - i - 1] = string[i];

}

/\* Compare the input string and its reverse \*/

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

{

if (revstring[i] == string[i])

flag = 1;

else

flag = 0;

}

if (flag == 1)

printf ("Entered string is a palindrome \n");

else

printf ("Entered string is not a palindrome \n");

return 0;

}