**10. Write a C program to find maximum and minimum between two numbers using functions.**

#include <stdio.h>

/\* Function declarations \*/

int max(int num1, int num2);

int min(int num1, int num2);

int main()

{

int num1, num2, maximum, minimum;

/\* Input two numbers from user \*/

printf("Enter any two numbers: ");

scanf("%d%d", &num1, &num2);

maximum = max(num1, num2); // Call maximum function

minimum = min(num1, num2); // Call minimum function

printf("\nMaximum = %d\n", maximum);

printf("Minimum = %d", minimum);

return 0;

}

/\*\*

\* Find maximum between two numbers.

\*/

int max(int num1, int num2)

{

return (num1 > num2 ) ? num1 : num2;

}

/\*\*

\* Find minimum between two numbers.

\*/

int min(int num1, int num2)

{

return (num1 > num2 ) ? num2 : num1;

}

**Output**

Enter any two numbers: 8

96

Maximum = 96

Minimum = 8

**11. Write C program to find GCD of two integers by using recursive function.**

#include <stdio.h>

int hcf(int n1, int n2);

int main() {

int n1, n2;

printf("Enter two positive integers: ");

scanf("%d %d", &n1, &n2);

printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));

return 0;

}

int hcf(int n1, int n2) {

if (n2 != 0)

return hcf(n2, n1 % n2);

else

return n1;

}

**OUTPUT**

Enter two positive integers: 6

9

G.C.D of 6 and 9 is 3.

## **12. Iterative program to find the smallest and largest elements in an array**

#include <stdio.h

int main()

{

int arr[100], n, i, small, large;

printf("Enter the number of elements you want to insert : ");

scanf("%d", &n);

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

{

printf("Enter element %d : ", i + 1);

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

}

small = arr[0];

large = arr[0];

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

{

if (arr[i] < small)

{

small = arr[i];

}

if (arr[i] > large)

{

large = arr[i];

}

}

printf("\nLargest element is : %d", large);

printf("\nSmallest element is : %d", small);

return 0;

}

**OUTPUT**

Enter the number of elements you want to insert : 5

Enter element 1 : 9

Enter element 2 : 6

Enter element 3 : 7

Enter element 4 : 8

Enter element 5 : 4

Largest element is : 9

Smallest element is : 4

**13. Develop, implement and execute a C program that reads two matrices A (m x n) and B (p x q) and Compute product of matrices A and B. Read matrix A and matrix B in row major order and in column major order respectively.**

#include<stdio.h>

int main() {

int m, n, p, q, i, j, k;

int a[10][10], b[10][10], res[10][10];

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

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

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

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

if (n != p) {

printf("Matrix is incompatible for multiplication\n");

} else {

printf("Enter the elements of Matrix-A:\n");

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

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

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

}

}

printf("Enter the elements of Matrix-B:\n");

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

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

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

}

}

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

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

res[i][j] = 0;

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

res[i][j] += a[i][k] \* b[k][j];

}

}

}

printf("The product of the two matrices is:-\n");

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

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

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

}

printf("\n");

}

}

return 0;

}

**OUTPUT**

Enter the order of first matrix

3

2

Enter the order of second matrix

2

3

Enter the elements of Matrix-A:

1

2

3

4

3

6

Enter the elements of Matrix-B:

8

7

5

6

2

3

The product of the two matrices is:-

20 11 11

48 29 27

60 33 33

**14. Write a Program for deletion of an element from the specified location from Array**

#include <stdio.h>

int main()

{

int array[100], position, c, n

printf("Enter number of elements in array\n");

scanf("%d", &n)

printf("Enter %d elements\n", n)

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

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

printf("Enter the location where you wish to delete element\n");

scanf("%d", &position);

if ( position >= n+1 )

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

else

{

for ( c = position - 1 ; c < n - 1 ; c++ )

array[c] = array[c+1];

printf("Resultant array is\n");

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

printf("%d\n", array[c]);

}

return 0;

}

**OUTPUT**

Enter number of elements in array

5

Enter 5 elements

9

8

7

4

5

Enter the location where you wish to delete element

3

Resultant array is

9

8

4

5

**15. Write a C program using user defined functions to determine whether the given string is palindrome or not.**

#include <string.h>

int main()

{

char s[1000];

int i,n,c=0;

printf("Enter the string : ");

gets(s);

n=strlen(s);

for(i=0;i<n/2;i++)

{

if(s[i]==s[n-i-1])

c++;

}

if(c==i)

printf("string is palindrome");

else

printf("string is not palindrome");

return 0;

}

**Output**

Enter the string : HELLO

string is not palindrome

#include<stdio.h>

int checkPalindromeStr(char []);

int main()

{

char str[50];

int chk;

printf("Enter any String (word): ");

scanf("%s", str);

chk = checkPalindromeStr(str);

if(chk==1)

printf("\nIt's not a Palindrome String");

else

printf("\nIt's a Palindrome String");

return 0;

}

int checkPalindromeStr(char str[])

{

int i, j, len;

len = 0;

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

len++;

for(i=0, j=(len-1); i<=(len-1); i++, j--)

{

if(str[i] != str[j])

return 1; }

return 0;

}

**OUTPUT**

Enter any String (word): RADAR

It's a Palindrome String

**16. Write C program to count the number of lines, words and characters in a given text.**

#include<stdio.h>

int main()

{

// declare variables

char str[200];

int line, word, ch;

// initialize count variables with zero

line = word = ch = 0;

// read multiline string

printf("Enter string terminated with \*:\n");

scanf("%[^\*]", str);

// check every character

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

{

// if it is new line then

// one line and one word completed

if(str[i]=='\n')

{

line++;

word++;

}

// else it is a character

else

{

// if character is space or tab

// then one word is also completed

if(str[i]==' '||str[i]=='\t')

{

word++;

ch++;

}

// it was not '\n', sapace or tab

// it is a normal character

else {

ch++;

}

}

}

// display count values

printf("\nCharacter counts = %d\n", ch);

printf("Word counts = %d\n", word);

printf("Line counts = %d\n", line);

return 0;

}

**OUTPUT**

Enter string terminated with \*:

HI HOW ARE YOU

COUNT CHAR,LINE,WORD USING STRING FUNCTION.

\*

Character counts = 57

Word counts = 9

Line counts = 2

**17. Write a program to swap two numbers using a function. Pass the values to be swapped to this function using the call-by-value method and call-by-reference method.**

CALL BY VALUE

#include <stdio.h>

void swap(int , int); //prototype of the function

int main()

{

int a = 10;

int b = 20;

printf("Before swapping the values in main a = %d, b = %d\n",a,b); // printing the value of a and b in main

swap(a,b);

printf("After swapping values in main a = %d, b = %d\n",a,b); // The value of actual parameters do not change by changing the formal parameters in call by value, a = 10, b = 20

}

void swap (int a, int b)

{

int temp;

temp = a;

a=b;

b=temp;

printf("After swapping values in function a = %d, b = %d\n",a,b); // Formal parameters, a = 20, b = 10

}

**OUTPUT**

Before swapping the values in main a = 10, b = 20

After swapping values in function a = 20, b = 10

After swapping values in main a = 10, b = 20

CALL BY REFERENCE

#include <stdio.h>

void swap(int \*, int \*); //prototype of the function

int main()

{

int a = 10;

int b = 20;

printf("Before swapping the values in main a = %d, b = %d\n",a,b);

// printing the value of a and b in main

swap(&a,&b);

printf("After swapping values in main a = %d, b = %d\n",a,b);

// The values of actual parameters do change in call by reference, a = 10, b = 20

}

void swap (int \*a, int \*b)

{

int temp;

temp = \*a;

\*a=\*b;

\*b=temp;

printf("After swapping values in function a = %d, b = %d\n",\*a,\*b); // Formal parameters, a = 20, b = 10

}

**OUTPUT**

Before swapping the values in main a = 10, b = 20

After swapping values in function a = 20, b = 10

After swapping values in main a = 20, b = 10

**18. Write a C program to find the length of the string using Pointer.**

#include<stdio.h>

int string\_ln(char\*);

void main() {

char str[20];

int length;

printf("\nEnter any string : ");

gets(str);

length = string\_ln(str);

printf("The length of the given string %s is : %d", str, length);

}

int string\_ln(char\*p) /\* p=&str[0] \*/

{

int count = 0;

while (\*p != '\0') {

count++;

p++;

}

return count;

}

**OUTPUT**

Enter any string : HELLO

The length of the given string HELLO is : 5

**19. Write a program to copy one array to another using pointer.**

#include<stdio.h>

#define MAX 5

void print\_array(int arr[], int n);

void main()

{

int arr1[MAX], arr2[MAX];

int n, i;

int \*ptr1, \*ptr2;

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

scanf("%d",&n);

printf("\nEnter elements in the array\n");

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

{

scanf("%d",&arr1[i]);

}

ptr1 = arr1; //pointer ptr1 points to arr1[0]

ptr2 = arr2; //pointer ptr2 points to arr2[0]

printf("\nSource array before copying: ");

print\_array(arr1, n);

printf("\nDestination array before copying: ");

print\_array(arr2, n);

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

{

\*ptr2 = \*ptr1;

\*ptr1++;

\*ptr2++;

}

printf("\n\nSource array after copying: ");

print\_array(arr1, n);

printf("\nDestination array after copying: ");

print\_array(arr2, n);

}

void print\_array(int \*arr, int n)

{

int i;

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

{

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

}

}

**OUTPUT**

Enter the size of the array :5

Enter elements in the array

7

8

9

4

5

Source array before copying: 7 8 9 4 5

Destination array before copying: 0 0 0 0 0

Source array after copying: 7 8 9 4 5

Destination array after copying: 7 8 9 4 5

**20. Write a program to compare two strings using pointers.**

#include <stdio.h>

int main()

{

//Compare Two strigs using Pointers

char string1[50],string2[50],\*str1,\*str2;

int i,equal = 0;

printf("Enter The First String: ");

scanf("%s",string1);

printf("Enter The Second String: ");

scanf("%s",string2);

str1 = string1;

str2 = string2;

while(\*str1 == \*str2)

{

if ( \*str1 == '\0' || \*str2 == '\0' )

break;

str1++;

str2++;

}

if( \*str1 == '\0' && \*str2 == '\0' )

printf("\n\nBoth Strings Are Equal.");

else

printf("\n\nBoth Strings Are Not Equal.");

}

**OUTPUT**

Enter The First String: HELLO

Enter The Second String: HELLO

Both Strings Are Equal.