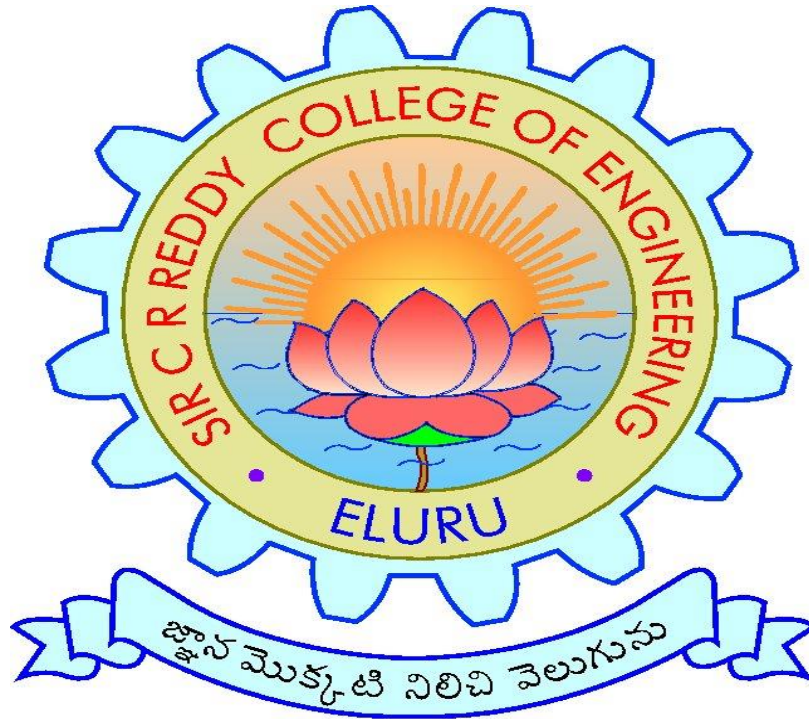


SIR C R REDDY COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL



Lab Name :Programming for Problem Solving using C

Regulation :R19&R20

Branch :CSE

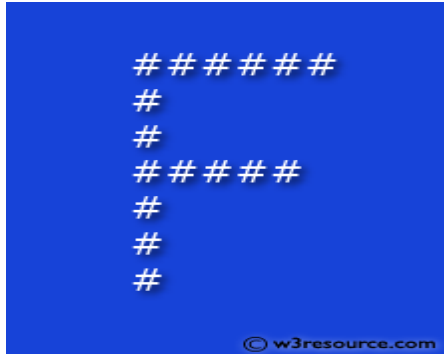
R20 Programming Problem solving Using C

Exercise 1:

1. Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters.

Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters.

Pictorial Presentation:



C Code:

```
#include <stdio.h>
int main()
{
    printf("#####\n");
    printf("#\n");
    printf("#\n");
    printf("#####\n");
    printf("#\n");
    printf("#\n");
    printf("#\n");
    return(0);
}
```

Copy

Sample Output:

```
#####
#
#
#####
#
#
#
```

2. Write a C program to compute the perimeter and area of a rectangle with a height of 7 inches and width of 5 inches.

```
#include <stdio.h>
```

```

int main()
{
    float length, width, perimeter;

    printf("Enter length of the rectangle: ");
    scanf("%f", &length);
    printf("Enter width of the rectangle: ");
    scanf("%f", &width);

    /* Calculate perimeter of rectangle */
    perimeter = 2 * (length + width);

    /* Print perimeter of rectangle */
    printf("Perimeter of rectangle = %f units ", perimeter);

    return 0;
}

```

3. Write a C program to display multiple variables.

```

#include <stdio.h>
int main()
{
    int a = 125, b = 12345;
    long ax = 1234567890;
    short s = 4043;
    float x = 2.13459;
    double dx = 1.1415927;
    char c = 'W';
    unsigned long ux = 2541567890;

    printf("a + c = %d\n", a + c);
    printf("x + c = %f\n", x + c);
    printf("dx + x = %f\n", dx + x);
    printf("((int) dx) + ax = %ld\n", ((int) dx) + ax);
    printf("a + x = %f\n", a + x);
    printf("s + b = %d\n", s + b);
    printf("ax + b = %ld\n", ax + b);
    printf("s + c = %hd\n", s + c);
    printf("ax + c = %ld\n", ax + c);
    printf("ax + ux = %lu\n", ax + ux);

    return 0;
}

```

Copy

Sample Output:

```
a + c = 212
x + c = 89.134590
dx + x = 3.276183
((int) dx) + ax = 1234567891
a + x = 127.134590
s + b = 16388
ax + b = 1234580235
s + c = 4130
ax + c = 1234567977
ax + ux = 3776135780
```

Exercise 2:

1. Write a C program to calculate the distance between the two points.

C program to calculate the distance between the two points.

Note: x1, y1, x2, y2 are all double values.

Formula:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
#include <stdio.h>
#include <math.h>

int main() {
    float x1, y1, x2, y2, gdistance;
    printf("Input x1: ");
    scanf("%f", &x1);
    printf("Input y1: ");
    scanf("%f", &y1);
    printf("Input x2: ");
    scanf("%f", &x2);
    printf("Input y2: ");
    scanf("%f", &y2);
    gdistance = ((x2-x1)*(x2-x1))+((y2-y1)*(y2-y1));
    printf("Distance between the said points: %.4f", sqrt(gdistance));
    printf("\n");
    return 0;
}
Copy
```

Sample Output:

```
Input x1: 25
Input y1: 15
Input x2: 35
```

Input y2: 10

Distance between the said points: 11.1803

2. Write a C program that accepts 4 integers p, q, r, s from the user where r and s are positive and p is even. If q is greater than r and s is greater than p and if the sum of r and s is greater than the sum of p and q print "Correct values", otherwise print "Wrong values".

Write a C program that accepts 4 integers p, q, r, s from the user where q, r and s are positive and p is even. If q is greater than r and s is greater than p and if the sum of r and s is greater than the sum of p and q print "Correct values", otherwise print "Wrong values".

C Code:

```
#include <stdio.h>
int main() {
    int p, q, r, s;

    printf("\nInput the first integer: ");
    scanf("%d", &p);
    printf("\nInput the second integer: ");
    scanf("%d", &q);
    printf("\nInput the third integer: ");
    scanf("%d", &r);
    printf("\nInput the fourth integer: ");
    scanf("%d", &s);
    if((q > r) && (s > p) && ((r+s) > (p+q)) && (r > 0) && (s > 0) && (p%2 == 0))
    {
        printf("\nCorrect values\n");
    }
    else {
        printf("\nWrong values\n");
    }
    return 0;
}
```

Sample Output:

Input the first integer: 25

Input the second integer: 35

Input the third integer: 15

Input the fourth integer: 46

Wrong values

Exercise 3:

1. Write a C program to convert a string to a long integer. Write a C program to convert a string to a long integer.**Sample Solution:**

C Code:

```
#include<stdio.h>
#include<stdlib.h>
int main ()
{
    char buffer[] = "2016 40a0b0 -1101110100110111100110 0x5abfff";
    char * ptr_end;
    long int i1, i2, i3, i4;

    i1 = strtol (buffer,&ptr_end,10);
    i2 = strtol (ptr_end,&ptr_end,16);
    i3 = strtol (ptr_end,&ptr_end,2);
    i4 = strtol (ptr_end,NULL,0);
    printf ("\nIn decimals: %ld, %ld, %ld, %ld.\n\n", i1, i2, i3, i4);
    return 0;
}
```

Copy

Sample Output:

```
In decimals: 2016, 4235440, -3624422, 5947391.
```

2. Write a program in C which is a Menu-Driven Program to compute the area of the various geometrical shape.

```
#include <stdio.h>
void main ()
{
    int choice,r,l,w,b,h;
    float area;
    printf("Input 1 for area of circle\n");
    printf("Input 2 for area of rectangle\n");
    printf("Input 3 for area of triangle\n");
    printf("Input your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
```

```

case 1:
    printf("Input radius of the circle : ");
    scanf("%d",&r);
    area=3.14*r*r;
    break;
case 2:
    printf("Input length and width of the rectangle : ");
    scanf("%d%d",&l,&w);
    area=l*w;
    break;
case 3:
    printf("Input the base and height of the triangle :");
    scanf("%d%d",&b,&h);
    area=.5*b*h;
    break;
}
printf("The area is : %f\n",area);
}

```

Copy

Sample Output:

```

Input 1 for area of circle
Input 2 for area of rectangle
Input 3 for area of triangle
Input your choice : 1
Input radius of the circle : 5
The area is : 78.500000

```

3. Write a C program to calculate the factorial of a given number.

```

#include <stdio.h>
void main(){
    int i,f=1,num;
    printf("Input the number : ");
    scanf("%d",&num);
    for(i=1;i<=num;i++)
        f=f*i;
    printf("The Factorial of %d is: %d\n",num,f);
}

```

Copy

Sample Output:

```

Input the number : 5
The Factorial of 5 is: 120

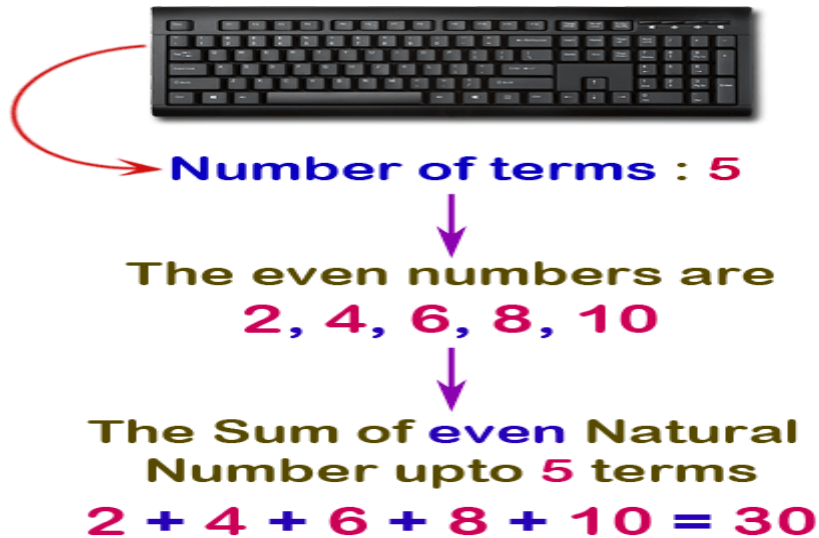
```


Exercise 4:

1. Write a program in C to display the n terms of even natural number and their sum.

Write a program in C to display the n terms of even natural number and their sum.

Pictorial Presentation:



© w3resource.com

Sample Solution:

C Code:

```
#include <stdio.h>
void main()
{
    int i,n,sum=0;
    printf("Input number of terms : ");
    scanf("%d",&n);
    printf("\nThe even numbers are :");
    for(i=1;i<=n;i++)
    {
        printf("%d ",2*i);
        sum+=2*i;
    }
    printf("\nThe Sum of even Natural Number upto %d terms : %d \n",n,sum);
}
```

Copy

Sample Output:

Input number of terms : 5

The even numbers are :2 4 6 8 10

The Sum of even Natural Number upto 5 terms : 30

2. Write a program in C to display the n terms of harmonic series and their sum.

$1 + 1/2 + 1/3 + 1/4 + 1/5 \dots 1/n$ terms.

```
#include <stdio.h>

void main()
{
    int i,n;
    float s=0.0;
    printf("Input the number of terms : ");
    scanf("%d",&n);
    printf("\n\n");
    for(i=1;i<=n;i++)
    {
        if(i<n)
        {
            printf("1/%d + ",i);
            s+=1/(float)i;
        }
        if(i==n)
        {
            printf("1/%d ",i);
            s+=1/(float)i;
        }
    }
    printf("\nSum of Series upto %d terms : %f \n",n,s);
}
```

Copy

Sample Output:

Input the number of terms : 5

$1/1 + 1/2 + 1/3 + 1/4 + 1/5$

Sum of Series upto 5 terms : 2.283334

3. Write a C program to check whether a given number is an Armstrong number or not.

Write a C program to check whether a given number is an armstrong number or not.

Pictorial Presentation:

Armstrong Number :

Number = 153

$$1^3 + 5^3 + 3^3$$
$$1 + 125 + 27 = 153$$

Sum = Original Number

153 is Armstrong Number

© w3resource.com

/*When the sum of the cube of the individual digits of a number is equal to that number, the number is called Armstrong number. For example 153.

Sum of its divisor is $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$ */

```
#include <stdio.h>
```

```
void main(){
```

```
    int num,r,sum=0,temp;
```

```
    printf("Input a number: ");
```

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

```
    for(temp=num;num!=0;num=num/10){
```

```
        r=num % 10;
```

```
        sum=sum+(r*r*r);
```

```
    }
```

```
    if(sum==temp)
```

```
        printf("%d is an Armstrong number.\n",temp);
```

```
    else
```

```
        printf("%d is not an Armstrong number.\n",temp);
```

```
}
```

Copy

Sample Output:

```
Input a number: 153
```

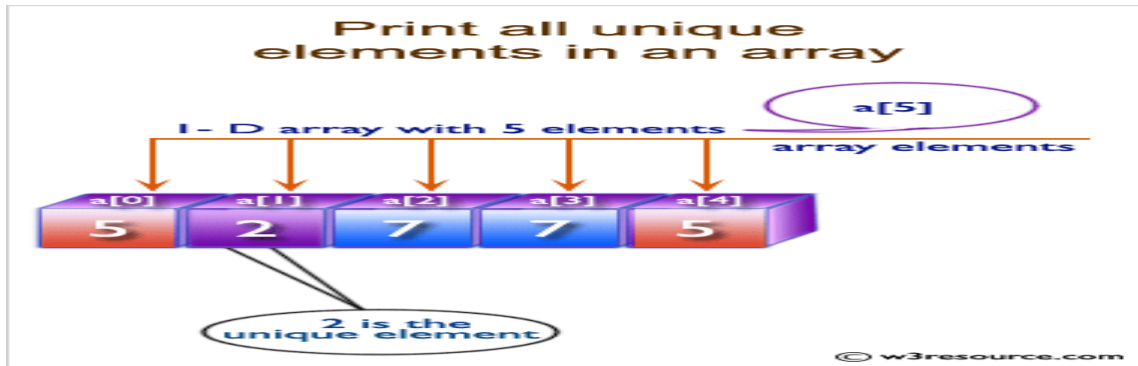
```
153 is an Armstrong number.
```

Exercise 5:

1. Write a program in C to print all unique elements in an array.

Write a program in C to print all unique elements in an array.

Pictorial Presentation:



Sample Solution:

C Code:

```
#include <stdio.h>

void main()
{
    int arr1[100], n,ctr=0;
    int i, j, k;

    printf("\n\nPrint all unique elements of an array:\n");
    printf(".....\n");
    printf("Input the number of elements to be stored in the array :");
    scanf("%d",&n);
    printf("Input %d elements in the array :\n",n);
    for(i=0;i<n;i++)
    {
        printf("element - %d : ",i);
        scanf("%d",&arr1[i]);
    }

    /*Checking duplicate elements in the array */
    printf("\nThe unique elements found in the array are : \n");
    for(i=0; i<n; i++)
    {
        ctr=0;
        /*Check duplicate before the current position and increase counter by 1 if found.*/
        for(j=0; j<i-1; j++)
        {
            /*Increment the counter when the search value is duplicate.*/
            if(arr1[i]==arr1[j])
            {
```

```

        ctr++;
    }
}
/*Check duplicate after the current position and
    increase counter by 1 if found.*/
for(k=i+1; k<n; k++)
{
    /*Increment the counter when the search value is duplicate.*/
    if(arr1[i]==arr1[k])
    {
        ctr++;
    }
}

/*Print the value of the current position of the array as unique value
when counter remain contains its initial value.*/

if(ctr==0)
{
    printf("%d ",arr1[i]);
}
}
printf("\n\n");
}
Copy

```

Sample Output:

Print all unique elements of an array:

Input the number of elements to be stored in the array :3

Input 3 elements in the array :

element - 0 : 1

element - 1 : 5

element - 2 : 1

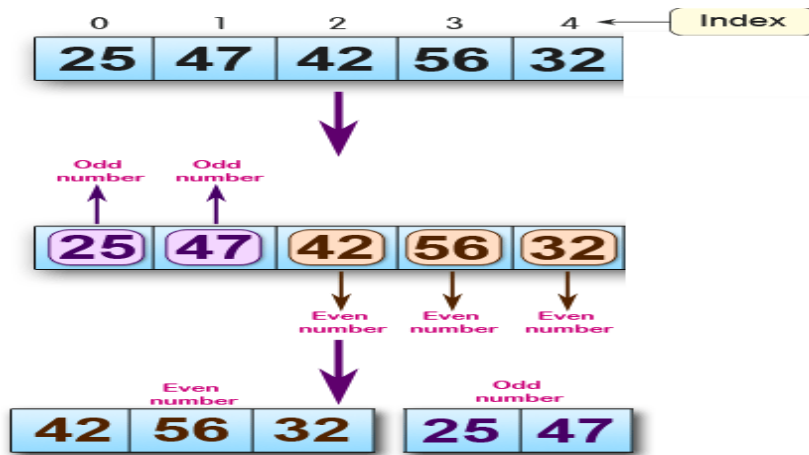
The unique elements found in the array are :

5

2. Write a program in C to separate odd and even integers in separate arrays.

Write a program in C to separate odd and even integers in separate arrays.

Pictorial Presentation:



© w3resource.com

Sample Solution:

C Code:

```
#include <stdio.h>

void main()
{
    int arr1[10], arr2[10], arr3[10];
    int i,j=0,k=0,n;
    printf("\n\nSeparate odd and even integers in separate arrays:\n");
    printf(".....\n");

    printf("Input the number of elements to be stored in the array :");
    scanf("%d",&n);
    printf("Input %d elements in the array :\n",n);
    for(i=0;i<n;i++)
    {
        printf("element - %d : ",i);
        scanf("%d",&arr1[i]);
    }
    for(i=0;i<n;i++)
    {
        if (arr1[i]%2 == 0)
        {
            arr2[j] = arr1[i];
            j++;
        }
        else
        {
```

```

        arr3[k] = arr1[i];
        k++;
    }
}
printf("\nThe Even elements are : \n");
for(i=0;i<j;i++)
{
    printf("%d ",arr2[i]);
}
printf("\nThe Odd elements are :\n");
for(i=0;i<k;i++)
{
    printf("%d ", arr3[i]);
}
printf("\n\n");
}

```

Copy

Sample Output:

Separate odd and even integers in separate arrays:

Input the number of elements to be stored in the array :5

Input 5 elements in the array :

element - 0 : 25

element - 1 : 47

element - 2 : 42

element - 3 : 56

element - 4 : 32

The Even elements are :

42 56 32

The Odd elements are :

25 47

3. Write a program in C to sort elements of array in ascending order.

```

#include <stdio.h>
void main()
{
    int arr1[100];
    int n, i, j, tmp;
    printf("\n\nsort elements of array in ascending order :\n ");
    printf(".....\n");
    printf("Input the size of array : ");

```

```

scanf("%d", &n);

printf("Input %d elements in the array :\n",n);
for(i=0;i<n;i++)
{
    printf("element - %d : ",i);
    scanf("%d",&arr1[i]);
}

for(i=0; i<n; i++)
{
    for(j=i+1; j<n; j++)
    {
        if(arr1[j] < arr1[i])
        {
            tmp = arr1[i];
            arr1[i] = arr1[j];
            arr1[j] = tmp;
        }
    }
}
printf("\nElements of array in sorted ascending order:\n");
for(i=0; i<n; i++)
{
    printf("%d ", arr1[i]);
}

    printf("\n\n");
}

```

Copy

Sample Output:

sort elements of array in ascending order :

Input the size of array : 5

Input 5 elements in the array :

element - 0 : 2

element - 1 : 7

element - 2 : 4

element - 3 : 5

element - 4 : 9

Elements of array in sorted ascending order:

Exercise 6:

1. Write a program in C for multiplication of two square Matrices.

Write a program in C for multiplication of two square Matrices.

Pictorial Presentation:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} \times b_{11} + a_{12} \times b_{21} & a_{11} \times b_{12} + a_{12} \times b_{22} \\ a_{21} \times b_{11} + a_{22} \times b_{21} & a_{21} \times b_{12} + a_{22} \times b_{22} \end{bmatrix}$$

1	2
3	4

5	6
7	8

$$\begin{bmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{bmatrix} = \begin{bmatrix} 5 + 14 & 6 + 16 \\ 15 + 28 & 18 + 32 \end{bmatrix}$$

19	22
43	50

© w3resource.com

Sample Solution:**C Code:**

```
#include <stdio.h>

void main()
{
    int arr1[50][50],arr2[50][50],arr3[50][50],i,j,k,r1,c1,r2,c2,sum=0;
    printf("\n\nMultiplication of two Matrices :\n");
    printf(".....\n");
    printf("\nInput the rows and columns of first matrix : ");
    scanf("%d %d",&r1,&c1);
    printf("\nInput the rows and columns of second matrix : ");
    scanf("%d %d",&r2,&c2);
    if(c1!=r2){
        printf("Mutiplication of Matrix is not possible.");
        printf("\nColumn of first matrix and row of second matrix must be same.");
    }
    else
    {
        printf("Input elements in the first matrix :\n");
```

```

for(i=0;i<r1;i++)
{
    for(j=0;j<c1;j++)
    {
        printf("element - [%d],[%d] : ",i,j);
        scanf("%d",&arr1[i][j]);
    }
}
printf("Input elements in the second matrix :\n");
for(i=0;i<r2;i++)
{
    for(j=0;j<c2;j++)
    {
        printf("element - [%d],[%d] : ",i,j);
        scanf("%d",&brr1[i][j]);
    }
}

printf("\nThe First matrix is :\n");
    for(i=0;i<r1;i++)
        {
            printf("\n");
            for(j=0;j<c1;j++)
                printf("%d\t",arr1[i][j]);
        }

printf("\nThe Second matrix is :\n");
    for(i=0;i<r2;i++)
        {
            printf("\n");
            for(j=0;j<c2;j++)
                printf("%d\t",brr1[i][j]);
        }

//multiplication of matrix
for(i=0;i<r1;i++)
    for(j=0;j<c2;j++)
        crr1[i][j]=0;
    for(i=0;i<r1;i++) //row of first matrix
        {
            for(j=0;j<c2;j++) //column of second matrix
                {

```

```

        sum=0;
        for(k=0;k<c1;k++)
            sum=sum+arr1[i][k]*brr1[k][j];
        crr1[i][j]=sum;
    }
}

printf("\nThe multiplication of two matrices is : \n");
for(i=0;i<r1;i++)
{
    printf("\n");
    for(j=0;j<c2;j++)
    {
        printf("%d\t",crr1[i][j]);
    }
}
printf("\n\n");
}

```

Copy

Sample Output:

Multiplication of two Matrices :

 Input the rows and columns of first matrix : 2

2

Input the rows and columns of second matrix : 2

2

Input elements in the first matrix :

element - [0],[0] : 1

element - [0],[1] : 2

element - [1],[0] : 3

element - [1],[1] : 4

Input elements in the second matrix :

element - [0],[0] : 5

element - [0],[1] : 6

element - [1],[0] : 7

element - [1],[1] : 8

The First matrix is :

1 2

3 4

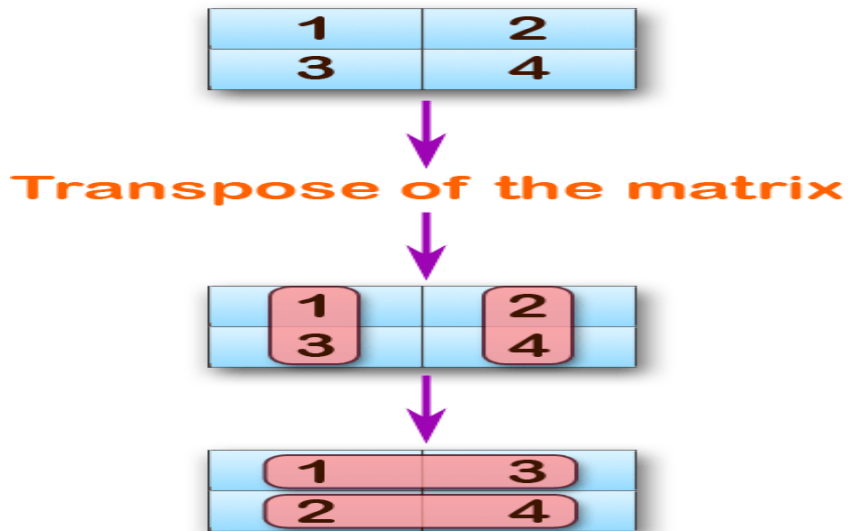
The Second matrix is :

5 6
7 8
19 22
43 50

The multiplication of two matrices is :

2. Write a program in C to find transpose of a given matrix.

Pictorial Presentation:



© w3resource.com

Sample Solution:

C Code:

```
#include <stdio.h>

void main()
{
    int arr1[50][50],brr1[50][50],i,j,k=0,r,c;
    printf("\n\nTranspose of a Matrix :\n");
    printf(".....\n");
    printf("\nInput the rows and columns of the matrix : ");
    scanf("%d %d",&r,&c);
    printf("Input elements in the first matrix :\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("element - [%d],[%d] : ",i,j);
            scanf("%d",&arr1[i][j]);
        }
    }
}
```

```

    }

    printf("\nThe matrix is :\n");
        for(i=0;i<r;i++)
            {
                printf("\n");
                for(j=0;j<c;j++)
                    printf("%d\t",arr1[i][j]);
            }
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
    brr1[j][i]=arr1[i][j];
}
}
printf("\n\nThe transpose of a matrix is : ");
for(i=0;i<c;i++){
printf("\n");
for(j=0;j<r;j++){
    printf("%d\t",brr1[i][j]);
}
}
printf("\n\n");
}

```

Copy

Sample Output:

Transpose of a Matrix :

Input the rows and columns of the matrix : 2 2

Input elements in the first matrix :

element - [0],[0] : 1

element - [0],[1] : 2

element - [1],[0] : 3

element - [1],[1] : 4

The matrix is :

1 2

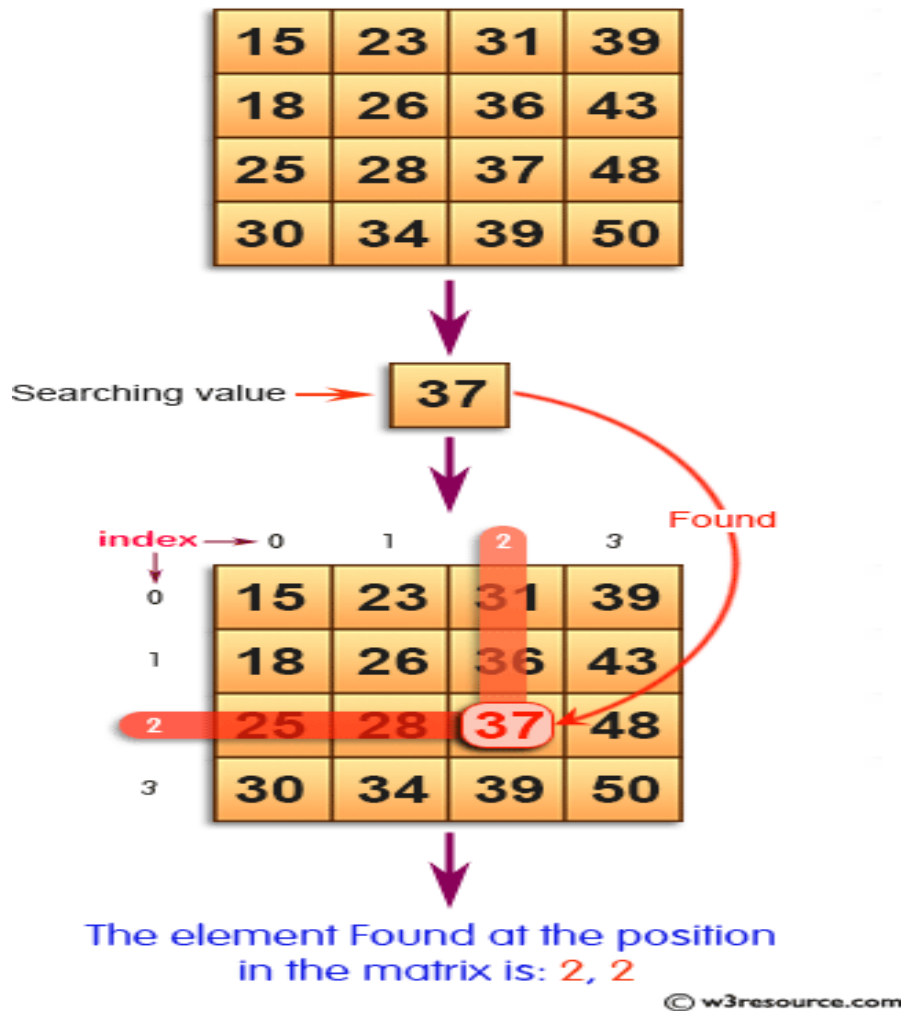
3 4

The transpose of a matrix is :

1 3

Exercise 7:

1. Write a program in C to search an element in a row wise and column wise sorted matrix.

Pictorial Presentation:**Sample Solution:****C Code:**

```
#include <stdio.h>
int searchElement(int arr2D[4][4], int n, int x)
{
    int i = 0, j = n-1;
    while ( i < n && j >= 0 )
    {
        if ( arr2D[i][j] == x )
        {
            printf("\nThe element Found at the position in the matrix is: %d, %d", i, j);
        }
    }
}
```

```

        return 1;
    }
    if ( arr2D[i][j] < x )
        j--;
    else
        i++;
    }
    printf("\n\nThe given element not found in the 2D array.");
    return 0;
}
int main()
{
    int arr2D[4][4] = { { 15, 23, 31, 39},
                       { 18, 26, 36, 43},
                       { 25, 28, 37, 48},
                       { 30, 34, 39, 50},
                       };

    int i,j,v;
    v=37;
    //.....print original array .....
    printf("The given array in matrix form is : \n");
    for(i = 0; i < 4; i++)
    {
        for (j=0;j<4;j++)
        {
            printf("%d ", arr2D[i][j]);
        }

        printf("\n");
    }

    //.....
    printf("The given value for searching is: %d",v);
    searchElement(arr2D, 4, v);
    return 0;
}

```

Copy

Sample Output:

```

The given array in matrix form is :
15 23 31 39
18 26 36 43

```

25 28 37 48

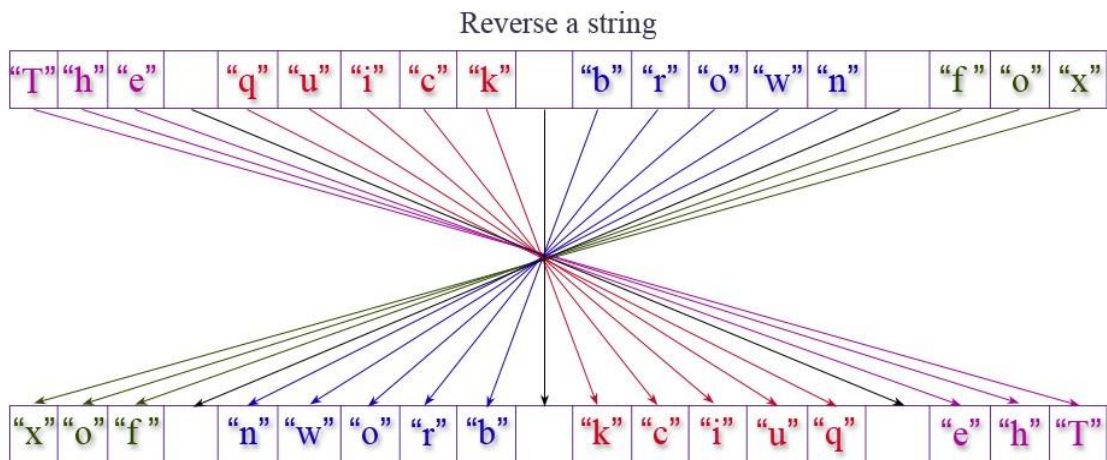
30 34 39 50

The given value for searching is: 37

The element Found at the position in the matrix is: 2, 2

2. Write a program in C to print individual characters of string in reverse order.

Write a program in C to print individual characters of string in reverse order.



© w3resource.com

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void main()
{
    char str[100]; /* Declares a string of size 100 */
    int l,i;
    printf("\n\nPrint individual characters of string in reverse order :\n");
    printf(".....\n");
    printf("Input the string : ");
    fgets(str, sizeof str, stdin);
    l=strlen(str);
    printf("The characters of the string in reverse are : \n");
    for(i=l;i>=0;i--)
    {
        printf("%c ", str[i]);
    }
    printf("\n");
}
```



```
}
```

Exercise 8:

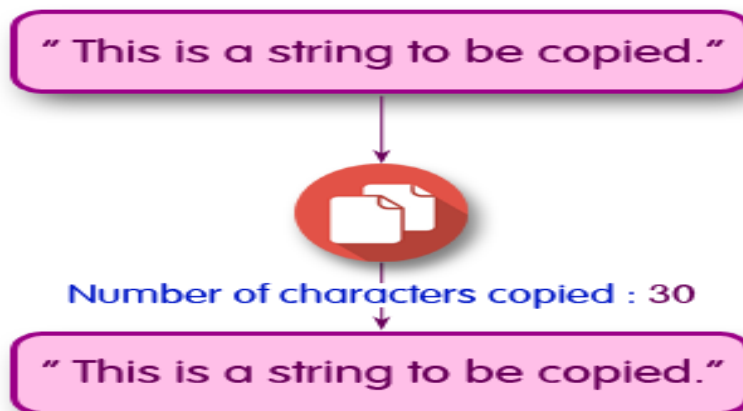
1. Write a program in C to compare two strings without using string library functions.

```
#include<stdio.h>
```

```
int main() {  
    char str1[30], str2[30];  
    int i;  
  
    printf("\nEnter two strings :");  
    gets(str1);  
    gets(str2);  
  
    i = 0;  
    while (str1[i] == str2[i] && str1[i] != '\0')  
        i++;  
    if (str1[i] > str2[i])  
        printf("str1 > str2");  
    else if (str1[i] < str2[i])  
        printf("str1 < str2");  
    else  
        printf("str1 = str2");  
  
    return (0);  
}
```

2. Write a program in C to copy one string to another string.

Write a program in C to copy one string to another string.



© w3resource.com

Sample Solution:

C Code:

```
#include <stdio.h>
```

```
#include <string.h>
```

```

#include <stdlib.h>
void main()
{
    char str1[100], str2[100];
    int i;
    printf("\n\nCopy one string into another string :\n");
    printf(".....\n");
    printf("Input the string : ");
    fgets(str1, sizeof str1, stdin);
    /* Copies string1 to string2 character by character */
    i=0;
    while(str1[i]!='\0')
    {
        str2[i] = str1[i];
        i++;
    }

    //Makes sure that the string is NULL terminated
    str2[i] = '\0';
    printf("\nThe First string is : %s\n", str1);
    printf("The Second string is : %s\n", str2);
    printf("Number of characters copied : %d\n\n", i);
}

```

Copy

Sample Output:

Copy one string into another string :

Input the string : This is a string to be copied.

The First string is : This is a string to be copied.

The Second string is : This is a string to be copied.

Number of characters copied : 31

Exercise 9:

1. Write a C Program to Store Information Using Structures with Dynamically Memory

Allocation

1. #include <stdio.h>
2. s struct studen
- 3.
4. char name[50]
5. int roll

```

6.   float marks;
7.   } s[10];
8.
9.   int main()
10.  {
11.   int i;
12.
13.   printf("Enter information of students:\n");
14.
15.   // storing information
16.   for(i=0; i<10; ++i)
17.   {
18.     s[i].roll = i+1;
19.
20.     printf("\nFor roll number%d,\n",s[i].roll);
21.
22.     printf("Enter name: ");
23.     scanf("%s",s[i].name);
24.
25.     printf("Enter marks: ");
26.     scanf("%f",&s[i].marks);
27.
28.     printf("\n");
29.   }
30.
31.   printf("Displaying Information:\n\n");
32.   // displaying information
33.   for(i=0; i<10; ++i)
34.   {
35.     printf("\nRoll number: %d\n",i+1);
36.     printf("Name: ");
37.     puts(s[i].name);
38.     printf("Marks: %.1f",s[i].marks);
39.     printf("\n");
40.   }
41.   return 0;
42. }
43.

```

Output

Enter information of students:

For roll number1,
Enter name: Tom
Enter marks: 98

For roll number2,
Enter name: Jerry
Enter marks: 89

.
.
.

Displaying Information:

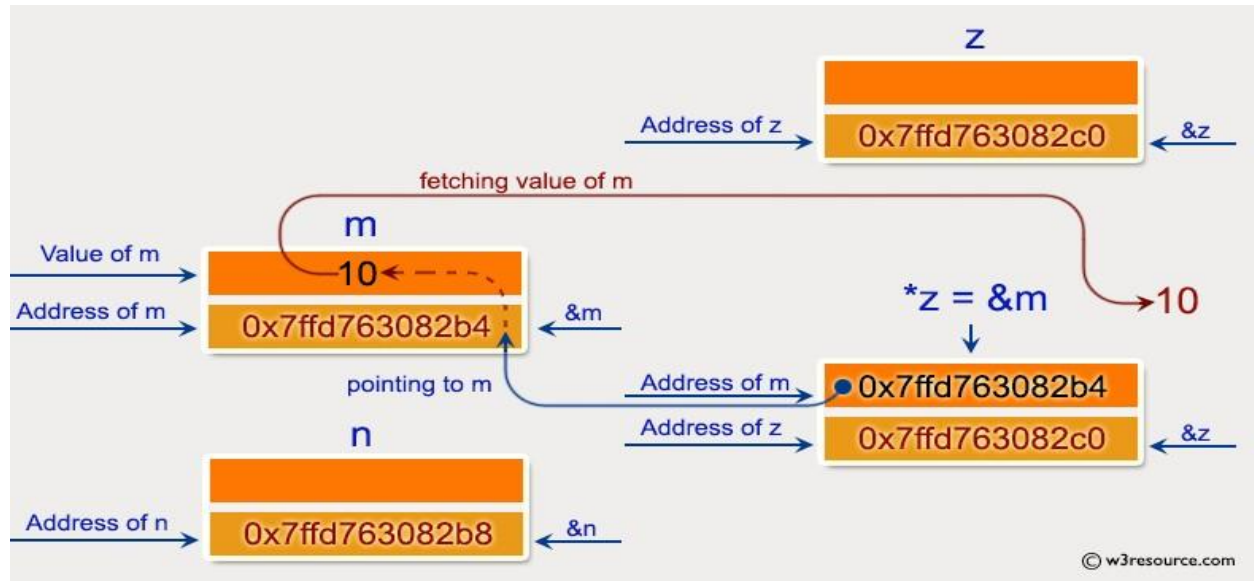
Roll number: 1
Name: Tom
Marks: 98

.
.
.

2. Write a program in C to demonstrate how to handle the pointers in the program.

Write a program in C to show the basic declaration of pointer.

Pictorial Presentation:



Sample Solution:

C Code:

```
#include <stdio.h>
void main(void)
{
int m=10,n,o;
int *z=&m ;

printf("\n\n Pointer : Show the basic declaration of pointer :\n");
printf(".....\n");
printf(" Here is m=10, n and o are two integer variable and *z is an integer");
printf("\n\n z stores the address of m = %p\n", z); // z is a pointer so %p would print the address
printf("\n *z stores the value of m = %i\n", *z);
printf("\n &m is the address of m = %p\n", &m); // &m gives the address of the integer variable m
// so %p is the specifier for that address
printf("\n &n stores the address of n = %p\n", &n);
printf("\n &o stores the address of o = %p\n", &o);
printf("\n &z stores the address of z = %p\n", &z); // &z gives the address, where the pointer z is
// stored -> still an address -> %p is the right
// specifier
}
```

Copy

Sample Output:

Pointer : Show the basic declaration of pointer :

Here is m=10, n and o are two integer variable and *z is an integer

z stores the address of m = 0x7ffd763082b4

*z stores the value of m = 10

&m is the address of m = 0x7ffd763082b4

&n stores the address of n = 0x7ffd763082b8

&o stores the address of o = 0x7ffd763082bc

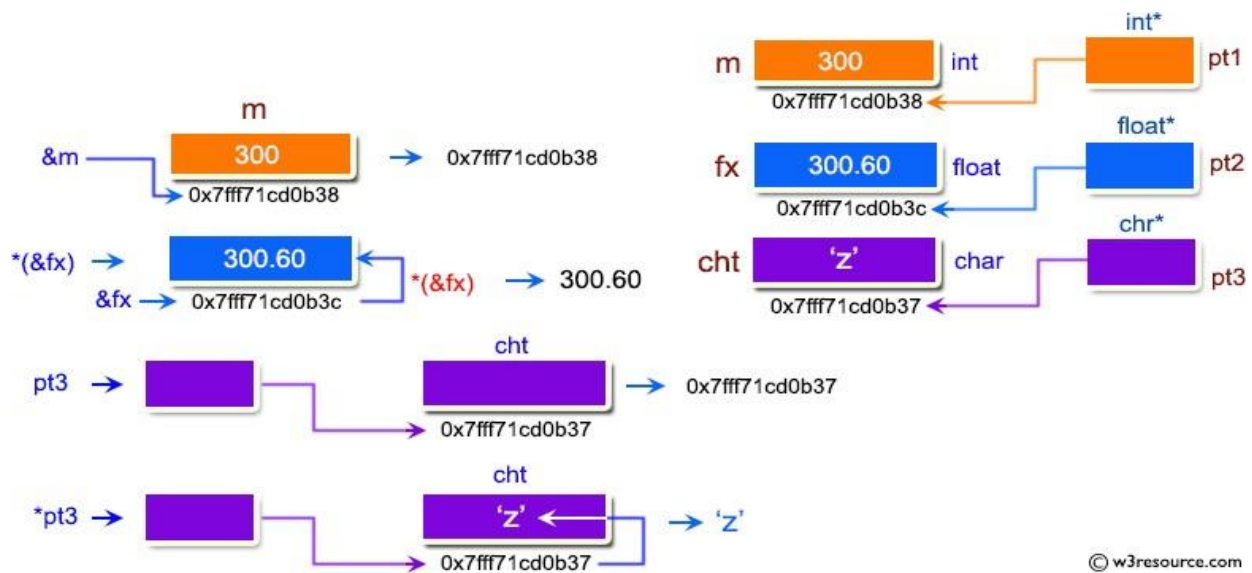
&z stores the address of z = 0x7ffd763082c0

Exercise 10:

1. Write a program in C to demonstrate the use of & (address of) and *(value at address) operator.

// Write a program in C to demonstrate the use of &(address of) and *(value at address) operator.

Pictorial Presentation:



Sample Solution:

C Code:

```
#include <stdio.h>
void main()
{
    int m=300;
```

```

float fx = 300.60;
char cht = 'z';
    printf("\n\n Pointer : Demonstrate the use of & and * operator :\n");
    printf(".....\n");
int *pt1;
float *pt2;
char *pt3;
pt1 = &m;
pt2 = &fx;
pt3 = &cht;
printf ( " m = %d\n",m);
printf ( " fx = %f\n",fx);
printf ( " cht = %c\n",cht);
printf("\n Using & operator :\n");
printf(".....\n");
printf ( " address of m = %p\n",&m);
printf ( " address of fx = %p\n",&fx);
printf ( " address of cht = %p\n",&cht);
printf("\n Using & and * operator :\n");
printf(".....\n");
printf ( " value at address of m = %d\n",*(&m));
printf ( " value at address of fx = %f\n",*(&fx));
printf ( " value at address of cht = %c\n",*(&cht));
printf("\n Using only pointer variable :\n");
printf(".....\n");
printf ( " address of m = %p\n",pt1);
printf ( " address of fx = %p\n",pt2);
printf ( " address of cht = %p\n",pt3);
printf("\n Using only pointer operator :\n");
printf(".....\n"); printf
( " value at address of m = %d\n",*pt1);printf
( " value at address of fx= %f\n",*pt2);
printf ( " value at address of cht= %c\n\n",*pt3);
}

```

Copy

Sample Output:

Pointer : Demonstrate the use of & and * operator :

```

-----
m = 300
fx = 300.600006

```

cht = z

Using & operator :

address of m = 0x7fff71cd0b38
address of fx = 0x7fff71cd0b3c
address of cht = 0x7fff71cd0b37

Using & and * operator :

value at address of m = 300
value at address of fx = 300.600006
value at address of cht = z

Using only pointer variable :

address of m = 0x7fff71cd0b38
address of fx = 0x7fff71cd0b3c
address of cht = 0x7fff71cd0b37

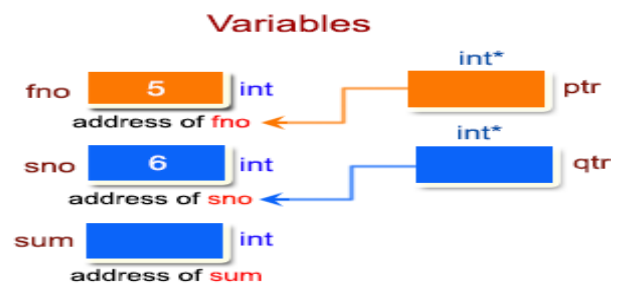
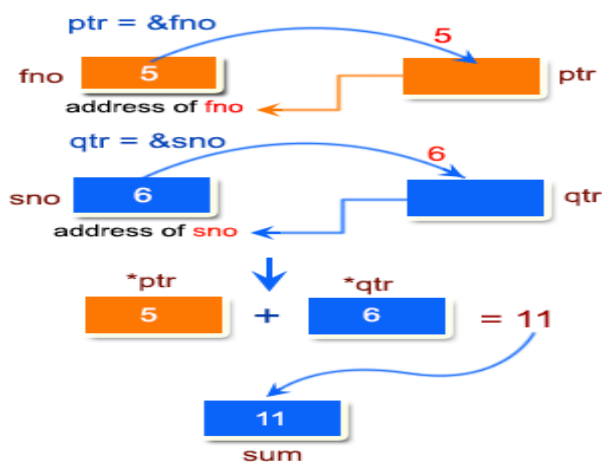
Using only pointer operator :

value at address of m = 300
value at address of fx = 300.600006
value at address of cht = z

2. Write a program in C to add two numbers using pointers.

Write a program in C to add two numbers using pointers.

Pictorial Presentation:



© w3resource.com

Sample Solution:

C Code:

```
#include <stdio.h>
int main()
{
    int fno, sno, *ptr, *qtr, sum;
```

```

printf("\n\n Pointer : Add two numbers :\n");
printf(".....\n");
printf(" Input the first number : ");
scanf("%d", &fno);
printf(" Input the second number : ");
scanf("%d", &sno);
ptr = &fno;
qtr = &sno;
sum = *ptr + *qtr;
printf(" The sum of the entered numbers is : %d\n\n",sum);
return 0;
}

```

Copy

Sample Output:

Pointer : Add two numbers :

Input the first number : 5

Input the second number : 6

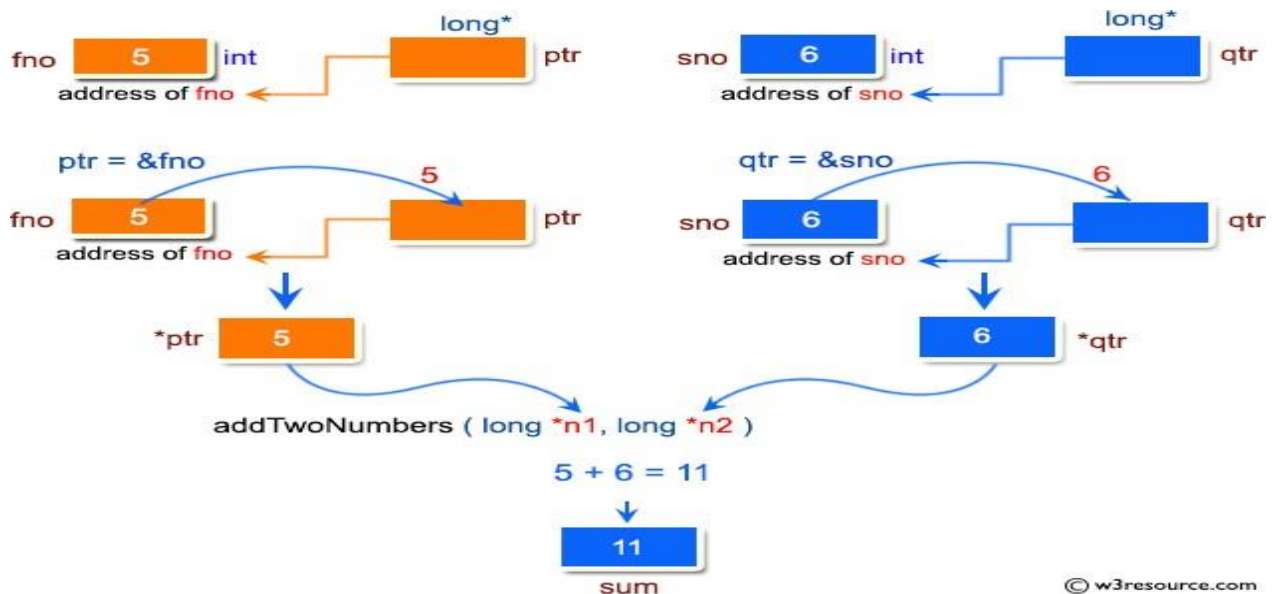
The sum of the entered numbers is : 11

Exercise 11:

1. Write a program in C to add numbers using call by reference.

Write a program in C to add numbers using call by reference.

Pictorial Presentation:



Sample Solution:

C Code:

```
#include <stdio.h>
long addTwoNumbers(long *, long *);
int main()
{
    long fno, sno, *ptr, *qtr, sum;
    printf("\n\n Pointer : Add two numbers using call by reference:\n");
    printf(".....\n");
    printf(" Input the first number : ");
    scanf("%ld", &fno);
    printf(" Input the second number : ");
    scanf("%ld", &sno);
    sum = addTwoNumbers(&fno, &sno);
    printf(" The sum of %ld and %ld is %ld\n\n", fno, sno, sum);
    return 0;
}
long addTwoNumbers(long *n1, long *n2)
{
    long sum;
    sum = *n1 + *n2;
    return sum;
}
```

Copy

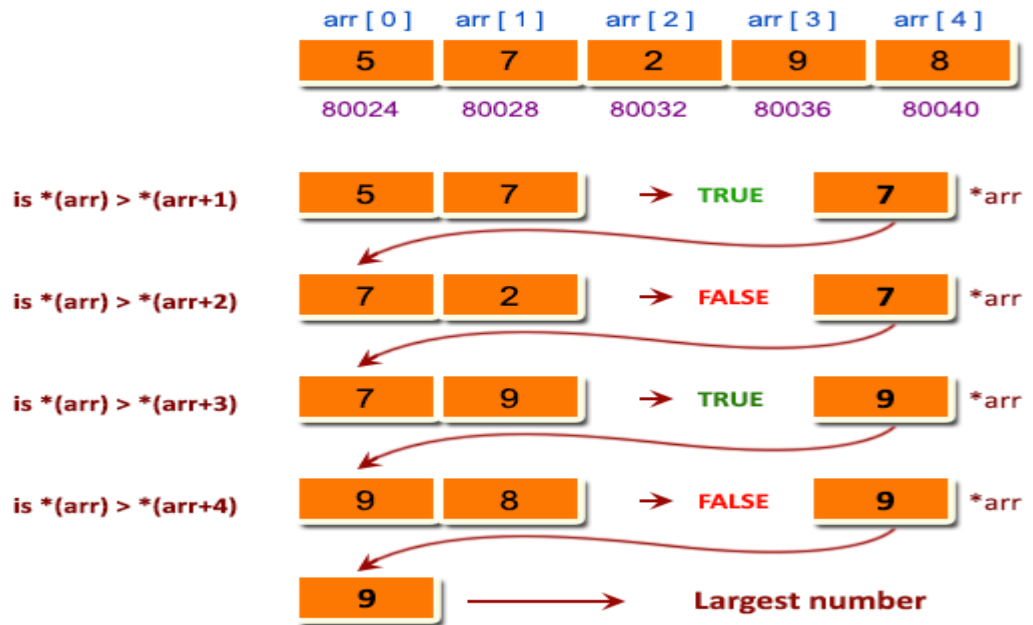
Sample Output:

Pointer : Add two numbers using call by reference:

.....
Input the first number : 5
Input the second number : 6
The sum of 5 and 6 is 11

2. Write a program in C to find the largest element using Dynamic Memory Allocation.
Write a program in C to find the largest element using Dynamic Memory Allocation.

Pictorial Presentation:



© w3resource.com

Sample Solution:

C Code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i,n;
    float *element;
    printf("\n\n Pointer : Find the largest element using Dynamic Memory Allocation :\n");
    printf(".....\n");
    printf(" Input total number of elements(1 to 100): ");
    scanf("%d",&n);
    element=(float*)calloc(n,sizeof(float)); // Memory is allocated for 'n' elements
    if(element==NULL)
    {
        printf(" No memory is allocated.");
        exit(0);
    }
    printf("\n");
    for(i=0;i<n;++i)
    {
        printf(" Number %d: ",i+1);
        scanf("%f",element+i);
    }
}
```

```

}
for(i=1;i<n;++i)
{
    if(*element<*(element+i))
        *element=*(element+i);
}
printf(" The Largest element is : %.2f \n\n",*element);
return 0;
}

```

Copy

Sample Output:

Pointer : Find the largest element using Dynamic Memory Allocation :

 Input total number of elements(1 to 100): 5

Number 1: 5

Number 2: 7

Number 3: 2

Number 4: 9

Number 5: 8

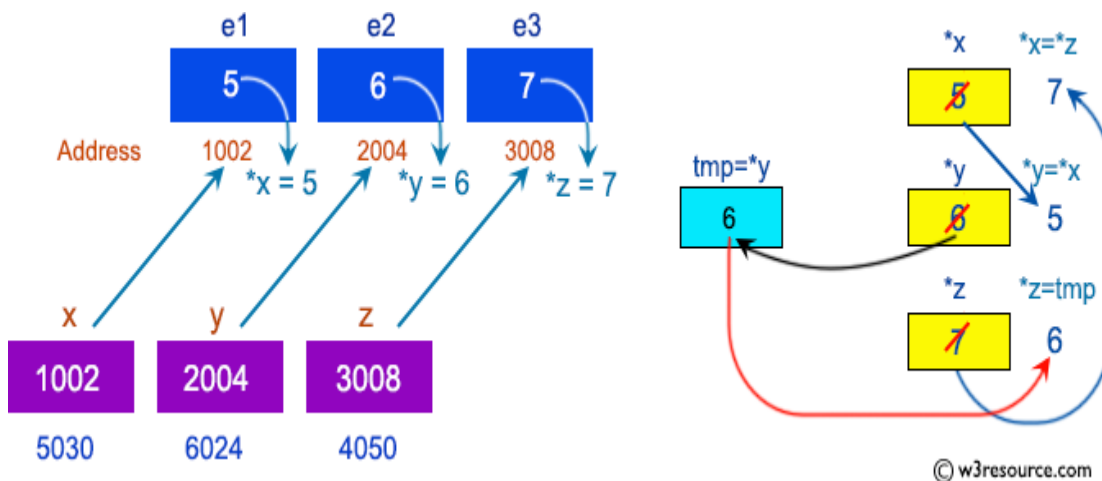
The Largest element is : 9.00

Exercise 12:

1. Write a program in C to swap elements using call by reference.

Write a program in C to swap elements using call by reference.

Pictorial Presentation:



Sample Solution:

C Code:

```
#include <stdio.h>
```

```

void swapNumbers(int *x,int *y,int *z);
int main()
{
    int e1,e2,e3;
        printf("\n\n Pointer : Swap elements using call by reference :\n");
        printf(".....\n");
    printf(" Input the value of 1st element : ");
    scanf("%d",&e1);
        printf(" Input the value of 2nd element : ");
    scanf("%d",&e2);
        printf(" Input the value of 3rd element : ");
    scanf("%d",&e3);
    printf("\n The value before swapping are :\n");
    printf(" element 1 = %d\n element 2 = %d\n element 3 = %d\n",e1,e2,e3);
    swapNumbers(&e1,&e2,&e3);
    printf("\n The value after swapping are :\n");
    printf(" element 1 = %d\n element 2 = %d\n element 3 = %d\n\n",e1,e2,e3);
    return 0;
}
void swapNumbers(int *x,int *y,int *z)
{
    int tmp;
    tmp=*y;
    *y=*x;
    *x=*z;
    *z=tmp;
}

```

Copy

Sample Output:

Pointer : Swap elements using call by reference :

.....
Input the value of 1st element : 5
Input the value of 2nd element : 6
Input the value of 3rd element : 7

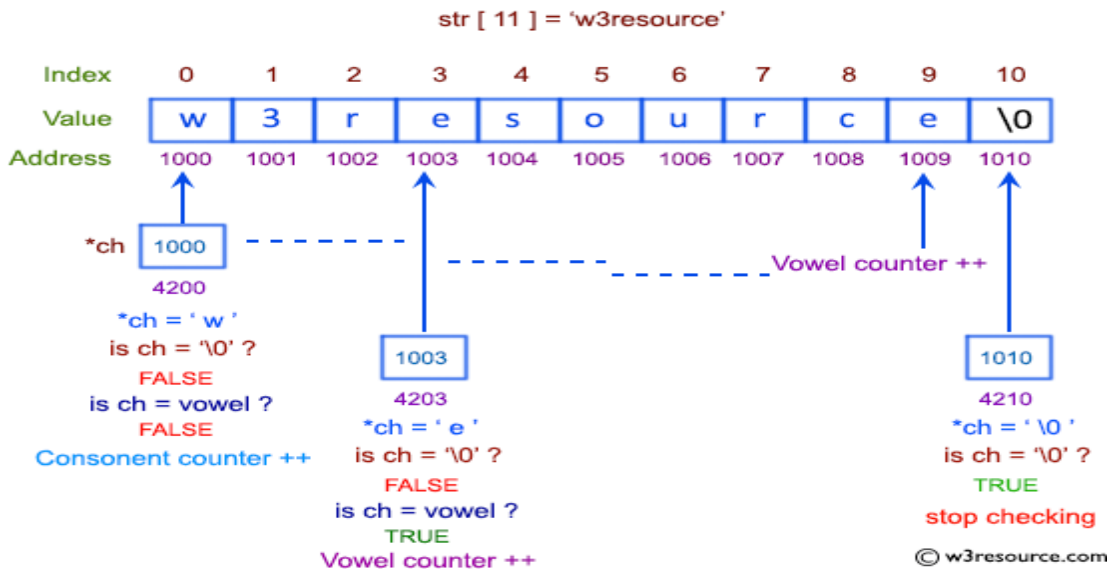
The value before swapping are :
element 1 = 5
element 2 = 6
element 3 = 7

The value after swapping are :
element 1 = 7

element 2 = 5
 element 3 = 6

2. Write a program in C to count the number of vowels and consonants in a string using a pointer.

Pictorial Presentation:



Sample Solution:

C Code:

```
#include <stdio.h>

int main()
{
    char str1[50];
    char *pt;
    int ctrV,ctrC;

    printf("\n\n Pointer : Count the number of vowels and consonants :\n");
    printf(".....\n");

    printf(" Input a string: ");
    fgets(str1, sizeof str1, stdin);

    //assign address of str1 to pt
    pt=str1;
    ctrV=ctrC=0;
    while(*pt!='\0')
    {
        if(*pt=='A' ||*pt=='E' ||*pt=='I' ||*pt=='O' ||*pt=='U' ||*pt=='a' ||*pt=='e' ||*pt=='i' ||*pt=='o' ||*pt=='u')
            ctrV++;
    }
}
```

```

else
    ctrC++;
    pt++; //pointer is increasing for searching the next character
}

printf(" Number of vowels : %d\n Number of consonants : %d\n",ctrV,ctrC-1);
return 0;
}

```

Copy

Sample Output:

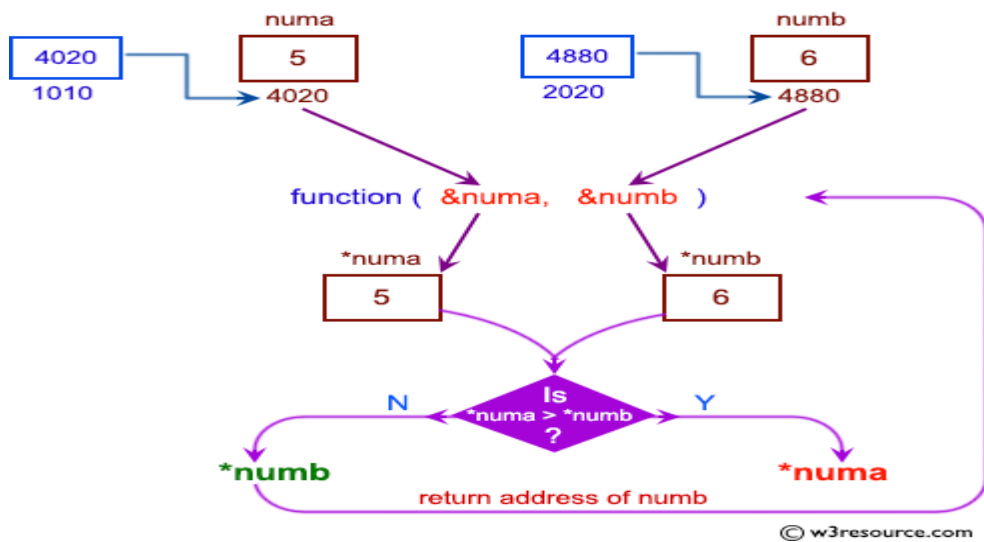
Pointer : Count the number of vowels and consonants :

 Input a string: string
 Number of vowels : 1
 Number of consonants : 5

Exercise 13:

1. Write a program in C to show how a function returning pointer.

Pictorial Presentation:



Sample Solution:

C Code:

```

#include <stdio.h>
int* findLarger(int*, int*);
void main()
{
    int numa=0;
    int numb=0;

```

```

int *result;

    printf("\n\n Pointer : Show a function returning pointer :\n");
    printf(".....\n");

printf(" Input the first number : ");
scanf("%d", &numa);
printf(" Input the second number : ");
scanf("%d", &numb);

result=findLarger(&numa, &numb);
printf(" The number %d is larger. \n\n",*result);
}

int* findLarger(int *n1, int *n2)
{
if(*n1 > *n2)
return n1;
else
return n2;
}

```

Copy

Sample Output:

```

Pointer : Show a function returning pointer :
.....
Input the first number : 5
Input the second number : 6
The number 6 is larger.

```

2. Write a C program to find sum of n elements entered by user. To perform this program, allocate memory dynamically using malloc() function.

malloc

malloc function **allocates memory at runtime**. It takes the size in bytes and allocates that much space in the memory. It means that malloc(50) will allocate 50 byte in the memory. It returns a void pointer and is defined in **stdlib.h**.

```

#include<stdio.h>
#include<stdlib.h>
main()
{
int *p,i,n,sum=0;
printf("\nEnter the elements size: ");
scanf("%d",&n);
p=(int *)malloc(n * sizeof(int));

```

```

printf("\nEnter the array values: \n");
for(i=0;i<n;i++)
scanf("%d",p+i);
for(i=0;i<n;i++)
{
sum = sum + *p;
p++;
}
printf("\nThe sum of elements is: %d\n",sum);
}

```

Exercise 14:

1. Write a C program to find sum of n elements entered by user. To perform this program, allocate memory dynamically using calloc() function. Understand the difference between the above two programs

calloc

Now, suppose you want to put more than one toy in a box and you have only an approximate idea of the number of toys and the size of each. For that, you need a box the size of which is equal to the sum of the sizes of all the toys.

In such cases, we use **calloc** function. Like malloc, calloc also **allocates memory at runtime** and is defined in **stdlib.h**. It takes the number of elements and the size of each element(in bytes), initializes each element to zero and then returns a void pointer to the memory.

Its syntax is

```
void *calloc(n, element-size);
```

Here, 'n' is the number of elements and 'element-size' is the size of each element.

Let's see the last example of malloc again, but this time with calloc.

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n,i,*p;
    printf("Enter number of elements: ");
    scanf("%d",&n);
    p=(int*)calloc(n, sizeof(int)); //memory allocated using malloc
    if(p == NULL)
    {
        printf("memory cannot be allocated\n");
    }
    else
    {
        printf("Enter elements of array:\n");
        for(i=0;i<n;++i)
        {
            scanf("%d",&*(p+i));
        }
        printf("Elements of array are\n");
        for(i=0;i<n;i++)
        {
            printf("%d\n",*(p+i));
        }
    }
    return 0;
}

```

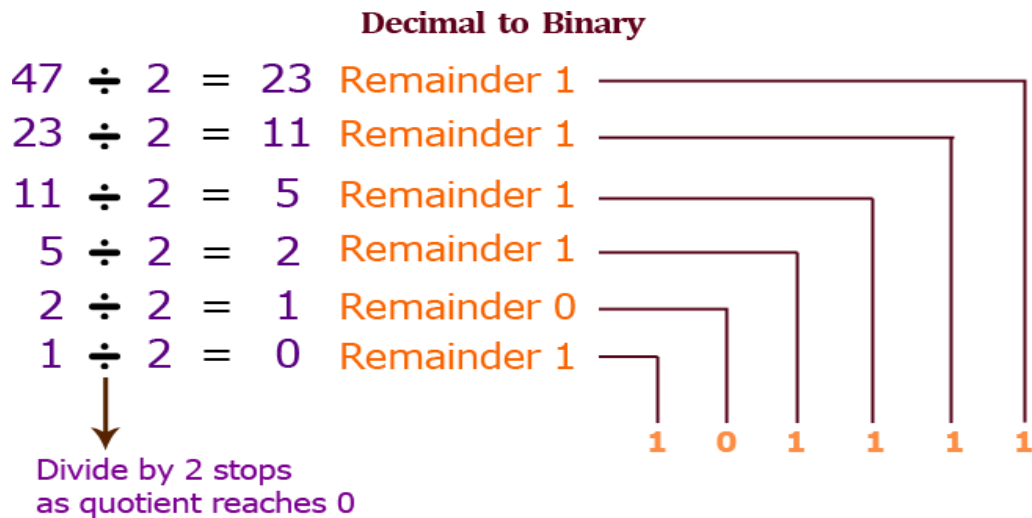

Output

So, this is same as the example of malloc, with a difference in the syntax of calloc. Here we wrote `(int*)calloc(n, sizeof(int))`, where `n` is the number of elements in the integer array (5 in this case) and `sizeof(int)` is the size of each of that element. So the total size of the array is `n * sizeof(int)`.

2. Write a program in C to convert decimal number to binary number using the function.

Write a program in C to convert decimal number to binary number using the function.

Pictorial Presentation:



© w3resource.com

Sample Solution:

C Code:

```
#include<stdio.h>
long toBin(int);
int main()
{
    long bno;
    int dno;

    printf("\n\n Function : convert decimal to binary :\n");
    printf(".....\n");

    printf(" Input any decimal number : ");
    scanf("%d",&dno);
    bno = toBin(dno);
    printf("\n The Binary value is : %ld\n\n",bno);
    return 0;
}
long toBin(int dno)
{
```

```
long bno=0,remainder,f=1;
while(dno != 0)
{
    remainder = dno % 2;
    bno = bno + remainder * f;
    f = f * 10;
    dno = dno / 2;
}
return bno;
}
```

Copy

Sample Output:

Function : convert decimal to binary :

Input any decimal number : 65

The Binary value is : 1000001

Exercise 15:

1. Write a program in C to check whether a number is a prime number or not using the function.

Pictorial Presentation:

A **prime** number is a positive integer with only two factors : itself and one

$$\begin{array}{r} 2 \overline{) 2} \\ 1 \end{array}$$

$2 = 2 \times 1$
two factors only

Prime Number

$$\begin{array}{r} 2 \overline{) 6} \\ 3 \overline{) 3} \\ 1 \end{array}$$

$6 = 2 \times 3 \times 1$
three factors !

Not a Prime Number

$$\begin{array}{r} 2 \overline{) 12} \\ 2 \overline{) 6} \\ 3 \overline{) 3} \\ 1 \end{array}$$

$12 = 2 \times 2 \times 3 \times 1$
four factors !

Not a Prime Number

Prime number between 1 to 100

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

© w3resource.com

Sample Solution:

C Code:

```
#include<stdio.h>
int PrimeOrNot(int);
int main()
{
    int n1,prime;
    printf("\n\n Function : check whether a number is prime number or not :\n");
```

```

        printf(".....\n");

printf(" Input a positive number : ");
scanf("%d",&n1);
prime = PrimeOrNot(n1);
if(prime==1)
    printf(" The number %d is a prime number.\n",n1);
else
    printf(" The number %d is not a prime number.\n",n1);
return 0;
}
int PrimeOrNot(int n1)
{
    int i=2;
    while(i<=n1/2)
    {
        if(n1%i==0)
            return 0;
        else
            i++;
    }
    return 1;
}

```

Copy

Sample Output:

Function : check whether a number is prime number or not :

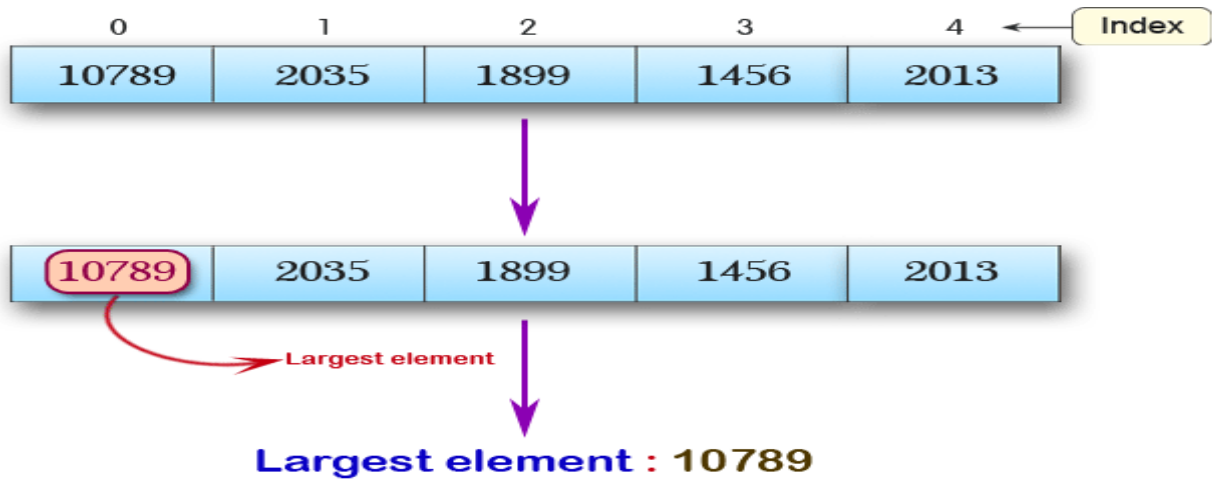
```

-----
Input a positive number : 5
The number 5 is a prime number.

```

2. Write a program in C to get the largest element of an array using the function.

Pictorial Presentation:



© w3resource.com

Sample Solution:

C Code:

```
#include<stdio.h>
#define MAX 100
int findMaxElem(int []);
int n;
int main()
{
    int arr1[MAX],mxelem,i;
    printf("\n\n Function : get largest element of an array :\n");
    printf(".....\n");

    printf(" Input the number of elements to be stored in the array :");
    scanf("%d",&n);

    printf(" Input %d elements in the array :\n",n);
    for(i=0;i<n;i++)
    {
        printf(" element - %d : ",i);
        scanf("%d",&arr1[i]);
    }
    mxelem=findMaxElem(arr1);
    printf(" The largest element in the array is : %d\n\n",mxelem);
    return 0;
}
int findMaxElem(int arr1[])
```

```

{
    int i=1,mxelem;
    mxelem=arr1[0];
    while(i < n)
        {
            if(mxelem<arr1[i])
                mxelem=arr1[i];
            i++;
        }
    return mxelem;
}

```

Copy

Sample Output:

Function : get largest element of an array :

Input the number of elements to be stored in the array :5

Input 5 elements in the array :

element - 0 : 1

element - 1 : 2

element - 2 : 3

element - 3 : 4

element - 4 : 5

The largest element in the array is : 5

Exercise 16:

1. Write a program in C to append multiple lines at the end of a text file.

Write a program in C to append multiple lines at the end of a text file.

Assume that the content of the file test.txt is :

test line 1

test line 2

test line 3

test line 4

Sample Solution:

C Code:

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    FILE * fptr;
```

```
    int i,n;
```

```
    char str[100];
```

```
    char fname[20];
```

```
    char str1;
```

```

printf("\n\n Append multiple lines at the end of a text file :\n");
printf(".....\n");
printf(" Input the file name to be opened : ");
scanf("%s",fname);

fptr = fopen(fname, "a");
printf(" Input the number of lines to be written : ");
scanf("%d", &n);
printf(" The lines are : \n");
for(i = 0; i < n+1;i++)
{
fgets(str, sizeof str, stdin);
fputs(str, fptr);
}
fclose (fptr);

//----- Read the file after appended -----
fptr = fopen (fname, "r");
printf("\n The content of the file %s is :\n",fname);
str1 = fgetc(fptr);
while (str1 != EOF)
{
printf ("%c", str1);
str1 = fgetc(fptr);
}

printf("\n\n");
fclose (fptr);

//.....End of reading .....
return 0;
}

```

Copy

Sample Output:

Append multiple lines at the end of a text file :

```

-----
Input the file name to be opened : test.txt
Input the number of lines to be written : 3
The lines are :
test line 5
test line 6
test line 7

```

The content of the file test.txt is :

```
test line 1
test line 2
test line 3
test line 4
```

```
test line 5
test line 6
test line 7
```

2. Write a program in C to copy a file in another name.

Write a program in C to copy a file in another name.

Assume that the content of the file test.txt is :

```
test line 1
test line 2
test line 3
test line 4
```

Sample Solution:

C Code:

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    FILE *fptr1, *fptr2;
    char ch, fname1[20], fname2[20];

    printf("\n\n Copy a file in another name :\n");
    printf(".....\n");

    printf(" Input the source file name : ");
    scanf("%s",fname1);

    fptr1=fopen(fname1, "r");
    if(fptr1==NULL)
    {
        printf(" File does not found or error in opening.!!");
        exit(1);
    }
    printf(" Input the new file name : ");
    scanf("%s",fname2);
    fptr2=fopen(fname2, "w");
    if(fptr2==NULL)
```



```

        {
            printf(" File does not found or error in opening.!!");
            fclose(fp1);
            exit(2);
        }
        while(1)
        {
            ch=fgetc(fp1);
            if(ch==EOF)
            {
                break;
            }
            else
            {
                fputc(ch, fp2);
            }
        }
        printf(" The file %s copied successfully in the file %s. \n\n",fname1,fname2);
        fclose(fp1);
        fclose(fp2);
        getchar();
    }

```

Copy

Sample Output:

Copy a file in another name :

 Input the source file name : test.txt

Input the new file name : test1.txt

The file test.txt copied successfully in the file test1.txt.

3. Write a program in C to remove a file from the disk.

Write a program in C to remove a file from the disk.

Sample Solution:

C Code:

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int status;
```

```
    char fname[20];
```

```
printf("\n\n Remove a file from the disk :\n");
printf(".....\n");
printf(" Input the name of file to delete : ");
scanf("%s",fname);
status=remove(fname);
if(status==0)
{
    printf(" The file %s is deleted successfully..!!\n",fname);
}
else
{
    printf(" Unable to delete file %s\n",fname);
}
}
```

Copy

Sample Output:

Remove a file from the disk :

Input the name of file to delete : test.txt

The file test.txt is deleted successfully..!!